Парадигма развития науки

Методологическое обеспечение

А. Е. Кононюк

ОБЩАЯ ТЕОРИЯ РАСПОЗНАВАНИЯ

Книга 2

Математические средства описания распознаваемых объектов и распознающих процессов

Киев
Освіта України
2012

УДК 51 (075.8)
ББК В161.я7
К 213

Рецензент: *Н.К.Печурин* - д-р техн. наук, проф. (Национальный авиационный университет).

Кононюк А. Е.
К65 Общая теория распознавания. К.2.
К.4:"Освіта України", 2012. - 588 с.
ISBN 978-966-7599-50-8

Настоящая работа является систематическим изложением общей теории распознавания. Основное внимание уделяется идейным основам теории методов распознавания, их сравнительному анализу и примерам использования. Охвачен широкий круг задач распознаваня — от общих задач распознавания до локальных задач распознавания, а именно: распознавание объектов по выполняемым функциям, по составу, по структуре, по форме, по организации, по управлению. Обсуждается методика постановки и решения проблем распознавания. Исследуются вопросы влияния помех на процессы распознавания. Рассматриваются средства математического описания объектов и процессов распознавания. Описываются системы автоматизированного распознавания и диагностики.

Работа предназначена для магистров, аспирантов, докторантов, инженеров, экономистов, статистиков, вычислителей и всех тех, кто сталкивается с задачами распознавания.

ББК В161.я7

ISBN 978-966-7599-50-8

Оглавление

Ведение

Как уже отмечалось в книге 1 настоящей работы, для решения задач распознавания, возникающих при проектировании, создании и эксплуатации различных объектов, требуется проводить многочисленные исследования и расчеты, связанные с оценкой показателей, характеризующих различные свойства распознаваемого объекта, а также с оценкой оптимальной структуры распознаваемого объекта и оптимальных значений его параметров. Выполнение таких исследований с целью распознавания объекта возможно лишь в том случае, если мы располагаем *математическим описанием (математической моделью)* распознаваемого объекта и распознающего процесса.

Сложность распознаваемых объектов не позволяет строить для них «абсолютно» адекватные математические модели. Математическая модель описывает некоторый упрощенный объект или процесс, в которых представлены лишь основные явления, входящие в распознающий процесс, и лишь главные факторы, действующие на распознаваемый объект, причем все они имитируются соответствующими формальными схемами, удобными с аналитической точки зрения или в вычислительном отношении.

Какие явления считать основными и какие факторы главными— существенно зависит от назначения модели, оттого, какие исследования и распознающие процессы с ее помощью предполагается проводить. Поэтому процесс функционирования одного и того же распознаваемого объекта может получить различные математические описания в зависимости от поставленной задачи.

Для построения простой и изящной математической модели, обладающей достаточной степенью адекватности распознающему процессу, требуется обычно немалое искусство. Помимо интуиции и понимания структуры формализуемых явлений, здесь существенную роль играет знание типичных формальных схем и математических моделей, пригодных для описания различных процессов. Мы остановимся на формализации общих схем процесса функционирования распознаваемого объекта и распознающего процесса. Желая получить математическую модель, охватывающую широкий класс реальных распознаваемых объектов распознающих процессов, мы будем исходить из весьма общих предположений о характере функционирования распознаваемого объекта:

1) распознаваемый объект функционирует во *времени;* в каждый момент времени распознаваемый объект находится в одном из возможных *состояний;*
2) на вход распознаваемого объекта могут поступать *входные сигналы;*
3) распознаваемый объект способен выдавать *выходные* сигналы;
4) состояние распознаваемого объекта в данный момент времени определяется *предыдущими* состояниями и входными сигналами, поступившими в данный момент времени и ранее;
5) выходной сигнал в данный момент времени определяется состояниями распознаваемого объекта и входными сигналами, относящимися к данному и предшествующим моментам времени.

Первое из перечисленных предположений имеет очевидную цель отразить *динамический* характер процесса функционирования распознаваемого объекта в пространстве и времени, подчеркнуть, что процесс функционирования распознаваемого объекта протекает как последовательная смена состояний распознаваемого объекта под действием внешних и внутренних причин. Второе и третье — описать взаимодействие распознаваемого объекта с внешней средой. В четвертом и пятом предположениях отражаются два важных аспекта, связанных с определением реакции распознаваемого объекта на внутренние факторы и воздействия внешней среды.

С одной стороны, здесь учитывается то обстоятельство, что многим распознаваемым явлениям и процессам свойственно *последействие,* вследствие которого тенденции, определяющие поведение распознаваемого объекта в будущем, зависят не только от того, в каком состоянии находится распознаваемый объект в настоящий момент времени, но в той или другой степени от его поведения в предыдущие моменты времени. С другой стороны, отражается принцип *физической реализуемости*: распознаваемый объект не реагирует в данный момент времени на «будущие» факторы и воздействия внешней среды.

Для того чтобы придать сформулированным предположениям более точный смысл и необходимый формальный вид, используем соответствующие математические средства.

1. Множества. Отображения. Операции

Будем придерживаться следующих общепринятых обозначений.

Множества обозначаются прописными латинскими буквами X, Y, Z, T и т. д., а элементы этих множеств — соответствующими строчными буквами: x, y, z, t и т.д. Запись $x \in X$ или $x \ni X$ означает: «x является элементом множества X», «x принадлежит X» или «множество X

содержит x в качестве элемента», а $Y \subset X$ или $X \supset Y$ — «множество Y является *подмножеством* множества X». *Объединение* $X \bigcup Y$ множеств X и Y есть множество всех элементов, принадлежащих либо множеству Y, либо множеству X. *Пересечение* $X \cap Y$ множеств X и Y есть множество всех элементов, принадлежащих одновременно и множеству X и множеству Y. *Разность* $X \setminus Y$ множеств X и Y есть множество элементов, принадлежащих множеству X, но не принадлежащих множеству Y. Пустое множество обозначается Ø.

Операции объединения и пересечения множеств удовлетворяют законам *коммутативности* $X \bigcup Y = Y \bigcup X$; $X \cap Y = Y \cap X$; *ассоциативности* $X \bigcup (Y \bigcup Z) = (X \bigcup Y) \bigcup Z$; $X \bigcap (Y \bigcap Z) = (X \bigcap Y) \bigcap Z$ и *идемпотентности* $X \bigcup X = X$, $X \cap X = X$.

Пусть $\{Z_i\}_n$ — конечная совокупность множеств Z_1, Z_2, ..., Z_n и z_i — их элементы, $i = 1, 2, \ldots n$. *Прямым* или *декартовым произведением* множеств Z_1, Z_2, ..., Z_n называется множество $\hat{Z}$ всех упорядоченных последовательностей $\hat{z} = (z_1, z_2, \ldots, z_n)$. Прямое произведение обозначается $\hat{Z} = Z_1 \times Z_2 \times \ldots \times Z_n$ или $\hat{Z} = \prod(z_i \in \{Z_i\}_n / i = 1, 2, \ldots, n)$. Например, прямое произведение Z множеств X и Y есть множество всех упорядоченных пар $z = (x,y)$, где $x \in X$, $y \in Y$. Прямое произведение U множеств X, Y и Z есть множество всех упорядоченных троек $u=(x, y, z)$, где $x \in X$, $y \in Y$ и $z \in Z$. Операция прямого произведения обладает свойством ассоциативности $X \times (Y \times Z) = (X \times Y) \times Z = X \times Y \times Z$, но в общем случае не коммутативна $X \times Y \neq Y \times X$, т. е. (x, y) не всегда совпадает с (y, x). Прямое произведение $X \times X$ обозначается X^2.

В выражении $\hat{Z} = Z_1 \times Z_2 \times \ldots \times Z_n$ множества Z_i, $i = 1, 2, \ldots, n$, называются *осями прямого произведения* $\hat{Z}$, а их элементы $z_i \in Z_i$ — *проекциями* элемента $\hat{z} = (z_1, z_2, \ldots, z_n) \in \hat{Z}$ на оси Z_i.

В дальнейшем мы будем пользоваться следующей терминологией: множество $\hat{Z}$ будем называть *пространством,* его элемент $\hat{z}$ — *точкой* пространства $\hat{Z}$, а элемент $z_i \in Z_i$ — проекцией точки $\hat{z}$ на ось Z_i. Нам также потребуются обобщения понятия оси и проекции. Обозначим, например, $Z^* = Z_1 \times Z_2 \times Z_3$ и $\bar{Z} = Z_5 \times Z_6 \times \ldots \times Z_n$. Тогда $\hat{Z} = Z^* \times Z_4 \times \bar{Z}$. Пространства Z* и Z можно также рассматривать как оси пространства $\hat{Z}$. Проекциями точки $\hat{z}$ на эти оси будут соответственно $z^* = (z_1, z_2, z_3)$ и $\bar{z} = (z_5, z_6, \ldots, z_n)$. В отличие от Z* и $\bar{Z}$ исходные множества Z_i, которые в условиях данной задачи не являются прямыми произведениями каких-либо множеств, будем называть *элементарными осями* пространства $\hat{Z}$.

В трехмерном евклидовом пространстве $\hat{Z} = Z_1 \times Z_2 \times Z_3$ множества Z_1, Z_2 и Z_3 (множества точек числовой прямой) являются элементарными осями пространства $\hat{Z}$; проекции точки $\hat{z} \in \hat{Z}$ на эти оси выражаются координатами (числами) z_1, z_2 и z_3; координатные плоскости $Z_1 \times Z_2$ и $Z_2 \times Z_3$ представляют собой оси пространства $\hat{Z}$; проекции точки $\hat{z} \in \hat{Z}$ на них описываются парами чисел (z_1, z_2) и (z_2, z_3) соответственно; координатную плоскость $Z_1 \times Z_3$ также будем считать осью пространства $\hat{Z}$, хотя она и не является «сомножителем» в выражения прямого произведения; проекции точки $\hat{z} \in \hat{Z}$ на эту ось описываются парой чисел (z_1, z_3).

В общем случае, пусть $\{Z_i\}_n$ — совокупность элементарных осей пространства $\hat{Z}$ и пусть $\{Z_i\}'$ — какое-нибудь подмножество множества $\{Z_i\}_n$; тогда прямое произведение

$$\tilde{Z} = \prod (Z_i \in \{Z_i\}' / \hat{Z})$$

элементарных осей, принадлежащих подмножеству $\{Z_i\}'$, взятых в том порядке, в котором они входят в прямое произведение $\hat{Z}$, будем называть осью пространства $\hat{Z}$.

Пусть $\hat{Z}$ — пространство с совокупностью элементарных осей $\{Z_i\}_n$. Рассмотрим некоторое подмножество Z^* множества точек $\hat{z} = (z_1, z_2, \ldots, z_n)$ пространства $\hat{Z}$. Если подмножество Z^* само является пространством в том же смысле, что и пространство $\hat{Z}$, то Z^* называется *подпространством* пространства $\hat{Z}$. В частности, пусть $\overline{Z}_i \subset Z_i$, $i = 1, 2, \ldots, n$, а $\{\overline{Z}_i\}_n$ — совокупность всех множеств $\overline{Z}_i$, рассматриваемых как элементарные оси некоторого пространства $\overline{Z} = \overline{Z}_1 \times \overline{Z}_2 \times \ldots \times \overline{Z}_n$; кроме того, пусть $\{\overline{Z}_i\}'$ — подмножество множества элементарных осей $\{\overline{Z}_i\}_n$; тогда пространство

$$\overline{Z}^* = \prod (\overline{Z}_i \in \{\overline{Z}_i\}' / \overline{Z})$$

представляет собой подпространство пространства $\hat{Z}$. Легко видеть, что любая ось пространства $\hat{Z}$ является его подпространством, но не наоборот.

Пусть множество A является подмножеством множества точек пространства $Z_1 \times Z_2$. Проекцией множества A на ось Z_1 называется множество всех $z_1 \in Z_1$, для которых существует такое z_2, что $(z_1, z_2) \in A$.

Множества, между элементами которых можно установить взаимно однозначное соответствие, называются *эквивалентными*.

Рассмотрим два произвольных множества X и Y. Если указано правило, согласно которому каждому элементу $x \in X$ ставится в соответствие вполне определенный элемент $y \in Y$, говорят, что задано *отображение* $X \to Y$ множества X в множество Y или задан *оператор*

$y=F(x)$, определенный на множестве X, с областью значений, принадлежащей множеству Y.

Аналогично можно задать оператор $y = \tilde{F}(x)$, определенный на множестве $X^* \subset X$. Будем называть X^* областью определения оператора $y = \tilde{F}(x)$ в множестве X. Оператор $y = \tilde{F}(x)$ реализует отображение множества X^* в множество Y (или на множество $Y^* \subset Y$, где Y^* — область значений оператора $y = \tilde{F}(x)$ в множестве Y).

Пусть оператор $y = F(x)$ определен для всех $x \in X$ и пусть каждый элемент $x \in X$ представляет собой упорядоченную совокупность вида $x = (x_1, x_2, \ldots, x_n)$, где $x_i \in X_i$, $i = 1,2, \ldots, n$. В этом случае оператор $y = F(x)$ можно записать в виде

$$y = F(x_1, x_2, \ldots, x_n).$$

Рассмотрим прямое произведение

$$\overline{X} = X_1 \times X_2 \times \ldots \times X_n$$

как множество всех упорядоченных совокупностей $(x_1, x_2, \ldots, x_n)$. Очевидно, что в общем случае множество X является некоторым подмножеством множества $\overline{X}$. Множество X представляет собой область определения оператора $y=F(x_1, x_2, \ldots, x_n)$ в множестве $X_1 \times X_2 \times \ldots \times X_n$. Другими словами, оператор $y = F(x_1, x_2, \ldots, x_n)$ реализует отображение множества $X \subset \overline{X}$ в множество Y. В частном случае, когда множество X совпадает с множеством $\overline{X}$, оператор $y = F(x_1, x_2, \ldots, x_n)$ реализует отображение прямого произведения $X_1 \times X_2 \times \ldots \times X_n$ в множество Y.

В качестве примера рассмотрим действительную функцию

$$y = +\sqrt{1 - x_1^2 - x_2^2}$$

двух действительных переменных x_1 и x_2. Здесь множество Y^* действительных чисел y, заключенных в интервале $0<y<1$, представляет собой область значений функции в множестве действительных чисел Y. Обозначим множество действительных чисел x_1, содержащихся в интервале $-1 \leqslant x_1 \leqslant 1$, через X_1, а множество действительных чисел x_2, содержащихся в том же интервале, — через X_2. Тогда прямое произведение $\overline{X} = X_1 \times X_2$ представляет собой множество точек (x_1, x_2) плоскости, принадлежащих квадрату с вершинами (—1, —1), (—1, 1), (1,1), (1,-1). Легко видеть, что областью определения функции будет множество точек $(x_1\ x_2)$, принадлежащих кругу (радиуса 1), вписанному в упомянутый квадрат.

Случай, когда область определения совпадает с множеством $X_1 \times X_2$, можно проиллюстрировать на примере функции

$$y = \arcsin x_1 + \arcsin x_2.$$

Говорят, что на множестве M определена *бинарная алгебраическая операция,* если указан закон, по которому любой паре элементов $a, b \in M$, взятых в определенном порядке, однозначно ставится в соответствие некоторый элемент $c \in M$. В дальнейшем бинарную операцию будем обозначать $a \otimes b = c$.

Непустое множество P с бинарной операцией $a \otimes b = c$, где $a, b, c \in P$, удовлетворяющей закону ассоциативности $a \otimes (b \otimes c) = (a \otimes b) \otimes c$, называется *полугруппой.* Закон ассоциативности позволяет осуществить операцию над любым конечным числом элементов полугруппы, заданных в определенном порядке. Если, кроме того, в P справедлив закон коммутативности $a \otimes b = b \otimes a$, полугруппа называется *коммутативной.* В коммутативной полугруппе порядок элементов, участвующих в операции, не является существенным.

Полугруппа P, при дополнительном условии существования решений уравнений $a \otimes x = b$ и $y \otimes a = b$ для любых $a, b \in P$, называется *группой.* Из определения группы вытекает существование однозначно определенного единичного (нейтрального) элемента e, обладающего свойством $a \otimes e = e \otimes a = a$. Кроме того, для всякого элемента группы имеется однозначно определенный обратный элемент, обозначаемый a^{-1}, удовлетворяющий условию $a \otimes a^{-1} = a^{-1} \otimes a = e$. Обратным для a^{-1} является элемент a. Полезно также иметь в виду тождества $(a_1 \otimes a_2)^{-1} = a_2^{-1} \otimes a_1^{-1}$ и $e^{-1} = e$.

Пусть элементы группы P однозначно описываются n параметрами: $a = a(\alpha_1, \alpha_2, \ldots \alpha_n)$. Тогда параметры обратного элемента $a^{-1} = a^{-1}(\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_n)$, а также параметры результата операции $a \otimes b = c$, где $b = b(\beta_1, \beta_2, \ldots, \beta_n)$, $c = c(\gamma_1, \gamma_2, \gamma_n)$, являются функциями от параметров исходных элементов

$$\tilde{\alpha}_i = \tilde{\alpha}_i(\alpha_1, \alpha_2, \ldots, \alpha_n),$$

$$\tilde{\gamma}_i = \tilde{\gamma}_i(\alpha_1, \alpha_2, \ldots, \alpha_n; \beta_1, \beta_2, \ldots, \beta_n), \quad i = 1, 2, \ldots, n.$$

Если параметры элементов непрерывно изменяются в некоторой области, а параметры обратного элемента и элемента $c = a \otimes b$ являются непрерывно дифференцируемыми функциями от параметров исходных элементов, то группа P называется *непрерывной n-параметрической* группой.

Простейшие свойства полугрупп и групп нам понадобятся в дальнейшем при изучении распознаваемых объектов.

2. Описание непрерывных динамических распознаваемых объектов

2.1. Динамические системы

Прежде чем мы перейдем к формализации процессов функционирования распознаваемых систем общего вида, полезно кратко остановиться на хорошо изученном частном случае — системах, которые могут быть распознаны по описываемым их обыкновенным дифференциальным уравнениям. Простейшие примеры таких систем нам доставляют задачи классической механики.

Рассмотрим движение материальной точки A с массой m по прямой OX под действием силы F. Положение точки A на прямой OX в момент времени t обозначим $x(t)$. Будем считать, что функция $x(t)$ имеет непрерывные производные до второго порядка включительно. Производная $dx/dt = \dot{x}(t)$ называется скоростью движения точки A, а $d^2x/dt^2 = \dot{x}(t)$ — ускорением.

В соответствии с законом Ньютона

$$m\ddot{x} = F, \tag{1}$$

где сила F может зависеть от x и $\dot{x}$. Например, сила, действующая на пружинный маятник, определяется расстоянием от положения равновесия; для малых колебаний она пропорциональна растяжению пружины. Соответствующее дифференциальное уравнение имеет вид

$$m\ddot{x} = -\psi x, \tag{2}$$

где ψ — коэффициент жесткости пружины.

Путем выбора новых масштабов по осям t и x уравнение (2) может быть приведено к виду

$$\ddot{x} = -x \tag{3}$$

Общее решение этого уравнения

$$x = c_1 \cos t + c_2 \sin t \tag{4}$$

зависит от двух произвольных постоянных c_1 и c_2, которые определяются начальными условиями. Пусть в момент $t_0 = 0$ имеем $x = x^0$ и $\dot{x} = \dot{x}^0$. Тогда

$$c_1 = x^0, \qquad \dot{x} = -c_1 \sin t + c_2 \cos t, \qquad c_2 = \dot{x}^0.$$

Таким образом, частное решение дифференциального уравнения (3), соответствующее начальным условиям $x = x^0$ и $\dot{x} = \dot{x}^0$ в момент

$t = 0$, имеет вид

$$x(t) = x^0 \cos t + \dot{x}^0 \sin t. \qquad (5)$$

Для движения в сопротивляющейся среде сила F (1) зависит от скорости $\dot{x}$.

В достаточно общем случае распознавание движения системы с одной степенью свободы приводит к описанию распознаваемой системы дифференциальным уравнением

$$\ddot{x} = f(x, \dot{x}), \qquad (6)$$

где функция $f(x, \dot{x})$ удовлетворяет условиям теоремы существования и единственности решения.

Под числом степеней свободы мы понимаем здесь число переменных, описывающих положение движущейся материальной точки A. Поэтому движение материальной точки $A\ (x, y)$ по евклидовой плоскости $X \times Y$, где X — множество значений абсциссы x, а Y — множество значений ординаты y, можно интерпретировать как систему с двумя степенями свободы.

В качестве примера системы с $3n$ степенями свободы рассмотрим движение n материальных точек $A_i\ (x_i, y_i, z_i)$ в трехмерном евклидовом пространстве $E_3 = X \times Y \times Z$, где $X \ni x_i,\ Y \ni y_i,\ Z \ni z_i$.

Обычно задачи механики системы n материальных точек сводятся главным образом, к распознаванию движения центра инерции системы. Однако нередки случаи, когда для ответа на поставленный вопрос важно распознать относительное расположение материальных точек A_i в процессе движения системы (конфигурацию системы) и индивидуальное поведение каждой точки. Для этой цели удобно использовать так называемое *конфигурационное* пространство, размерность которого совпадает с числом степеней свободы системы. Перейдем к обозначениям $x_i = x_{3i-2},\ y_i = x_{3i-1},\ z_i = x_{3i}$, где $x_{3i-k} \in X_{3i-k},\ k = 0, 1, 2$.

Под конфигурационным пространством распознаваемой системы будем понимать прямое произведение

$$\tilde{X} = X_1 \times X_2 \times \ldots \times X_{3n}, \qquad (7)$$

точки которого представляют собой упорядоченные совокупности вида $(x_1, x_2, \ldots, x_{3n})$.

Уравнения движения системы материальных точек A_i в этих обозначениях можно записать следующим образом:

$$\ddot{x}_j = f_j(x_1, x_2, \ldots, x_{3n};\ \dot{x}_1, \dot{x}_2, \ldots, \dot{x}_{3n}),\quad j = 1, 2, \ldots, 3n. \quad (8)$$

Начальные условия:

$$x_1 = x_1^0,\ x_2 = x_2^0, \ldots, x_{3n} = x_{3n}^0;\ \dot{x}_1 = \dot{x}_1^0,\ \dot{x}_2 = \dot{x}_2^0, \ldots, \dot{x}_{3n} = \dot{x}_{3n}^0$$

в момент $t = t_0$. Любое частное решение системы дифференциальных уравнений (8) представляется совокупностью функций

$$x_j(t) = x_j(t,\ x_1^0,\ x_2^0, \ldots, x_{3n}^0;\ \dot{x}_1^0, \dot{x}_2^0, \ldots, \dot{x}_{3n}^0),\quad j = 1, 2, \ldots, 3n. \quad (9)$$

Для понимания качественной картины поведения распознаваемой системы существенную роль играет геометрическая интерпретация решений дифференциальных уравнений. Во многих случаях представляет интерес построение *графика* движения распознаваемой системы, т. е. линии в пространстве $T \times X$, описываемой соотношениями (9) при фиксированных начальных условиях, а также *траектории* движения распознаваемой системы, которая является проекцией графика движения на конфигурационное пространство.

Для распознаваемой системы с одной степенью свободы (6) конфигурационное пространство есть множество X точек x, характеризующих положение движущейся точки A на прямой OX. График движения представляет собой линию на плоскости $T \times X$. Например, для распознаваемой системы, описываемой дифференциальным уравнением (3), график движения определяется соотношением (5). Траектория движения распознаваемой системы — совокупность точек конфигурационного пространства, соответствующих положениям точки A при ее движении,— является отрезком оси OX.

Представление о характере поведения распознаваемой системы в различных ситуациях можно было бы получить из рассмотрения семейства графиков движения (или траекторий движения), соответствующих всевозможным начальным условиям. Однако такая геометрическая картина распознаваемой системы не обладает желаемой наглядностью даже в случае распознаваемой системы с одной степенью свободы. В самом деле, частное решение (5) дифференциального уравнения (3) зависит от начальных условий $(x^0, \dot{x}^0)$ в момент t_0, поэтому через каждую точку $(t_0,\ x^0)$ плоскости $T \times X$ проходит бесконечное множество линий $x = x(t)$, соответствующих различным значениям $\dot{x}^0$. Аналогично для распознаваемой системы с $3n$ степенями свободы, описываемой дифференциальными уравнениями (8), через каждую точку $(t_0,\ x_1^0,\ x_2^0, \ldots, x_{3n}^0)$ пространства $T \times X$ проходит бесконечное

множество линий $x_j = x_j(t)$, $j = 1, 2, \ldots, 3n$, соответствующих различным значениям $x_1^0, x_2^0, \ldots, x_{3n}^0$.

Изложенное позволяет подчеркнуть одну очевидную мысль, представляющую интерес в теории распознавания. А именно, знания положения распознаваемой системы $(x_1^0, x_2^0, \ldots, x_{3n}^0)$ в конфигурационном пространстве в некоторый момент времени t_0 *недостаточно* для определения ее положения в другие моменты времени. В этом можно убедиться непосредственно из (9).

Заметим, что имеется возможность более наглядного геометрического представления качественной картины поведения распознаваемой системы. От системы уравнений (8) перейдем к эквивалентной ей нормальной системе дифференциальных уравнений

$$\dot{x}_j = x_{3n+j},$$

$$\dot{x}_{3n+j} = f_j(x_1, x_2, \ldots, x_{3n};\ \dot{x}_1, \dot{x}_2, \ldots, \dot{x}_{3n}),\quad j = 1, 2, \ldots, 3n, \quad (10)$$

решение которой представляется совокупностью функций

$$x_j = x_j(t, x_1^0, x_2^0, \ldots, x_{3n}^0;\ \dot{x}_1^0, \dot{x}_2^0 \ldots, \dot{x}_{3n}^0),$$

$$\dot{x}_j = \dot{x}_j(t, x_1^0, x_2^0, \ldots, x_{3n}^0;\ \dot{x}_1^0, \dot{x}_2^0, \ldots, \dot{x}_{3n}^0),\quad j = 1, 2, \ldots, 3n. \quad (11)$$

Пусть X_j — множество значений x_j, а $\dot{X}_j$ — множество значений $\dot{x}_j$, $j = 1, 2, \ldots, 3n$. Для распознаваемой системы с $3n$ степенями свободы будем называть: упорядоченную совокупность $(x_1, x_2, \ldots, x_{3n}; \dot{x}_1, \dot{x}_2, \ldots, \dot{x}_{3n})$ — *состоянием* распознаваемой системы, прямое произведение

$$\hat{X} = X_1 \times X_2 \times \ldots \times X_{3n} \times \dot{X}_1 \times \dot{X}_2 \times \ldots \times \dot{X}_{3n} \quad (12)$$

— *пространством состояний,* а $(6n + 1)$-мерное пространство, $T \times \hat{X}$ — *фазовым пространством* распознаваемой системы.

Линия в фазовом пространстве, определяемая соотношениями (11), называется *фазовой траекторией* распознаваемой системы, а ее проекция на пространство состояний — *траекторией распознаваемой системы в пространстве состояний.*

Легко видеть, что проекция фазовой траектории или траектории распознаваемой системы в пространстве состояний на конфигурационное пространство оказывается траекторией движения системы материальных точек A_i (в конфигурационном пространстве).

В силу теоремы существования и единственности для нормальной системы дифференциальных уравнений, через каждую точку $(x_1^0, x_2^0, \ldots, x_{3n}^0; \dot{x}_1^0, \dot{x}_2^0, \ldots, \dot{x}_{3n}^0)$ $6n$-мерного пространства состояний проходит единственная траектория распознаваемой системы, причем траектории, соответствующие различным начальным состояниям

$(x_1^0, x_2^0, \ldots, x_{3n}^0; \dot{x}_1^0, \dot{x}_2^0, \ldots, \dot{x}_{3n}^0)$, не пересекаются. Это нужно понимать в том смысле, что любая точка $\hat{x} = (x_1, x_2, \ldots, x_{3n}; \dot{x}_1, \dot{x}_2, \ldots, \dot{x}_{3n})$ пространства состояний может служить начальной точкой некоторой траектории распознаваемой системы и, кроме того, все траектории распознаваемой системы, начинающиеся в точках $\hat{x}(t) \in \hat{X}$ и $\hat{x}(t + c) \in \hat{X}$, при всевозможных c совпадают между собой. В силу той же теоремы траектории распознаваемой системы в пространстве состояний не имеют точек самопересечения, в то время как траектории движения системы материальных точек A_i в конфигурационном пространстве в общем случае могут иметь точки самопересечения.

Траектории распознаваемой системы разбивают пространство состояний на *классы эквивалентности:* если точки лежат на одной траектории, то они принадлежат одному классу. Такое разбиение называется *портретом системы в пространстве состояний.* Например, для дифференциального уравнения (3) эквивалентная нормальная система уравнений имеет вид

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -x_1. \tag{13}$$

Пространство состояний представляет собой плоскость $X \times \dot{X}$, где $X \ni x$, $\dot{X} \ni \dot{x}$. Фазовые траектории задаются выражениями

$$x(t) = x^0 \cos t + \dot{x}^0 \sin t, \quad \dot{x}(t) = -x^0 \sin t + \dot{x}^0 \cos t \tag{14}$$

и представляют собой семейство винтовых линий в пространстве $T \times X \times X$, проходящих через точки $(t^0, x^0, \dot{x}^0)$. Исключив из (14) параметр t, получим уравнение траектории распознаваемой системы

$$x^2 + \dot{x}^2 = (x^0)^2 + (\dot{x}^0)^2. \tag{15}$$

Поэтому траектории распознаваемой системы, описываемой дифференциальным уравнением (3), являются концентрическими окружностями с центром в точке $x^0 = 0$; $\dot{x}^0 = 0$, а портрет этой распознаваемой системы — совокупность упомянутых окружностей. *Изображающая точка* $(x, \dot{x})$ движется по окружности в направлении часовой стрелки (при возрастании t) с единичной угловой скоростью. Функции $x(t)$ и $\dot{x}(t)$ — периодические функции. Начало координат представляет состояние покоя и является траекторией распознаваемой системы, соответствующей нулевым начальным условиям $x^0 = 0$, $\dot{x}^0 = 0$.

Портрет распознаваемой системы позволяет дать качественную характеристику ее поведения. В случае распознаваемой системы,

описываемой дифференциальным уравнением (3), он имеет простой механический смысл. В самом деле, выражение $x^2 + \dot{x}^2$ пропорционально *полной энергии* материальной точки (кинетическая энергия есть $\dot{x}^2/2$, а потенциальная — есть $x^2/2$). Таким образом, при движении распознаваемой системы сумма кинетической и потенциальной энергии остается постоянной, равной полной энергии ее в начальный момент времени t_0. Когда одно из слагаемых обращается в нуль (например, кинетическая энергия при $\dot{x} = 0$), другое принимает максимальное значение. Начало координат (точка покоя) соответствует полной энергии, равной нулю.

Естественно, что с увеличением числа степеней свободы распознаваемой системы, а вместе с ним и размерности пространства состояний — портрет распознаваемой системы в пространстве состояний оказывается все более сложным и менее наглядным. Однако теоремы существования и единственности решений, а также другие теоремы качественной теории дифференциальных уравнений дают возможность выработать формальные критерии для распознания поведения распознаваемой систем, позволяющие решать существенно важные теоретические и практические задачи распознавания. В качестве примера можно указать на теорию устойчивости движения и ее применение теории распознавания.

В результате обобщения разнообразных задач классической механики и математической физики возникло понятие *динамической системы* как некоторого абстрактного объекта, обладающего важными в теоретическом и практическом отношениях свойствами.

Пусть задана нормальная система дифференциальных уравнений

$$\dot{x}_i = f_i(x_1, x_2, \ldots, x_n), \qquad i = 1, 2, \ldots, n, \tag{16}$$

правые части которой не зависят явно от времени t (автономная система) и являются непрерывными функциями своих аргументов в некоторой замкнутой ограниченной области G n-мерного евклидова пространства. Пусть в G задана произвольная внутренняя точка $A_0(x_1^0, x_2^0, \ldots, x_n^0)$; существует решение $x_i(t) = x_i(t, t_0, x_1^0, x_2^0, \ldots, x_n^0)$ системы (16), проходящее через точку A_0 в момент t_0.

Если правые части системы (16) в области G удовлетворяют условиям Липшица

$$|f_i(x_1', x_2', \ldots, x_n') - f_i(x_1'', x_2'', \ldots, x_n'')| \leqslant L \sum_{i=1}^{n} |x_i' - x_i''|,$$

где L — константа, определяющаяся областью G, существует единственное решение, соответствующее заданным начальным условиям,

причем функции $x_i(t) = x_i(t, t_0, x_1^0, x_2^0, \ldots, x_n^0)$ являются непрерывными по всем аргументам. Поскольку правые части (16) не зависят явно от времени, решение системы можно записать в виде

$$x_i = x_i(t - t_0, x_1^0, x_2^0, \ldots, x_n^0).$$

Обозначим точку $(x_1, x_2, \ldots, x_n)$ через p, а решение системы (16), проходящее в момент t_0 через точку p_0, — символом $F(p_0, t - t_0)$. Тогда

$$p(t) = F(p_0, t - t_0).$$

Зафиксируем t_1 и положим

$$p_1 = p(t_1) = F(p_0, t_1 - t_0). \tag{17}$$

По начальным условиям t_1, p_1 составим решение

$$p(t) = F(p_1, t - t_1).$$

Очевидно, что

$$F(p_1, t - t_1) = F(p_0, t - t_0).$$

Подставляя в левую часть вместо p_1 его значение из (17), получаем

$$F[F(p_0, t_1 - t_0), t - t_1] = F(p_0, t - t_0)$$

или, обозначая $t_1 - t_0 = t'$, а $t - t_1 = t''$,

$$F[F(p_0, t'), t''] = F(p_0, t' + t''). \tag{18}$$

Функция $F(p, t)$ обладает следующими свойствами:

1) она непрерывна по обоим аргументам в области существования решения уравнений (16);

2) $F(p_0, 0) = p_0$;

3) $F[F(p_0, t'), t''] = F(p_0, t' + t'')$.

Если для любой точки $p \in G$ функция $F(p, t)$ определена для $-\infty < t < +\infty$, то она задает *однопараметрическое семейство преобразований* области G в себя, называемое *динамической системой*.

В соответствии с терминологией теории динамических систем точку $p(t)$ будем называть *состоянием* системы в момент времени t, а область G — *пространством состояний* динамической системы. Преобразование $F(p', t)$ при фиксированном p' называется движением, а совокупность точек $p(t)$ при всех t — *траекторией* динамической системы.

Во множестве преобразований $F(p, t)$ определим бинарную операцию, заключающуюся в том, что к результату преобразования с параметром t_1 применяется преобразование с параметром t_2; результат операции тоже является преобразованием (с параметром $t_1 + t_2$). Операция такого рода над преобразованиями всегда оказывается ас-

социативной. Кроме того, имеет место очевидное равенство, которое нам пригодится в дальнейшем:

$$F\{F[F(p,\ t_1),\ t_2],\ t_3\} = F\{F(p,\ t_1+t_2),\ t_3\} = F\{F(p_1,\ t_1),\ t_2+t_3\}. \quad (19)$$

Рассматриваемое семейство преобразований по отношению к введенной операции является *полугруппой.*

В полугруппе преобразований $F\ (p,\ t)$ существует обратная операция. Действительно, преобразованием, обратным $F(p,\ t)$, будет преобразование $F\ (p,\ -t)$, так как $F\ [F\ (p,\ t),\ -t] = F\ (p,\ 0)$; а нейтральным элементом — преобразование $F(p,0)$.

Таким образом, семейство преобразований $F\ (p,\ t)$ по отношению к введенной операции оказывается *группой.* Учитывая, что параметры преобразования, являющегося результатом операции, и обратного преобразования оказываются непрерывно дифференцируемыми функциями t, можно утверждать, что динамическая система есть *однопараметрическая непрерывная группа преобразований* области G в себя.

Рассмотренные выше задачи из классической механики дают примеры систем, являющихся частными случаями динамической распознаваемой системы. Возвратимся к распознаваемой системе с одной степенью свободы (3). С помощью траекторий распознаваемой системы (15) определяется для каждого $t = t_0 + \tau$ отображение пространства состояний в себя, переводящее изображающую точку $[x,\ (t_0),\ \dot{x}\ (t_0)]$ в точку $[x\ (t_0+\tau),\ \dot{x}\ (t_0+\tau)]$ на той же траектории распознаваемой системы. Это отображение является взаимно однозначным и непрерывным. Кроме того, легко проверить, что оно удовлетворяет условиям группы. Например, дифференциальное уравнение (3) определяет группу вращений плоскости $X \times \dot{X}$ относительно начала координат, см. (15).

Пусть дано произвольное метрическое пространство*' Z и семейство его отображений на себя $F\ (z,\ t)$, которое любой точке $z \in Z$ и любому вещественному числу t, $-\infty < t < +\infty$, ставит в соответствие определенную точку $F\ (z,\ t) \in Z$. (Множество Z называется метрическим пространством, если каждой паре его элементов z_1 и z_2 отнесено число $\rho(z_1,\ z_2) \geqslant 0$, называемое расстоянием и удовлетворяющее условиям: 1) $\rho(z_1,\ z_2) = 0$ в том и только в том случае, когда $z_1 = z_2$ (аксиома тождества), 2) $\rho(z_1,\ z_2) = \rho(z_2,\ z_1)$ (аксиома симметрии) и 3) $\rho(z_1,\ z_2) + \rho(z_2,\ z_3) \geqslant \rho(z_1,\ z_3)$ (аксиома треугольника).) На функцию $F\ (z,\ t)$ наложим следующие ограничения: 1) начальное условие $F\ (z,\ 0) = z$; 2) непрерывности по совокупности переменных z и t: для заданной точки z_0 и числа t_0 для любого $\varepsilon > 0$ найдется такое $\delta > 0$, что если $\rho\ (z_0,\ z) < \delta$ и $|\ t - t_0\ | < \delta$, то

$\rho\,[F\,(z,\ t),\ F\,(z_0,\ t_0)] < \varepsilon$; 3) условие группы: для любого $z \in Z$ и любых действительных t_1 и t_2

$$F\,[F\,(z,\ t_1),\ t_2] = F\,(z,\ t_1 + t_2). \qquad (20)$$

Из условий (1) и (3) следует существование обратного преобразования $F\,(z, -t)$ к преобразованию $F\ (z,\ t)$. В самом деле, $F\,[F\,(z, -t), t] = z$; тождественное преобразование $F\,(z,\ 0) = z$.

Однопараметрическая группа $F\,(z,\ t), \quad -\infty < t < +\infty$, преобразований метрического пространства Z на себя, $F\,(z,\ t) \in Z$, удовлетворяющая перечисленным условиям, называется *динамической системой* (А. А. Марков). Другими словами, под динамической системой понимается упорядоченная пара *(Z, F)* в предположении, что справедливы условия 1), 2) и 3).

Из условия 2) следует, что если начальные точки z_0 и z'_0 выбраны достаточно близкими, $\rho\,(z_0,\ z_0') < \delta$, то в течение произвольного, но фиксированного промежутка времени $-T \leqslant t \leqslant T$ расстояние между одновременными положениями движущихся точек будет оставаться меньше заданного $\varepsilon > 0$. Другими словами, для любых $\varepsilon > 0$, и $T > 0$ существует $\delta = \delta\,(\varepsilon,\ T)$ такое, что для любых $z_1,\ z_2, \in Z$ и $\rho\,(z_1,\ z_2) < \delta$ выполняется неравенство $\rho\,[F\,(z_1,\ t),\ F\,(z_2,\ t)] < \varepsilon$, при любых t, таких что $|\,t\,| \leqslant T$.

В теории динамических систем существенную роль играют так называемые *инвариантные множества.* Пусть *F(A, t)* — образ множества $A \subset Z$ при преобразованиях группы, соответствующих данному *t.* Множество *A* называется инвариантным по отношению к динамической системе (Z, *F),* если при всех преобразованиях *F (z, t)* оно переходит в себя, т. е.

$$F\,(A,\ t) \subset A, \qquad -\infty < t < +\infty. \qquad (21)$$

Рассмотрим точку $z \in A$; в силу (21)

$$F\,(z,\ t) \in F\,(A,\ t) \subset A;$$

поэтому, если точка *z* принадлежит инвариантному множеству, то в это множество входит вся траектория, определяемая точкой *z*. С другой стороны, каждая целая траектория является инвариантным множеством. Объединение и пересечение инвариантных множеств, а также замыкание инвариантного множества, представляют собой инвариантные множества данной динамической системы.

Для динамической системы характерны **три типа движений**: **покой, периодическое движение и непериодическое движение**. Точка z_0, для которой при всех значениях *t* справедливо $F\ (z_0,\ t) = z_0$, называется точкой *покоя.* Если для какого-нибудь движения существует такое τ, что $F\ (z,\ t+\tau) = F\ (z,\ t)$ при любом *t*, рассматриваемое

движение называется *периодическим,* а наименьшее τ, обладающее этим свойством, — периодом. Таким образом, имеются три существенно различных **топологических типа траекторий динамических систем: 1) точка, 2) замкнутая линия и 3) взаимно однозначный и непрерывный образ открытого отрезка.**

Можно показать, что ни одна траектория не входит в точку покоя z_0 при конечном значении t, однако если существует $\lim_{t \to \infty} F(z, t) = z_0$, то z_0 есть точка покоя.

Изучение упомянутых типов движений и условий, при которых они осуществляются, составляют основу качественной теории динамических систем. Особенно большой интерес представляют результаты (устойчивость, эргодические теоремы и др.) для динамических систем с инвариантной мерой .

В связи с изложенным поставим следующие два вопроса: 1) существуют ли динамические системы, выходящие за рамки систем, описываемых обыкновенными дифференциальными уравнениями; 2) всегда ли система, определяемая обыкновенными дифференциальными уравнениями (16), является динамической системой?

На первый вопрос дается утвердительный ответ. Это можно обосновать примерами. Ответ на второй вопрос в общем случае оказывается отрицательным. Это объясняется тем обстоятельством, что не всегда все решения уравнений (16) могут быть продолжены на $-\infty < t < +\infty$. Однако для уравнений вида (16) всегда можно подобрать динамическую систему, движения которой совпадают с решениями уравнений (16) в области их существования.

Условие группы для динамической системы (Z, *F)* предопределяет существование преобразования *F (z, —t),* обратного преобразованию *F (z, t).* Преобразование *F (z, —t)* соответствует движению системы в сторону уменьшения значений *t.* Однако в теории распознавания мы будем часто встречаться с системами, для которых упомянутое движение, по тем или другим причинам, не принадлежит множеству допустимых движений.

Поэтому в теории распознавания представляет интерес рассмотрение также систем, для которых семейство преобразований *F (z, t)* оказывается *полугруппой* по отношению к введенной в нем бинарной операции.

Определение полугрупповой динамической системы можно получить из определения А. А. Маркова, если вместо области изменения параметра $t (-\infty < t < +\infty)$ рассматривать область $(0 \leq t < +\infty)$, и условие 3) понимать как условие полугруппы.

Мы приведем также определение. Пусть T — множество значений t ($0 \leq t \leq +\infty$), Z — метрическое пространство, а F — непрерывное отображение топологического произведения $Z \times T$ в пространство Z. (Топологическое произведение двух топологических пространств X и Y есть топологическое пространство Z, определяемое следующим образом: а) множество точек пространства Z есть прямое произведение множеств точек X и Y; б) множество в Z, являющееся прямым произведением двух открытых множеств в X и в Y соответственно, представляет собой открытое множество). Упорядоченную пару (Z, F) будем называть *полугрупповой* динамической системой, если выполнены аксиомы:

1) $F(z, 0) = z$ для любой точки $z \in Z$;

2) для любых $z \in Z$ и $t_1, t_2 \in T$ имеет место равенство

$$F[F(z, t_1), t_2] = F(z, t_1 + t_2). \qquad (22)$$

Из этого определения как следствие вытекает свойство интегральной непрерывности: для любых $\varepsilon > 0$, $L > 0$ и $z_0 \in Z$ существует число $\delta > 0$, такое, что если $z \in Z$ и $\rho(z_0, z) < \delta$, то $\rho[F(z_0, t), F(z, t)] < \varepsilon$ при всех t, $0 \leqslant t \leqslant L$.

Полугрупповые динамические системы обладают рядом свойств, аналогичных свойствам обычных динамических систем. В частности, на полугрупповые динамические системы переносятся рассмотренные выше свойства инвариантных множеств (в определении инвариантного множества нужно ($-\infty < t < +\infty$) заменить на T). Сохраняются также некоторые результаты, относящиеся к трем типам движений системы (покой, периодическое движение и непериодическое движение) и соответствующим топологическим типам траекторий и т. д.

В качестве примера полугрупповой динамической системы можно привести систему, описываемую, дифференциальным уравнением

$$\frac{dx}{dt} = \frac{3}{2} x^{1/3}, \qquad x \geqslant 0, \ t \geqslant 0.$$

Существуют и некоторые дальнейшие обобщения понятия системы. Не лишено интереса, например, следующее понятие общей системы. Пусть

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_n, t), \quad i = 1, 2, \ldots, n \qquad (23)$$

— система дифференциальных уравнений, правые части которой определены при $t \geqslant 0$ и $x = (x_1, x_2, \ldots, x_n)$, принадлежащих n-мерному евклидову пространству E_n. Предполагается, что выполнены условия, при которых существует решение системы

$$x = x(t, t_0, x^0)$$

при $t \geqslant t_0 \geqslant 0$ и $x^0 \in E_n$, определенное при всех $t \geqslant t_0$ и удовлетворяющее условию $x(t, t_0, x^0) \to x^0$ при $t \to t_0 + 0$. Через точку (t_0, x^0) может проходить бесконечное множество решений системы (23). Пусть $F^t_{t_0}(x^0)$ — множество точек $x \in E_n$, принадлежащих всем упомянутым решениям в фиксированный момент времени $t \geqslant t_0$, причем $F^t_{t_0}(x^0) \to x^0$ при $t \to t_0 + 0$. $F^t_{t_0}(x)$ определяет семейство преобразований, зависящее от двух параметров t_0 и t, и сопоставляющих каждой точке $x^0 \in E_n$ некоторое множество точек $F^t_{t_0}(x^0) \in E_n$. Это семейство преобразований обладает следующими свойствами.

1. Для любого $x^0 \in E_n$ множество точек $F^t_{t_0}(x^0) \subset E_n$ не пусто и определенопри любом $t \geqslant t_0$.

2. $F^t_{t_0}(x^0) \to x^0$ при $t \to t_0 + 0$.

3. Если $x^{(1)}$ — некоторая точка множества $F^{t_1}_{t_0}(x^0)$, то при $t > t_1$ имеет место $F^t_{t_0}(x^0) = \bigcup\limits_{x^{(1)}} F^t_{t_1}(x^{(1)})$ по всем $x^{(1)} \in F^{t_1}_{t_0}(x^0)$.

Рассмотренное семейство преобразований называется *общей системой* в E_n.

Приведем также абстрактное определение общей системы.

В метрическом пространстве R задана общая система, если определено двупараметрическое семейство преобразований $F^t_{t_0}$ пространства R на себя, обладающее следующими свойствами.

1. Для любого $p \in R$ и $t_0 > 0$ определено множество $F^t_{t_0}(p) \subset R$ при $t \geqslant t_0$; $F^t_{t_0}(p)$ — не пусто.

2. $F^t_{t_0}(p) \to p$ при $t \to t_0 + 0$.

3. Для любого элемента $p_1 \in F^t_{t_0}(p)$ и $t > t_1$ определено множество $F^t_{t_1}(p)$ такое, что $F^t_{t_0}(p) = \bigcup\limits_{p_1} F^t_{t_1}(p_1)$ по всем $p_1 \in F^t_{t_0}(p)$.

При фиксированных t_0 и p **преобразование** $F^t_{t_0}(p)$ называется ***движением***, а множество всех точек, принадлежащих движению при $t \geqslant t_0$ —*траекторией* этого движения. Траектория движения $F^t_{t_0}(p)$ обозначается $F(t_0, p)$.

Для общих систем вводится понятие *инвариантного множества.*

Множество A называется инвариантным по отношению к общей системе, если из $p \in A$ следует, что $F(t, p) \subset A$, при любом $t \geqslant 0$.

Функционирование распознаваемой системы можно представить себе как совокупность двух функций времени: одна из этих функций описывает внутреннее состояние распознаваемой системы, другая —

выходной процесс распознаваемой системы, т. е. воздействие распознаваемой системы на внешнюю среду. Обе эти функции зависят, с одной стороны, от воздействия внешней среды на распознаваемую систему, т. е. входного процесса распознаваемой системы, и, с другой стороны, от воздействия случайных факторов, присущих самой распознаваемой системе. Каким же образом описать функционирование распознаваемой системы, если не принимать в расчет ни того, ни другого, т. е. допустить, что состояние распознаваемой системы изменяется в соответствии с некоторой детерминированной закономерностью, причем случайные процессы внутри распознаваемой системы и внешние воздействия отсутствуют? Легко видеть, что таким образом мы приходим к автономной детерминистической распознаваемой системе без последействия.

Оператор переходов в новое состояние для такой распознаваемой системы имеет вид

$$z(t) = H(t,\ t_0,\ z_0), \tag{24}$$

а условие однозначности — вид

$$H(t,\ t_1,\ H(t_1,\ t_0,\ z_0)) = H(t,\ t_0,\ z_0), \quad t_0 \leqslant t_1 \leqslant t. \tag{25}$$

Тождество (25) можно также интерпретировать в терминах теории меры. Именно, представим себе, что существует σ-алгебра γ подмножеств множества состояний системы Z с таким свойством. Если $t_0,\ t \in I,\ t_0 < t$, то для любого $A \in \gamma$ образ множества A состояний системы в момент t_0 при преобразовании оператором $H(t,\ t_0,\ z_0)$ снова принадлежит γ. Короче, $\{H(t,\ t_0,\ z_0),\ z_0 \in A\} \in \gamma$ при $A \in \gamma$.

Будем рассматривать меры $\mu(A)$ на подмножествах A множества Z, принадлежащих σ-алгебре γ. Тогда оператору $H(t,\ t_0,\ z_0)$ будет соответствовать оператор $\mathcal{H}(t,\ t_0,\ \mu_0)$, переводящий одни меры в другие. Именно, значением $\mathcal{H}(t,\ t_0,\ \mu_0)$ является мера μ на множествах $A \in \gamma$, определенная тем свойством, что если

$$B \in \gamma,\ A = \{z_0 : H(t,\ t_0,\ z_0) \in B\},$$

то $\mu(B) = \mu_0(A)$.

Чтобы избежать тривиальных усложнений, допустим, что оператор $H(t,\ t_0,\ z_0)$ при фиксированных t_0 и z_0 можно рассматривать как функцию $z = z(t),\ t \in I$, со значениями во множестве Z. Эту функцию называют траекторией системы. Для объяснения того, как преобразуются меры оператором $\mathcal{H}$, рассмотрим меры, сосредоточенные в конечном или счетном числе точек. Итак, допустим, что фиксированы две последовательности точек:

$z_1, z_2, \ldots, z_n, \ldots \in Z$ и $p_1 \geqslant 0,\ p_2 \geqslant 0,\ \ldots,\ p_n \geqslant 0,\ \ldots, n$ и мера μ_0 определена следующим образом:

$$\mu_0(A) = \sum_{z_n \in A} p_n.$$

Тогда, если $\mu = \mathcal{H}(t,\ t_0,\ \mu_0)$, будем иметь:

$$\mu(B) = \sum p_n, \qquad H(t,\ t_0,\ z_n) \in B. \tag{26}$$

Отсюда легко представить себе тот случай, когда меры задаются плотностями.

Важно заметить, что оператор $\mathcal{H}(t,\ t_0,\ \mu_0)$ является линейным, в отличие (в общем случае) от оператора $H(t,\ t_0,\ z_0)$:

$$\mathcal{H}(t,\ t_0,\ \alpha\mu_0 + \beta\mu_1) = \\ = \alpha\mathcal{H}(t,\ t_0,\ \mu_0) + \beta\mathcal{H}(t,\ t_0,\ \mu_1). \tag{27}$$

Это обстоятельство является отправной точкой важных методов распознавания свойств системы. Для оператора $\mathcal{H}$, действующего во множестве мер, справедлив следующий аналог тождества (25):

$$\mathcal{H}(t,\ t_1,\ \mathcal{H}(t_1,\ t_0,\ \mu_0)) = \mathcal{H}(t,\ t_0,\ \mu_0). \tag{28}$$

Введенный оператор $\mathcal{H}$ можно также рассматривать как оператор переходов в новое состояние, если состояние распознаваемой системы в момент t представлять себя как меру μ_t, определенную на измеримом пространстве $(Z,\ \gamma)$. Однако если принять такую точку зрения, то можно охватить гораздо более общую ситуацию, чем детерминированные движения $H(t, t_0, z_0)$ (то есть, не каждому оператору $\mathcal{H}$ со свойством (28) соответствует оператор H со свойством (25). Именно, представим себе, что состояния системы в момент t, т. е. $z(t)$, случайны, и описываются распределением вероятностей или, что то же, вероятностной мерой μ_t:

$$\mathbf{P}\{z(t) \in A\} = \mu_t(A), \qquad A \in \gamma. \tag{29}$$

Тогда оператор $\mu_t = \mathcal{H}(t, t_0, \mu_0)$ будет задавать распределение вероятностей состояний системы в момент t при условии, что в момент t_0 имело место распределение μ_0.

Рассмотрим вероятностную меру μ_{z_0}, целиком сосредоточенную в z_0:

$$\mu_{z_0}(A) = \begin{cases} 1, & \text{если } z_0 \in A, \\ 0, & \text{если } z_0 \overline{\in} A. \end{cases}$$

Обозначим

$$\mathcal{H}(t,\ t_0,\ \mu_{z_0}) = \mu_{t, t_0, z_0}.$$

Тогда $\mu_{t, t_0, z_0}(A)$ естественно рассматривать как условную вероятность того, что $z(t) \in A$, при условии, что $z(t_0) = z_0$. В силу линейности оператора $\mathcal{H}$ имеем

$$\mathcal{H}(t,\ t_0,\ \mu_0) = \int_Z \mu_{t, t_0, z_0}\, \mu_0(dz_0). \tag{30}$$

Более подробно, значение меры $\mathcal{H}(t,\ t_0,\ \mu_0)$ на любом множестве $A \in \gamma$ равно $\int_Z \mu_{t, t_0, z_0}(A)\, \mu_0(dz_0)$.

Допустим теперь, что $z(t),\ t \in I$, является марковским процессом. Любые конечномерные распределения случайных величин $z(t)$ (а при соответствующих аналитических условиях — и более сложные распределения — такие, как вероятность нахождения процесса в данной области в течение заданного времени) однозначно определяются начальным распределением $\mu_{t_0}(A) = \mathbf{P}\{z(t_0) \in A\}$ и марковской переходной функцией

$$P_{t, t_0, z_0}(A) = \mathbf{P}\{z(t) \in A / z(t_0) = z_0\},$$

представляющей собой при фиксированных $t,\ t_0,\ z_0$ вероятностную меру $\mu_{t,\ t_0,\ z_0}$ в Z. Если теперь задать оператор $\mathcal{H}$ во множестве мер соотношением

$$\mathcal{H}(t,\ t_0,\ \mu_0) = \int_Z \mu_{t, t_0, z_0}\, \mu_0(dz_0), \tag{31}$$

то этот оператор будет удовлетворять соотношению (28). Действительно, по уравнению Колмогорова—Чепмена, при $t_0 < t_1 < t$

$$\begin{aligned}\mathbf{P}\{z(t) \in A\} &= \iint \mathbf{P}\{z(t_0) \in dz_0\}\mathbf{P}\{z(t_1) \in dz_1 / z(t_0) = z_0\} \times \\ &\times \mathbf{P}\{z(t) \in A / z(t_1) = z_1\} = \\ &= \int \mathbf{P}\{z(t_0) \in dz_0\}\mathbf{P}\{z(t) \in A / z(t_0) = z_0\}.\end{aligned} \tag{32}$$

С учетом предыдущего последнее можно переписать так:

$$\begin{aligned}\mu_t(A) &= \iint \mu_{t_0}(dz_0)\, \mu_{t_1, t_0, z_0}(dz_1)\, \mu_{t, t_1, z_1}(A) = \\ &= \int \mu_{t_0}(dz_0)\, \mu_{t, t_0, z_0}(A).\end{aligned} \tag{33}$$

Поскольку

$$\mathcal{H}(t_1,\ t_0,\ \mu_{t_0}) = \int \mu_{t_0}(dz_0)\, \mu_{t_1, t_0, z_0},$$

$$\mathcal{H}(t,\ t_0,\ \mu_{t_0}) = \int \mu_{t_0}(dz_0)\, \mu_{t, t_0, z_0},$$

$$\mathcal{H}(t,\ t_1,\ \mu_{t_1}) = \int \mu_{t_1}(dz_1)\, \mu_{t, t_1, z_1},$$

то мы будем иметь

$$\mathcal{H}(t,\ t_0,\ \mu_{t_0}) = \mathcal{H}\left(t,\ t_1,\ \mathcal{H}(t_1,\ t_0,\ \mu_{t_0})\right), \qquad (34)$$

что соответствует формуле (28).

Итак, марковские процессы можно изучать с помощью операторов перехода, определенных на вероятностных мерах в измеримом пространстве $(Z,\ \gamma)$.

Системы с оператором перехода вида

$$H(t,\ t_0,\ z_0) = H(t - t_0,\ z_0) \qquad (35)$$

называются *динамическими системами*. Обычно динамическиесистемы рассматриваются при $I = (-\infty,\ \infty)$ либо при $I = \{\ldots,\ -2,\ -1,\ 0,\ 1,\ 2,\ \ldots\}$.

Формула (25) в случае динамической системы преобразуется к следующему виду:

$$H(t_1 + t_2,\ z_0) = H\left(t_2,\ H(t_1,\ z_0)\right). \qquad (36)$$

Это соотношение выражает собой полугрупповое свойство оператора H. Оно выглядит более изящно, если использовать обозначение

$$H_t z_0 = H(t,\ z_0).$$

Тогда (36) запишется так:

$$H_{t_1+t_2} z_0 = H_{t_2} H_{t_1} z_0 \qquad (37)$$

или еще проще:

$$H_{t_1+t_2} = H_{t_2} H_{t_1}. \qquad (38)$$

Особую роль играют динамические системы с инвариантной мерой. Обозначим

$$\mathcal{H}(t_0 + t,\ t_0,\ \mu_0) = \mathcal{H}(t,\ \mu_0). \qquad (39)$$

Напомним, что $\mathcal{H}(t,\ \mu_0)$ — мера. Предположим, что для любых $t > 0$ и некоторой меры μ

$$\mathcal{H}(t,\ \mu) = \mu. \qquad (40)$$

Тогда μ называется инвариантной мерой данной динамической системы. Если мера представляет собой распределение вероятностей марковского процесса $z(t)$, то инвариантная мера обладает следующим свойством. Если $\mu_{t_0} = \mu$, то $\mu_t = \mu$ для всех $t \geqslant t_0$. Таким образом, μ является стационарным распределением вероятностей состояний марковского процесса. Кстати, динамическими системами описываются *однородные* марковские процессы, т. е. такие, для которых

$$P_{t,\ t_0,\ z_0}(A) = P_{t+\tau,\ t_0+\tau,\ z_0}(A) \quad \text{для любого } \tau > 0.$$

Основой для распознавания динамических систем с инвариантной мерой является построение некоторого множества линейных операторов, связанных с данной динамической системой.

Рассмотрим класс K комплекснозначных функций $f(x)$, интегрируемых с квадратом относительно инвариантной меры μ, и определим в множестве K оператор $U_t\ [f]$ соотношением

$$(U_t\,[f])\,(z) = f\,[H\,(t,\ z)]. \tag{41}$$

Так, если Z — плоскость, задаваемая полярными координатами r, φ, и динамическая система задается соотношением $G_t\,[(r,\ \varphi)] = (r,\ \varphi + ct)$, то любая мера вида

$$\mu\,(A) = \iint\limits_A \sigma\,(r)\,dr\,d\varphi \tag{42}$$

будет инвариантной. Оператор U_t будет иметь такой вид:

$$(U_t f)\,(r,\ \varphi) = f\,(r,\ \varphi + ct). \quad (4.76)$$

Оказывается, что U_t — линейный оператор, т. е. $U_t\,(\alpha f_1 + \beta f_2) = \alpha U_t f_1 + \beta U_t f_2$, где f_1 и f_2 — любые элементы K, α и β — любые комплексные постоянные.

Более того, U_t является унитарным оператором в следующем смысле. Пусть f, g — функции из K. Определим скалярное произведение (f, g) формулой

$$(f,\ g) = \int f\,(x)\,\overline{g\,(x)}\,d\mu\,(x). \tag{44}$$

Тогда

$$(U_t f,\ U_t g) = (f,\ g), \tag{45}$$

что означает унитарность оператора U_t. Далее, операторы U_t образуют однопараметрическую группу: $U_{t_1 + t_2} = U_{t_1} U_{t_2}$, где, как обычно, произведение понимается как последовательное применение операторов. Это позволяет применить к анализу U_t спектральную теорию линейных операторов в гильбертовом пространстве, что, в свою очередь, приводит к доказательству эргодических теорем о поведении движений динамической системы.

Подход к теории динамических систем состоит в следующем. Динамической системой называется тройка (Z, μ, G), где Z — абстрактное множество состояний системы, G — группа операторов, μ — квазиинвариантная относительно G мера. Если элементы группы определяются числовым параметром t, т. е. $G = \{H_t, -\infty < t < \infty\}$, то элемент H_t этой группы можно интерпретировать как оператор переходов $H\,(t, z)$ рассматриваемого выше типа. В общем случае, если H — произвольный элемент G, обозначим через H_z, результат применения оператора H к элементу z множества Z. Подобным же образом, для любого множества $A \in Z$ обозначим

$$H(A) = \{Hz,\ z \in A\}.$$

Упомянутое выше определение квазиинвариантности меры μ означает:

1) если A — μ-измеримо, т. е. μ(A) определена, то также $H(A)$ μ-измеримо при любом $H \in G$;

2) $\mu(A) = 0$ в том и только в том случае, когда $\mu(HA) = 0$.

Для определенной таким образом динамической системы можно ввести семейство унитарных операторов:

$$(T_H f)(z) = \sqrt{\frac{d\mu(Hz)}{d\mu(z)}}\, f(Hz),$$

где $d\mu(Hz)/d\mu(z)$ — производная меры $\mu(Hz)$ по мере $\mu(z)$; f — любая комплексная функция, для которой $\int f^2 d\mu < \infty$ (см. [5]).

Использование результатов теории динамических систем для изучения моделей реальных объектов представляет большой теоретический и практический интерес.

2.2. Дифференциальные уравнения с запаздывающим аргументом

Рассмотренные в предыдущем параграфе динамические системы являются типичным представителем систем без последействия. Последние характеризуются тем свойством, что состояние системы в некоторый начальный момент времени t_0 полностью определяет состояния системы в любые моменты времени $t > t_0$.

Обратимся к какой-нибудь конкретной траектории динамической системы. Она описывается преобразованием (17) п.2.1.

$$p(t) = F(p_0,\ t - t_0)$$

или частным решением

$$x_i(t) = x_i(t - t_0,\ x_1^0,\ x_2^0,\ \ldots,\ x_n^0),\ i = 1,\ 2,\ \ldots,\ n,$$

системы дифференциальных уравнений (16) п.2.1 при начальных условиях $x_1 = x_1^0,\ x_2 = x_2^0,\ \ldots,\ x_n = x_n^0$ в момент t_0.

Легко видеть, что в рассматриваемом случае начальными данными, достаточными для распознавания состояния системы в любой момент времени $t > t_0$, оказываются координаты состояния системы в момент t_0. Однако отмеченная закономерность отнюдь не является всеобщей. На практике встречаются системы (так называемые системы с *последействием),* для которых начальные условия имеют более сложный вид и не могут быть исчерпаны начальным состоянием $(x_1^0,\ x_2^0,$

$\dots, x_n^{v})$ системы в момент t_0. Более конкретно: для распознавания состояний системы с последействием в моменты времени $t > t_0$ необходимо знать не только ее состояние в момент времени t_0, но и состояния системы в некоторые моменты времени $t < t_0$. Другими словами, поведение системы в будущем зависит не только от ее настоящего состояния, но и от *предыстории,* т. е. от того, каким образом система пришла в это состояние.

Важный класс систем с последействием представляют собой системы, описываемые дифференциальными уравнениями с запаздывающим аргументом. Рассмотрение таких систем начнем с простых практических примеров.

На молоточек электромагнитного прерывателя в момент времени t действуют следующие силы:

$kx(t)$ — восстанавливающая сила пружины, пропорциональная (для малых отклонений $x(t)$ от положения равновесия) ее растяжению;

$gx(t-\tau)$ — сила притяжения якоря, зависящая от положения молоточка и величины тока в цепи обмотки, нарастающей благодаря самоиндукции с запаздыванием τ;

$r\dot{x}(t)$ — сила трения, пропорциональная скорости движения молоточка.

Уравнение Ньютона имеет вид

$$m\,x(t) = -r\,\dot{x}(t) - kx(t) - qx(t-\tau). \qquad (1)$$

Автоколебания генератора с запаздывающей обратной связью описываются уравнением

$$(t) + RC\,\dot{x}(t) + x(t) = M\dot{x}(t-\tau)\,s[x(t-\tau)], \qquad (2)$$

где $x(t)$ — напряжение в резонансном контуре, а τ — запаздывание.

Аналогичный вид имеют уравнения для процесса горения жидкого ракетного топлива, уравнения автоматического регулирования с запаздыванием, некоторые уравнения математической экономики и др.

Дифференциальным уравнением n-го порядка с запаздывающим аргументом мы будем называть уравнение вида

$$x^{(n)}(t) = f[t,\ x(t),\ x(t),\ \dots,\ x^{(n-1)}(t),$$
$$x(t-\tau(t)),\ \dot{x}(t-\tau(t)),\ \dots,\ x^{(n-1)}(t-\tau(t))],$$
$$t_0 \leqslant t \leqslant T,\quad \tau(t) > 0. \qquad (3)$$

Так же как и обычные дифференциальные уравнения, уравнение (3) может быть сведено к «нормальной» системе дифференциальных уравнений первого порядка

$$x_i(t) = f_i\left[t,\ x_1(t),\ x_2(t),\ \ldots,\ x_n(t),\right.$$

$$\left.x_1(t-\tau(t)),\ x_2(t-\tau(t)),\ \ldots,\ x_n(t-\tau(t))\right],\quad i=1,\ 2,\ \ldots,\ n. \quad (4)$$

Задачи распознавания иногда приводят к уравнениям с несколькими запаздываниями, например

$$\dot{x}(t) = f\left[t,\ x(t),\ x(t-\tau_1(t)),\ \ldots,\ x(t-\tau_n(t))\right],$$

$$t_0 \leqslant t \leqslant T,\quad \tau_i(t) \geqslant 0,\quad i=1,\ 2,\ \ldots,\ n. \quad (5)$$

Рассмотрим простейшее дифференциальное уравнение с запаздывающим аргументом вида

$$\dot{x}(t) = f[t,\ x(t),\ x(t-\tau)], \quad (6)$$

где $\tau > 0$ — постоянная величина.

Чтобы подойти к основной начальной задаче для уравнения (6), перейдем к эквивалентному интегральному уравнению

$$x(t) - x(t_0) + \int_{t_0}^{t} f[t,\ x(t),\ x(t-\tau)]\,dt. \quad (7)$$

Для решения этого уравнения необходимо помимо начального значения $x(t_0)$ задать функцию $x(t)$ в полуинтервале $t_0 - \tau \leqslant t < t_0$. В самом деле, правая часть (7) будет определена для любого $t > t_0$ только в том случае, если известно значение $x(t-\tau)$. Поэтому начальные условия задаются в виде некоторой функции $x(t) = \varphi(t)$ при $t_0 - \tau \leqslant t \leqslant t_0$, которая называется *начальной функцией.*

Основная начальная задача распознавания для уравнения (6) формулируется следующим образом: определить непрерывное решение $x(t)$ для $t \geqslant t_0$ при условии, что $x(t) \equiv \varphi(t)$ для $t_0 - \tau \leqslant t \leqslant t_0$.

Если функции f и φ непрерывны и первая из них удовлетворяет условию Липшица по $x(t)$ и $x(t-\tau)$, искомое решение уравнения (6) существует и единственно. Это решение может быть найдено методом последовательного интегрирования (методом шагов). Сущность его заключается в том, что непрерывное решение $x(t)$ находится из дифференциальных уравнений без запаздывания:

$$\begin{aligned}
&\dot{x}(t) = f[t,\ x(t),\ \varphi(t-\tau)] \quad \text{при } t_0 \leqslant t \leqslant t_0+\tau,\ x(t_0) = \varphi(t_0),\\
&\dot{x}(t) = f[t,\ x(t),\ \psi_1(t-\tau)] \quad \text{при } t_0+\tau \leqslant t \leqslant t_0+2\tau,\ x(t_0+\tau) = \\
&\qquad = \psi_1(t_0+\tau),\\
&\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\dot{x}(t) = f[t,\ x(t),\ \psi_n(t-\tau)] \quad \text{при } t_0+n\tau \leqslant t \leqslant t_0 + \\
&\qquad + (n+1)\tau,\ x(t_0+n\tau) = \psi_n(t_0+n\tau),
\end{aligned} \quad (8)$$

где $\psi_i(t)$—решение уравнений (8) на интервале

$$t_0 + (i-1)\tau \leqslant t \leqslant t_0 + i\tau, \quad i = 1, 2, \ldots, n.$$

Если функция f достаточное число раз дифференцируема, то решение $x(t)$ в точках $t_0 + k\tau$ имеет непрерывную производную порядка не более чем k. В частности, в точке t_0 производная $\dot{x}(t)$ имеет в общем случае разрыв первого рода.

Пусть теперь запаздывание $\tau(t)$ будет функцией времени. По-прежнему рассмотрим простейшее дифференциальное уравнение

$$\dot{x}(t) = f\left[t, x(t), x(t-\tau(t))\right]. \tag{9}$$

Для решения основной начальной задачи в этом случае начальная функция $x(t) = \varphi(t)$ должна быть задана на так называемом *начальном множестве* B_0 значений t, которое состоит из точки $t = t_0$ и из значений разности $t - \tau(t)$ при $t_0 \leqslant t \leqslant T$, меньших, чем t_0 (решение уравнения (7) определяется для $t_0 \leqslant t \leqslant T$).

В случае нескольких запаздываний (5) начальная функция $x(t) = \varphi(t)$ задается на начальном множестве $B_0 = \bigcup_{i=1}^{n} B_0^{(i)}$, где $B_0^{(i)}$ — начальное множество, соответствующее запаздыванию $\tau_i(t)$.

Решение основной начальной задачи для переменного запаздывания и нескольких запаздываний может быть получено методом последовательного интегрирования. Это утверждение справедливо также для дифференциальных уравнений n-го порядка с запаздывающим аргументом.

Существенный интерес с точки зрения распознавания систем представляют также стохастические дифференциальные уравнения с запаздывающим аргументом.

Имеется значительное число работ, посвященных методам решения и качественным исследованиям дифференциальных уравнений с запаздывающим аргументом. С точки зрения проблематики настоящей работы наибольший интерес представляют *автономные* системы дифференциальных уравнений с запаздывающим аргументом, т. е. таких уравнений, правые части которых не зависят явно от t. Ограничимся рассмотрением простейшего дифференциального уравнения вида

$$\dot{x}(t) = f[x(t-\tau)], \tag{10}$$

где x — вектор в n-мерном евклидовом пространстве E_n; $t_0 \leqslant t < +\infty$; f — вектор-функция в предположениях, обеспечивающих существование, единственность и продолжаемость решений на всю полупрямую $t_0 \leqslant t < +\infty$.

Пусть Ф — множество начальных функций $\varphi(t)$, заданных на отрезке $S = [t_0 - \tau, t_0]$, со значениями в E_n и непрерывных на этом отрезке. Заметим, что для любых $t_1 \geqslant 0$ и $\varphi \in \Phi$ функция $\psi(t) = \varphi(t_1 + t)$ принадлежит Ф. Здесь функция $\psi(t)$—результат сдвига влево функции $\varphi(t)$ на t_1. Обозначим

$$x = x(\varphi, t) \tag{11}$$

решение уравнения (10), которое определено для всех t на полупрямой $T_r = (t_0 - \tau \leqslant t < +\infty)$ и любого $\varphi \in \Phi$, причем $x(\varphi, t) \equiv \varphi(t)$ для $t \in S$. Соотношение (11) определяет отображение $\Phi \times T_r \to E_n$ прямого произведения множеств Ф и T_r во множество точек пространства E_n. Это отображение сопоставляет каждой начальной функции $\varphi \in \Phi$ и каждому $t \in T_r$ точку x пространства E_n,

Можно показать, что решение (11) уравнения (10) обладает следующими свойствами:

1) начальное условие $x(\varphi, t) \equiv \varphi(t)$ для всех $t \in S$ и любого $\varphi \in \Phi$;

2) непрерывно по совокупности аргументов φ и t;

3) $x(\varphi, t_1 + t_2) = x(\{x(\varphi, \theta + t_1)\}_{\theta = t_1 - \tau}^{\theta = t_1}, t_2)$ для любых $\varphi \in \Phi$ и $t_1, t_2 > 0$.

Свойство 3 имеет тот смысл, что для перехода от значения $x(\varphi, t_1)$ к $x(\varphi, t_1 + t_2)$ нужно знать начальную функцию $\{x(\varphi, \theta)\}_{\theta = t_1 - \tau}^{\theta = t_1}$ на отрезке $[t_1 - \tau, t_1]$.

Решение (11) уравнения (10) определяет однопараметрическое семейство преобразований (с параметром t) множества начальных функций Ф во множество точек пространства E_n. Это обстоятельство послужило поводом для введения понятия *динамической системы с последействием.*

Пусть R — метрическое пространство; S — ограниченное замкнутое подмножество точек числовой прямой и r — его нижняя грань. Рассмотрим метрическое пространство Ф, точками которого являются непрерывные отображения φ множества S в R, с метрикой равномерной сходимости на S. Кроме того, пусть $F(\varphi, t)$— непрерывное отображение топологического произведения $\Phi \times T_r$ в R, где T_r — прямая $(r \leqslant t < +\infty)$.

Упорядоченную тройку (Φ, T_r, F) будем называть динамической системой с последействием, если выполнены следующие аксиомы.

1. Для любой $\varphi \in \Phi$ и любого $s \in S$ выполняется равенство

$$F(\varphi, s) = \varphi(s).$$

2. Для любой функции $\varphi \in \Phi$ и любого $\theta \geqslant 0$ функция $F_\theta^\varphi(s)$, определяемая равенством $F_0^\varphi(s) = F(\varphi, \theta + s)$, при любом $s \in S$ принадлежит Ф.

3. Для любых $\varphi \in \Phi$, $\theta \geqslant 0$ и $\varphi \in \Phi$, $\theta \geqslant 0$ и $t \in T_r$ меет место равенство

$$F(\varphi, \theta + t) = F(F_\theta^\psi, t).$$

Заметим, что функция $F_\theta^\varphi(s)$ представляет собой сдвиг влево на G функции $F(\varphi, \theta + s)$. При фиксированном $\varphi \in \Phi$ функция $F(\varphi, t)$ называется движением системы, параметр t — временем, множество точек $F(\varphi, t)$ пространства R при $t \in T_r$ — траекторией этого движения.

Решение дифференциального уравнения (10) удовлетворяют аксиомам 1, 2 и 3 и, следовательно, определяют систему, принадлежащую к классу динамических систем с последействием. К этому же классу относятся системы, описываемые автономным дифференциальными уравнениями (без запаздывания) n-го порядка с краевыми условиями $x(a_k) = A_k$, $k = 1, 2, \ldots, n$, системы, описываемые непрерывными на T_r функциями ψ со значениями в метрическом пространстве, если множество этих функций $\Psi \ni \psi$ удовлетворяет аксиомам 1—3, и т. д.

Для динамических систем с последействием имеются некоторые результаты, относящиеся к качественным методам. В частности, показано, что существуют динамические системы с последействием, обладающие стационарными и периодическими движениями, рассмотрены теоремы об инвариантных множествах, изучены некоторые вопросы устойчивости и т. д. Интерес в теории распознавания представляют также идеи перехода от решений дифференциальных уравнений с запаздывающим аргументом к траекториям в пространстве начальных функций и ряд других.

Пусть (Φ, T_r, F) — произвольная динамическая система с последействием. Определим отображение X топологического произведения $\Phi \times T$ в Ф, полагая $\lambda(\varphi, t) = F_t^\varphi$. Оказывается, что (Φ, λ) — полугрупповая динамическая система. В самом деле, можно показать, что:

1) $\lambda(\varphi, 0) = \varphi$,

2) отображение λ непрерывно,

3) $\lambda[\lambda(\varphi, t_1), t_2] = \lambda(\varphi, t_1 + t_2)$ для любых $\varphi \in \Phi$ и $t_1, t_2 > 0$.

Таким образом, представляется возможность переходить от данной динамической системы с последействием (Φ, T_r, F) к полугрупповой динамической системе (Φ, λ), пространством состояний которой служит множество Ф начальных функций φ. Этот переход позволяет перенести на динамические системы с

последействием некоторые результаты, относящиеся к полугрупповым динамическим системам.

Всякую полугрупповую динамическую систему *(z, F)* можно рассматривать как вырожденный случай динамической системы с последействием (Φ, T_r, F), для которой начальное множество состоит из одной точки t_0. В этом случае пространство Ф перейдет в Z, а система (Φ, T_r, F) в систему *(Z, F)*.

Динамические системы с последействием представляют самостоятельный интерес для решения многих задач распознавания. Кроме того, они являются примером, опираясь на который легче найти подходы к распознаванию систем с последействием более общего характера.

3. Конечно-разностная аппроксимация и описание процессов распознавания

Характеристики распознаваемого объекта, представляющего собой сложную систему, определяются большим числом параметров, которое может доходить до нескольких сотен, а иногда и тысяч. В процессе распознавания практически не представляется возможным учесть всю их совокупность. Как правило, результаты распознавания получаются неоднозначными и принимать решение об их пригодности приходится на основе испытаний опытных образцов.

С ростом сложности и ответственности распознаваемых объектов подобные испытания становятся слишком дорогими. Имея на вооружении современную вычислительную технику, ставится перед ЛРО задача разработки методов распознавания объектов на основе аналогового или цифрового моделирования протекающих в ней физических процессов под воздействием внешних и внутренних дестабилизирующих факторов, которые должны заменить на начальной стадии распознавания объекта дорогостоящие испытания. Анализ вариантов создаваемых распознаваемых объектов на аналоговых или цифровых моделях позволит быстро, качественно, всесторонне их распознать, выявив сильные и слабые стороны этих вариантов для более обоснованного принятия решений.

Многие распознаваемые физические явления и процессы в объектах (электрические и магнитные связи, явления теплопередачи и распределение механических деформаций) могут быть описаны с помощью системы дифференциальных уравнений в частных производных. Такое описание является математической моделью соответствующего распознающего процесса.

Часто при распознавании объектов удается воспользоваться электрическими аналоговыми моделями. Электромоделирование позволяет воспроизводить процессы, аналогичные процессам в распознаваемых объектах другой физической природы (механических, тепловых, акустических и др.). С помощью электрических моделей можно решать только прямую задачу распознавания, которую сформулируем следующим образом. Задан некоторый распознаваемый объекта (физический процесс), который описывается известными уравнениями. На этот распознаваемый объекта действуют определенные внутренние или внешние факторы (силы). Требуется распознать (установить) реакцию системы на эти факторы.

Возможны и другие формулировки задач распознавания. Задачу, когда известны реакции распознаваемого объекта и требуется распознать внешние и внутренние факторы, которые привели бы этот распознаваемый объекта к известному виду реакции, будем называть обратной задачей распознавания. С применением вычислительных средств решение обратной задачи распознавания возможно лишь путем последовательных приближений.

Задачу распознавания будем называть инверсной, когда необходимо распознать физические параметры распознаваемого объекта по заданным уравнениям процесса, известным внешним и внутренним факторам (возмущениям), реакциям распознаваемого объекта. Индуктивная задача распознавания имеет место, если необходимо составить уравнения процесса при известных свойствах элементов распознаваемого объекта, внешних и внутренних факторах и реакциях распознаваемого объекта.

3.1. Использование дифференциальных уравнений в частных производных для описания процессов в распознаваемых объектах

Основной задачей распознавания ряда объектов является распознавание полевых связей между их элементами. В теории поля все анализируемое полевое пространство рассматривают как систему, которая может быть проводником тепла, областью ограниченного пространства электростатических сил, механически деформируемой системой и т. д. Такая система является пассивной, если в ней отсутствуют источники энергии, активной — если таковые имеются. Подведение к системе энергии вызывает в ней реакцию.

В механической системе подачу энергии осуществляют в виде приложения сил или сообщения скоростей в определенных точках цепи системы. В результате элементы механической цепи приводятся в движение или деформируются.

К электрической цепи энергия подводится от источника напряжения или тока, в результате чего в цепи может происходить падение напряжения, утечка тока или возникновение паразитных электромагнитных связей.

В любом случае существуют определенные соотношения, связывающие причину и следствие и обусловленные характеристиками элементов распознаваемых объектов. Причину называют возмущением или воздействующей функцией, а следствие — реакцией распознаваемого объекта.

На рис. 1 изображена схема, показывающая взаимозависимость причины и следствия при распознавании объекта.

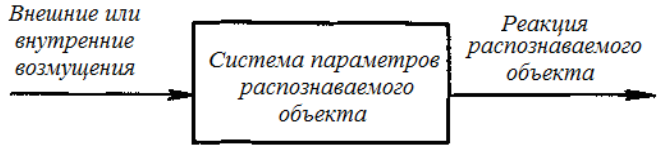


Рис. 1

Распознаваемые условия или процессы будем считать стационарными, когда внешние или внутренние возмущения практически не изменяются во времени, т. е. наблюдается состояние установившегося режима функционирования распознаваемого объекта. Если внешние или внутренние возмущения изменяются во времени, стационарность условий нарушается; такие условия или процессы называют нестационарными.

Следует отметить, что возмущение при распознавании объекта не обязательно относится к какой-либо одной точке или фрагменту. Источники возмущения могут действовать одновременно в различных частях распознаваемого объекта. Реакция также включает отклик распознаваемого объекта одновременно во всех точках (элементах) распознаваемого объекта.

При электрическом моделировании как механические, так и тепловые распознаваемые процессы можно задавать в виде структуры электрических цепей. При этом должны быть известны положения, величины и временные характеристики всех источников энергии, действующих в цепи. Аналогичный способ задания требуется и при анализе задач теории поля, но здесь необходимо знать распределенные

характеристики всех точек внутри поля и расположение границ поля, а также пространственные координаты всех возмущений. Источники возмущений могут быть на границах поля, в определенных областях внутри поля или же распределяться непрерывно в поле.

Отличие между нестационарными и стационарными условиями существенно для задач теории поля. Если реакция распознаваемого объекта является функцией времени, то задачу распознавания будем называть задачей с начальными условиями. В таких задачах для определения реакции распознаваемого объекта необходимо знать его поведение в начальный и последующие интервалы времени.

Задачу определения статического поля называют краевой задачей. При этом реакция распознаваемого объекта со временем не меняется, поэтому достаточно найти величину этой реакции и ее распределение в распознаваемом объекте внутри поля.

Обе задачи распознавания имеют одно и только одно правильное решение.

Для аналитического решения задач теории поля разработаны строгие аналитические методы (например, методы конформных преобразований), но в основном они пригодны для решений узкого круга задач распознавания. Многие из используемых способов являются эффективными для полей с очень простой геометрией (правильные плоские фигуры, цилиндр, сфера и т п.). Большинство решений представляют собой бесконечные ряды от специальных математических функций, которым часто трудно дать физическую интерпретацию.

Для решения практических задач теории поля М. Л. Быховским, В А. Вгниковым, Л. И. Гутенмахером и другими разработаны методы искусственных аналогий, которые могут быть широко использованы при решении задач распознавания.

Две системы называют аналогичными, если имеется однозначное соответствие между каждым элементом систем, а также между функциями возмущения и реакциями элементов и всей системы в целом.

Источником аналогии распознаваемых объектов является принцип, лежащий в основе большинства физических явлений — закона сохранения энергии и закона непрерывности. Перевод этих законов на математический язык приводит к характерным дифференциальным уравнениям, а так как эти законы применимы к электрическим, механическим, тепловым, акустическим и другим системам, то соответствующие дифференциальные уравнения подобны по форме. Это и является основой аналогии.

Во всех распознаваемых явлениях и процессах взаимодействий полей или анализа распределения характеристик поля величины источников возмущений и реакции в общем случае выражают через переменные, зависящие от времени и положения внутри поля. Эти зависимые переменные классифицируют как переменные первого и второго видов.

К переменным первого вида относят такие переменные, которые отличаются в различных точках пространства поля или цепи. Эти величины задают в виде разности значений в двух различных точках пространства или цепи. К таким переменным относятся: разность потенциалов электрической цепи, относительное перемещение или скорость в механике, разность концентраций примесей в полупроводниках, разность температур и т. д.

Переменными второго вида называют величины, которые не изменяются внутри некоторого исследуемого элемента. Для определения переменных этого вида не требуется измерений в двух различных точках системы. К таким переменным относятся: ток в последовательной электрической цепи, электрический или магнитный поток, силы в механике, тепловой поток и т. д. Однако при определении таких переменных необходимо знать не только собственные их величины, но и направление.

Математически переменные первого вида представляют собой разность между скалярными потенциалами в двух различных точках поля или цепи, в то время как переменная второго вида является векторной величиной. Любая физическая система, в том числе и физические процессы в элементах распознаваемого объекта, описываются переменными первого и второго вида.

Необходимо отметить, что если внешние или внутренние воздействия распознаваемого объекта заданы с помощью переменных первого вида, то параметры распознаваемого объекта, определяющие его реакцию, находят в виде переменных второго вида, и наоборот.

Параметры, связанные с конкретным распознаваемым объектом, могут быть классифицированы в соответствии с теми функциями, которые они выполняют при передаче энергии системы. В табл. приведены наиболее характерные переменные первого и второго видов, а также параметры систем для различных физических явлений

Переменные первого и второго видов, а также параметры при различных физических явлениях

Область физики	Переменная первого вида	Переменная второго вида	Параметры для		
			элемента, рассеивающего энергию (демпфер)	накопителя кинетической энергии	накопителя потенциальной энергии
Электродинамика	Напряжение	Ток			Емкость
Электростатика	Электрический потенциал	Поток напряженности электрического поля	Сопротивление	Индуктивность	Диэлектрик
Магнетизм	Потенциал м. д. с.	Магнитный поток	Магнитное сопротивление	Проницаемость	Магнитное вещество (магнитодиэлектрик)
Электромагнетизм	Электромагнитный потенциал	Магнитный поток электрического тока	Проводимость	Проницаемость	Магнитное вещество
Механика (статика)	Перемещение	Сила			Жесткость пружины
Механика (динамика)	Перемещение или скорость	Сила	Вязкое затухание	Инерция (масса)	Жесткость пружины
Упругость	Деформация	Напряжение	Вязкое трение	Инерция	Модуль Юнга
Механика жидкости	Потенциал скорости (давление)	Скорость потока	Вязкость	Инерция (плотность)	Сжимаемость
Диффузия частиц	Концентрация	Групповая скорость частиц	Коэффициент диффузии	Инерция (подвижность)	Сжимаемость
Теплопередача	Температура	Тепловой поток	Тепловое сопротивление		Теплоемкость

При математическом описании задачи распознавания полевых процессов нужно помнить о двух основных принципах — сохранения и непрерывности. Принцип сохранения означает, что возмущения или источники привносят в распознаваемый объект или выводят из него некоторое количество энергии. В любой момент времени полное количество этой энергии должно равняться первоначальному количеству, содержащемуся в распознаваемом объекте, плюс часть, добавленная (убавленная) возмущением. Принцип непрерывности распространяется на переменные второго вида. Согласно этому принципу переменная второго вида непрерывна и должна исходить от источника (внутреннего или внешнего) и возвращаться к нему или другому источнику. В общем случае один принцип подразумевает одновременно выполнение другого, и наоборот.

Различное представление этих принципов для основных областей физики приведено в таблице

Законы физики, базирующиеся на принципе сохранения энергии

Область физики	Основные законы (внутренние источники энергии отсутствуют)
Электродинамика	Первый закон Кирхгофа (сохранение заряда): $\Sigma i_n = 0$
Электростатика	Закон Гаусса [суммарный поток (входящий и выходящий) в объеме равен нулю]: $\Sigma \Phi_n = 0$
Механика (статика)	Алгебраическая сумма всех сил, действующих на точку, равна нулю· $\Sigma F = 0$
Механика (динамика)	Второй закон Ньютона (алгебраическая сумма сил равна изменению инерции): $\Sigma F - ma = 0$
Теплопередача	Первый закон термодинамики [сохранение тепловой энергии (калорий); полный тепловой поток в объеме равен энергии, накопленной в этом объеме]

При этом в задачах распознавания различных полей в распознаваемых объектах приходится решать дифференциальные уравнения в частных

производных, с помощью которых описываются распознаваемые процессы. Такие уравнения в отличие от обыкновенных дифференциальных уравнений содержат не одну переменную, а связаны с поиском функции от нескольких переменных и поэтому содержат частные производные по каждой переменной.

Дифференциальное уравнение в частных производных, описывающее многие распознаваемые процессы в пространстве, имеет вид

$$\frac{\partial}{\partial x}\left[A_1(x, y, z, \varphi, t)\frac{\partial \varphi}{\partial x}\right] + \frac{\partial}{\partial y}\left[A_2(x, y, z, \varphi, t)\frac{\partial \varphi}{\partial y}\right] + \\ + \frac{\partial}{\partial z}\left[A_3(x, y, z, \varphi, t)\frac{\partial \varphi}{\partial z}\right] = a\frac{\partial^2 \varphi}{\partial t^2} + b\frac{\partial \varphi}{\partial t} + c\varphi + d, \qquad (1)$$

где

$$a = f_1(x, y, z, \varphi, t) \geqslant 0;$$
$$b = f_2(x, y, z, \varphi, t) \geqslant 0;$$
$$c = f_3(x, y, z, \varphi, t) \geqslant 0;$$
$$d = f_4(x, y, z, \varphi, t) \geqslant 0.$$

Функции A_1, A_2 и A_3 определяют параметры распознаваемого вещества пространства. Если среда однородна (распознаваемое вещество обладает изотропными свойствами), то $A_1 = A_2 = A_3 = \text{const} > 0$; если же среда неоднородна, то $A_1 \neq A_2 \neq A_3$, причем полагают

$$0 < A_1 = \text{const}; \quad 0 < A_2 = \text{const}; \quad 0 < A_3 = \text{const}.$$

В том случае, когда A_1, A_2 или A_3 равны нулю, получаем плоскую или линейную задачу.

Значение потенциальной функции φ находят внутри некоторой области $\vee$, ограниченной поверхностью S для трехмерной и линией S для двумерной задачи. На границе поверхности S задаются граничные условия

$$\left(\alpha\varphi + \beta\frac{\partial \varphi}{\partial n}\right)_S = F, \qquad (2)$$

где α и β — заданные функции точки в граничной области; $F = (F(x, y, z, \varphi, t)$ — некоторая функция, значения которой в граничной области известны; $\partial\varphi/\partial n$ — производная искомой потенциальной функции по нормали к граничной области в рассматриваемой точке.

Если во всех точках граничной области $\beta = 0$, т. е. функция F во всех этих точках определяет значения искомой функции φ, то такие условия называют граничными условиями первого рода: $\varphi|_S = F_1$.

Если же во всех точках граничной области $\alpha = 0$, т. е. определены лишь значения производной искомой функции по нормали к этой

области, то такие условия считают граничными условиями второго рода: $\left.\frac{\partial \varphi}{\partial n}\right|_S = F_2$.

В том случае, когда имеют место смешанные варианты условий, заданные выражением граничных условий общего вида, то их называют граничными условиями третьего рода.

В зависимости от характера уравнения (1) (значения *a, b* или *c* равны нулю) определим частные случаи. Если $a = b = 0;\ c \neq 0;\ d \neq 0,$ то получим уравнения эллиптического вида. Если же $a = 0;\ b > 0;\ c \neq 0,$ то получим уравнения параболического вида. При $a > 0;\ b \neq 0;\ c \neq 0,\ d \neq 0$ уравнения называют уравнениями гиперболического вида.

Наиболее важным и часто встречающимся в теории распознавания уравнением эллиптического вида является уравнение Лапласа, описывающее стационарное состояние поля без внутренних источников и стоков. Любые установившиеся процессы теплопередачи, процессы электростатики и магнитостатики описываются этим уравнением. В общем случае уравнение Лапласа имеет вид

$$\nabla^2 \varphi = 0,$$

где ∇^2 —лапласиан, т. е. сумма вторых производных по отношению к рассматриваемым пространственным переменным. Лапласиан ∇^2 от некоторого потенциала φ в декартовых, цилиндрических и сферических координатах принимает соответственно следующие значения:

$$\begin{aligned}
\nabla^2 \varphi_{\text{д}} &= \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2};\\
\nabla^2 \varphi_{\text{ц}} &= \frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial \varphi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \varphi}{\partial \alpha^2} + \frac{\partial^2 \varphi}{\partial z^2};\\
\nabla^2 \varphi_{\text{с}} &= \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial \varphi}{\partial r}\right) + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta} \times\\
&\times \left(\sin\theta \frac{\partial \varphi}{\partial \theta}\right) + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2 \varphi}{\partial \alpha^2},
\end{aligned} \tag{3}$$

где $r = \sqrt{x^2 + y^2 + z^2}$ — расстояние точки *(x, y, z)* от начала координат (радиус-вектор); α — угол между проекцией радиус-вектора на плоскость *Oxy* и осью *Ox;* θ — угол между радиусом-вектором и осью *Oz.*

Функцию φ, удовлетворяющую уравнению Лапласа, называют гармонической. В каждой задаче, связанной с уравнением Лапласа, искомое решение выделяется из множества всех гармонических

функций определенным дополнительным условием, которое часто является краевым.

Наибольшее распространение в теории распознавания получила краевая задача Дирихле, которую формулируют для трехмерного случая так: найти функцию $\varphi(x, y, z)$, удовлетворяющую внутри замкнутой поверхности S уравнению Лапласа $\nabla^2\varphi = 0$ и принимающую на границе S заданные значения: $\varphi|_S = \tilde{\varphi}$.

Для отдельных распознаваемых объектов, представленных плоскими поверхностями, задача Дирихле может быть сформулирована в двух измерениях, например x и y (r и α — в полярных координатах). При этом уравнение Лапласа примет вид

$$\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = 0 \text{ или } \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\varphi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2\varphi}{\partial\alpha^2} = 0.$$

Тогда задача Дирихле для двумерного случая формулируется следующим образом: найти функцию $\varphi(x, y)$, удовлетворяющую внутри замкнутой поверхности S уравнению Лапласа $\nabla^2\varphi = 0$ и принимающую на границе S заданные значения: $\varphi|_S = \tilde{\varphi}$.

Эта задача имеет единственное решение.

Другим уравнением распознаваемого объекта эллиптического вида является уравнение Пуассона, представляющее собой неоднородное уравнение относительно лапласиана:

$$\nabla^2\varphi = d. \qquad (4)$$

Уравнение Пуассона описывает установившуюся систему, внутри которой равномерно распределены источники энергии. В электростатике к такому уравнению приводится задача с равномерно распределенным в поле зарядом. Это уравнение применяют при расчете распознаваемых объектов типа теплопередач, когда тепловая энергия генерируется внутри температурного поля (например, для определения распределения температуры по поверхности подложки микросхемы с источниками тепла — транзисторами и резисторами).

Граничные условия для уравнения Пуассона определяют и записывают таким же образом, как и для уравнения Лапласа.

При рассмотрении, исследовании и описании нестационарных процессов в распознаваемых объектах используют уравнение диффузии параболического вида. Этот вид уравнения, решаемый для однородной области, известен как уравнение Фурье:

$$\nabla^2\varphi = k\frac{\partial\varphi}{\partial t}, \qquad (5)$$

где k —постоянная времени диффузии, характеризующая скорость затухания процесса и перехода его в стационарный процесс, определяемая параметрами системы.

Наиболее общей является задача продвижения границы раздела двух фаз при наличии дополнительных условий на этой границе. Подобная задача относительно температуры и теплового баланса входящего и выходящего потоков и скрытой теплоты получила название задачи Стефана. Она является особенно сложной в случае, когда коэффициенты при координатах производных оказываются нелинейными вследствие зависимости теплопроводности распознаваемого вещества от температуры. Задача Стефана с подвижными границами и нелинейными коэффициентами описывается системой параболических уравнений:

$$\frac{\partial}{\partial x}\left[\lambda_1^{(V)}(x, y, z, T_V)\ \frac{\partial T_V}{\partial x}\right] + \frac{\partial}{\partial y}\left[\lambda_2^{(V)}(x, y, z, T_V)\ \frac{\partial T_V}{\partial y}\right] + \\ + \frac{\partial}{\partial z}\left[\lambda_3^{(V)}(x, y, z, T_V)\ \frac{\partial T_V}{\partial z}\right] = C_T^{(V)}(x, y, z, T_V)\ \frac{\partial T_V}{\partial t};$$

$$\frac{\partial}{\partial x}\left[\lambda_1^{(W)}(x, y, z, T_W)\ \frac{\partial T_W}{\partial x}\right] + \frac{\partial}{\partial y}\left[\lambda_2^{(W)}(x, y, z, T_W)\ \frac{\partial T_W}{\partial y}\right] + \\ + \frac{\partial}{\partial z}\left[\lambda_3^{(W)}(x, y, z, T_W)\ \frac{\partial T_W}{\partial z}\right] = C_T^{(W)}(x, y, z, T_W)\ \frac{\partial T_W}{\partial t},$$

где λ и C_r — теплопроводности и теплоемкости среды; индексы V и W означают принадлежность функций к жидкой *(V)* или твердой *(W)* фазам вещества. Если пространство, заполненное фазами V и W распознаваемого вещества, ограничено поверхностями S_1 и S_2, то задаются не только граничные, но и начальные условия

$$\varphi(x, y, z)_{t=0} = F(x, y, z, 0),$$

а на поверхности раздела фаз $\xi(t)$ —дополнительные условия: уравнение теплового баланса $\Theta_V - \Theta_W = \Theta_C$ и уравнение температурного скачка $T_V|_{\xi(t)} - T_W|_{\xi(t)} = \Delta T_0$, где $\Theta_V - \Theta_W$ — тепловые потоки, притекающие к граничной области со стороны фаз V и W; Θ_C — поток тепла, поглощаемого (выделяющегося) при фазовом превращении распознаваемого вещества; ΔT_0 — разность температур смежных точек пространства, находящихся по разные стороны граничной области $\xi(t)$.

Если в среде присутствуют распределенные источники, то, как и в уравнении Пуассона, в уравнении Фурье появляется свободный член $F(x, y, z, t)$. Если же на движение параметров внутри моделируемого поля накладывается механическое перемещение всего поля, то в уравнении Фурье появляются члены, пропорциональные первым

производным искомой функции по пространственным координатам. Таким образом, уравнение Фурье принимает вид

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial \varphi}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial \varphi}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial \varphi}{\partial z}\right) +$$

$$+ v_x \frac{\partial \varphi}{\partial x} + v_y \frac{\partial \varphi}{\partial y} + v_z \frac{\partial \varphi}{\partial z} + F(x, y, z, t) = k \frac{\partial \varphi}{\partial t};$$

где v_x, v_y, v_z — компоненты скорости механического перемещения по осям X, Y и Z; k_x, k_y, k_z — функции координат x, y и z, характеризующие анизотропность среды; k —функция координат x, y, z, являющаяся постоянной времени диффузии, которая в свою очередь зависит от параметров распознаваемого объекта.

Решение уравнения Фурье в однородной распознаваемой области, как и решение уравнения Лапласа, обладает свойством, называемым принципом максимального значения. Это свойство заключается в том, что максимальное (минимальное) значение достигается в граничной распознаваемой области или в начальный период времени. Физический смысл этого принципа очевиден: если нигде на границе температура ни в данный, ни в последующие моменты времени не превышает некоторого значения T_S , а в начальный момент для любой точки области она была бы равна (меньше) T_S, то ни в данный, ни в последующие моменты нигде в распознаваемой области температура не может быть больше T_S, так как в противном случае нарушается первый закон термодинамики.

Рассмотрим уравнения математической физики гиперболического вида — волновое уравнение и его разновидности. Волновое уравнение описывает колебательные распознаваемые процессы, происходящие в различных физических средах. В простейшем случае уравнение имеет вид

$$\nabla^2 \varphi = k \frac{\partial^2 \varphi}{\partial t^2}, \qquad (6)$$

где k — постоянная, определяемая параметрами распознаваемого объекта и характеризующая период распространения возмущений; чем меньше постоянная, тем быстрее передается возмущение от одной точки пространства к другой.

Волновое уравнение в отличие от уравнения диффузии имеет вторую производную от искомой функции по времени.

Простейший одномерный случай этого уравнения, полученный впервые Даламбером, описывает поперечные колебания струны:

$$\frac{\partial^2 \varphi}{\partial x^2} = k \frac{\partial^2 \varphi}{\partial t^2}.$$

Двумерное волновое уравнение описывает распространение колебаний по поверхности, а трехмерное — распространение волн в объеме любой сплошной распознаваемой среды: жидкости, газе, твердом теле. Для однозначного решения этого уравнения необходимо задать и граничное и начальные условия. Поскольку в уравнение входит вторая производная искомой функции по времени, следует задать два начальных условия. Одно представляет собой значение искомой функции в начальный период времени $t = 0$, т. е

$$\varphi(x, y, z, 0) = \varphi_0(x, y, z). \qquad (7)$$

В качестве второго условия выбирают начальное значение первой производной искомой функции по времени:

$$\frac{\partial \varphi}{\partial t} = \varphi'(x, y, z, 0) = \varphi_0^1(x, y, z). \qquad (8)$$

Волновое уравнение может описывать и более общий случай, когда колебательный процесс происходит в распознаваемой среде, обладающей сопротивлением. При этом происходит затухание колебания, пропорциональное первой производной по времени от искомой функции. Если же учитывается нахождение в распознаваемой среде распределенных источников энергии, то уравнение принимает вид

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = k \frac{\partial^2 \varphi}{\partial t^2} + k_1 \frac{\partial \varphi}{\partial t} + F. \qquad (9)$$

Характерной задачей распознавания, связанной с волновыми процессами, является задача нахождения собственных чисел и собственных функций распознаваемого объекта (задача Штурма—Лиувилля). Собственными числами колебательного распознаваемого объекта будем называть такие числа X, при которых существует ненулевое решение уравнения $\nabla^2 \varphi + \lambda \varphi = 0$ при $\varphi_S = 0$.

Физически же собственные числа и собственные функции описывают собственные колебания распознаваемого объекта, т. е. характеризуют собственные частоты такого объекта.

Кроме приведенного волнового уравнения гиперболического вида второго порядка на практике используют уравнение четвертого порядка, называемое бигармоническим. С помощью таких уравнений могут быть описаны колебания специального вида, главным образом применяемое в теории упругости. Для двумерной статической задачи распознавания бигармоническое уравнение записывают в виде

$$\frac{\partial^4 \varphi}{\partial x^4} + 2 \frac{\partial^4 \varphi}{\partial x^2 \partial y^2} + \frac{\partial^4 \varphi}{\partial y^4} = 0 \qquad (10)$$

или же представляют с помощью оператора $\nabla^4 \varphi = 0$.

Подобные уравнения применяют для распознавания характеристик плоских пластин, мембран, плоских оболочек и т. д., на основании выполненых расчетов. Если на распознаваемый объект, представленный мембранной, действует вынужденная сила, то однородное уравнение превращается в неоднородное, аналогичное уравнению Пуассона:

$$\nabla^4\varphi = F(x,\ y). \tag{11}$$

При расчете динамических характеристик мембран бигармоническое уравнение имеет вид

$$\nabla^4\varphi = k\frac{\partial^2\varphi}{\partial t^2}\ , \tag{12}$$

где k — коэффициент пропорциональности, характеризующий период распространения возмущений.

Для однозначного решения этого уравнения нужно задать граничные условия φ_S и $(\partial\varphi/\partial n)_S$, а также начальные условия:

$$\left.\begin{aligned} &\varphi(x,\ y,\ 0) = \varphi_0(x,\ y);\\ &\frac{\partial\varphi}{\partial t}(x,\ y,\ 0) = \varphi_0'(x,\ y). \end{aligned}\right\} \tag{13}$$

Решение однородного бигармонического уравнения представляет собой функцию, называемую бигармонической.

Бигармоническую функцию можно представить в виде двух гармонических функций. Если заданы две функции φ_1 и φ_2, гармонические в некоторой области G, то функция $\varphi = x\varphi_1 + \varphi_2$ является бигармонической в этой же области, и наоборот, каждую бигармоническую функцию φ, заданную в области G, можно представить с помощью двух заданных в этой области гармонических функций φ_1 и φ_2 в виде

$$\varphi = x\varphi_1 + \varphi_2$$

Это свойство бигармонических функций позволяет моделировать бигармоническое уравнение, т. е. уравнение четвертого порядка, методами, сходными с теми, какими моделируют гармонические уравнения Лапласа и Пуассона, являющиеся уравнениями второго порядка.

Для бигармонического уравнения, как и для волнового, собственные числа и функции находят из соотношения $\nabla^4\varphi - \lambda\varphi = 0$.

Аппроксимация уравнений Лапласа и Пуассона. Функции, которые находят в результате решений уравнений Лапласа, Пуассона, а также диффузионных и волновых уравнений, имеют непрерывный характер. Однако такие функции трудно моделировать как аналоговыми, так и цифровыми методами.

Конечно-разностная аппроксимация таких уравнений представляет собой замену системы с распределенными параметрами набором дискретных элементов таким образом, что характеристики первоначально заданного поля остаются неизменными. Процесс дискретизации является возможным при условия, что расстояние между соседними дискретами (узлами) достаточно мало.

Дискретное представление поля создает благоприятные условия для аналогового моделирования с использованием интеграторов на основе дискретных элементов, упрощающих конструкцию интегратора и способы его управления. Такие устройства обладают более высокой точностью и надежностью, чем соответствующие электрические системы.

При моделировании поля с применением ЭВМ использование конечно-разностной аппроксимации позволяет заменить дифференциальные уравнения в частных производных, описывающих распознаваеый объект, большим числом связанных между собой алгебраических уравнений. Решение задачи, приведенной к этому виду, требует только основных операций, таких, как умножение, сложение и вычитание, т. е. операций, для которых в максимальной степени приспособлены ЭВМ.

Рассмотрим простой двумерный распознаваеый объект, состоящий из прямоугольной проводящей пластины, с трех сторон к которой с помощью медных шин подведены электрические потенциалы E_1, E_2, Земля (рис. 2). Выделим из этого объекта маленький дифференциальный элемент квадратной формы (рис. 3).

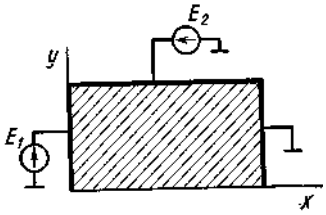


Рис. 2

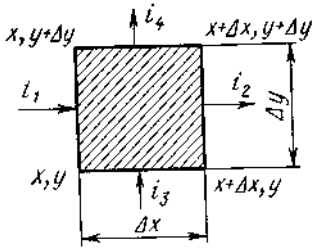


Рис. 3

Градиент напряжения на какой-либо стороне этого элемента приравняем к полному току, протекающему через соответствующую сторону и умноженному на сопротивление квадрата проводящей пластины.

Определим градиент напряжения для четырех сторон элемента, который приближенно может быть вычислен следующим образом:

$$(\partial u/\partial x)_1 = -i_1 R/\Delta y;\ (\partial u/\partial x)_2 = -i_2 R/\Delta y;$$
$$(\partial u/\partial y)_3 = -i_3 R/\Delta x;\ (\partial u/\partial y)_4 = -i_4 R/\Delta x,$$

где индексы 1, 2, 3, 4 относятся соответственно к левой, правой, нижней и верхней сторонам элемента. В этих уравнениях используются частные производные, так как число независимых переменных больше единицы.

Аналогично вычислим и скорость изменения градиента напряжения: вычтем градиенты на противоположных сторонах элемента и разделим эту разность на расстояние между сторонами квадрата. Если устремить размеры дифференциального элемента к нулю, то получим окончательное приближенное значение второй производной по координате:

$$\frac{\partial^2 u}{\partial x^2} = \frac{(\partial u/\partial x)_2 - (\partial u/\partial x)_1}{\Delta x} \text{ при } \Delta x \to 0;$$
$$\frac{\partial^2 u}{\partial y^2} = \frac{(\partial u/\partial y)_4 - (\partial u/\partial y)_3}{\Delta y} \text{ при } \Delta y \to 0.$$

Сложив эти уравнения и подставив выражения для градиента напряжения, найдем:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\frac{i_2 R}{\Delta x \Delta y} + \frac{i_1 R}{\Delta x \Delta y} - \frac{i_4 R}{\Delta x \Delta y} + \frac{i_3 R}{\Delta x \Delta y}$$

или

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{R}{\Delta x \Delta y}(i_1 - i_2 + i_3 - i_4). \qquad (14)$$

Согласно уравнению полного тока по первому закону Кирхгофа алгебраическая сумма токов в элементе должна равняться нулю: $i_1 - i_2 + i_3 - i_4 = 0$. В этом случае получаем соотношение

$$\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = 0.$$

Приведенное уравнение при заданных возмущениях, действующих на четыре границы распознаваемой проводящей пластины, полностью описывает напряжение внутри пластины. На трех сторонах пластины напряжение поддерживается постоянным по величине (E_1, E_2, Земля). Так как к нижней стороне не приложено напряжение, через ее границу не может проходить ток и, следовательно, градиент напряжения в направлении нормали к этой стороне равен нулю (ток прямо пропорционален градиенту напряжения). При этом получим следующие граничные условия:

$$
\begin{aligned}
&u = E_1 \text{ при } \begin{cases} x = 0; \\ 0 < y < M; \end{cases} \\
&u = 0 \text{ при } \begin{cases} x = L; \\ 0 < y < M; \end{cases} \\
&u = E_2 \text{ при } \begin{cases} 0 < x < L; \\ y = M; \end{cases} \\
&\partial u/\partial y = 0 \text{ при } \begin{cases} 0 < x < L; \\ y = 0. \end{cases}
\end{aligned} \tag{15}
$$

Аналогичным образом можно получить уравнение Лапласа и для трехмерного распознаваемого объекта, где дифференциальным элементом является куб из проводящего материала. Аппроксимацией каждого такого дифференциального элемента для плоской и объемной структур являются звенья резисторов.

К каждому узлу применим первый закон Кирхгофа $(\Sigma i_j = 0)$, а токи в узлах выразим через сопротивления и разности потенциалов. Например, для узла 0 и узлов 1, 2, 3 и 4 плоского дифференциального элемента (рис. 4, *а)*

$$
\frac{U_0 - U_1}{R_1} + \frac{U_0 - U_3}{R_3} - \frac{U_2 - U_0}{R_2} - \frac{U_4 - U_0}{R_4} = 0. \tag{16}
$$

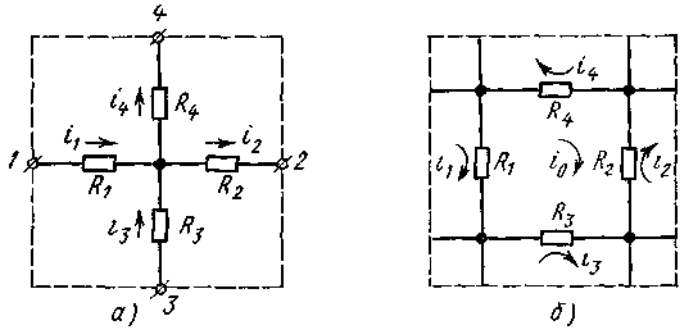


Рис. 4

Считаем, что все сопротивления дифференциального элемента одинаковы. Тогда

$$U_0 - U_1 + U_0 - U_3 - U_2 + U_0 - U_4 + U_0 = 0$$

или

$$4U_0 - U_1 - U_2 - U_3 - U_4 = 0. \tag{17}$$

Последнее уравнение является дискретным аналогом уравнения Лапласа в двумерной системе координат.

Подобным образом по второму закону Кирхгофа можно записать уравнение для контура (рис. 4, б):

$$R_1(i_1 - i_0) + R_2(i_2 - i_0) + R_3(i_3 - i_0) + R_4(i_4 - i_0) = 0.$$

Предположив, что сопротивления дифференциального элемента равны между собой, получим:

$$i_1 - i_0 + i_2 - i_0 + i_3 - i_0 + i_4 - i_0 = 0$$

или

$$i_1 + i_2 + i_3 + i_4 - 4i_0 = 0. \qquad (18)$$

Приведенное соотношение представляет собой другой вид дискретной записи уравнения Лапласа.

В случае трехмерного пространства (рис. 5, а) для узла 0 уравнение по первому закону Кирхгофа запишем в виде

$$\frac{U_1 - U_0}{R_1} + \frac{U_2 - U_0}{R_2} + \frac{U_3 - U_0}{R_3} + \frac{U_4 - U_0}{R_4} + \frac{U_5 - U_0}{R_5} + \frac{U_6 - U_0}{R_6} = 0. \qquad (19)$$

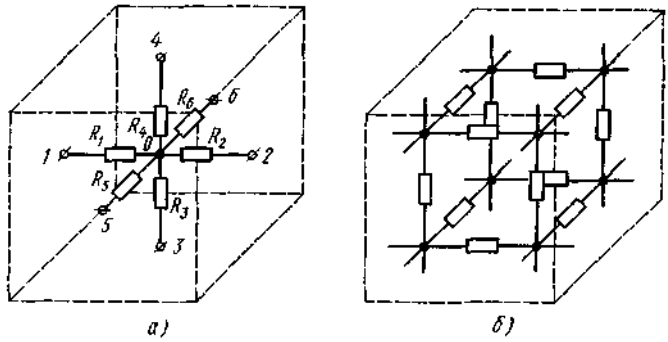


Рис. 5

Если пространство, в котором анализируется поле, изотропно, т. е. во всех направлениях сопротивление дифференциального элемента равно одному и тому же значению, то

$$U_1 - U_0 + U_2 - U_0 + U_3 - U_0 + U_4 - U_0 + U_5 - U_0 + U_6 - U_0 = 0$$

или

$$U_1 + U_2 + U_3 + U_4 + U_5 + U_6 - 6U_0 = 0. \qquad (20)$$

Найденное соотношение эквивалентно уравнению Лапласа вида

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = 0.$$

Как для плоского, так и для объемного дифференциального элемента сопротивление R является аналогом сопротивления среды соответствующей площади или объема. Полагая, что для плоской резистивной пленки прямоугольной формы

$$R = R_{\square} \frac{l}{w},$$

где $R_{\square}$ — сопротивление квадрата пленки; l — расстояние между сторонами пленки, w — ширина сторон пленки, определим следующие значения сопротивлений в дискретном аналоге:

$$\left.\begin{aligned} R_1 &= R_{\square} \frac{\Delta x}{\Delta y}; \\ R_2 &= R_{\square} \frac{\Delta x}{\Delta y}; \\ R_3 &= R_{\square} \frac{\Delta y}{\Delta x}; \\ R_4 &= R_{\square} \frac{\Delta y}{\Delta x}. \end{aligned}\right\} \qquad (21)$$

В случае трехмерного дифференциального элемента его резистивные свойства определяются удельным объемным сопротивлением среды ρ. Сопротивления резистивных элементов дискретного аналога рассчитываем следующим образом:

$$\left.\begin{aligned} &R_1 = \rho \frac{\Delta x}{\Delta y \Delta z}; \quad R_2 = \rho \frac{\Delta x}{\Delta y \Delta z}; \quad R_3 = \rho \frac{\Delta y}{\Delta x \Delta z}; \\ &R_4 = \rho \frac{\Delta y}{\Delta x \Delta z}; \quad R_5 = \rho \frac{\Delta z}{\Delta x \Delta y}; \quad R_6 = \rho \frac{\Delta z}{\Delta x \Delta y}. \end{aligned}\right\} \qquad (22)$$

Подставив эти величины в уравнение (19) и умножив его на $\rho/(\Delta x \Delta y \Delta z)$, получим новое уравнение

$$\frac{1}{\Delta x^2}(U_1 - U_0) + \frac{1}{\Delta x^2}(U_2 - U_0) + \frac{1}{\Delta y^2}(U_3 - U_0) +$$

$$+ \frac{1}{\Delta y^2}(U_4 + U_0) + \frac{1}{\Delta z^2}(U_5 - U_0) + \frac{1}{\Delta z^2}(U_6 - U_0) = 0,$$

являющееся более общим случаем конечно-разностной аппроксимации уравнения Лапласа, поскольку Δx, Δy и Δz не предполагаются равными.

Если распознаваемый объект имеет внутренние источники энергии, то распределение поля в таком объекте при установившемся режиме описывается уравнением Пуассона. Для двумерного пространства распознаваемый объект можно представить в виде проводящей пластины с равномерно распределенными по ее поверхности источниками тока при постоянном значении этого тока на единицу площади. Дифференциальный элемент распознаваемого объекта показан на рис. 6.

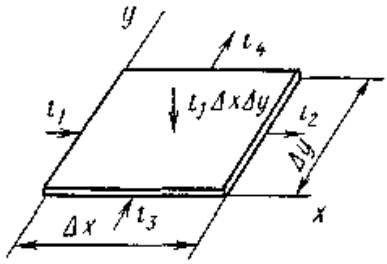


Рис. 6

В этом объекте ток i_j представляет собой возмущение среды на единицу площади моделируемой поверхности. Уравнение токов, согласно принципу сохранения,

$$i_1 - i_2 + i_3 - i_4 = -i_j \Delta x \Delta y$$

является аналогом уравнения Пуассона для двумерного пространства:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = -Ri_j. \qquad (23)$$

С помощью уравнения Пуассона $\nabla^2 \varphi = -ki_j$ можно описать не только стационарные процессы генерирования тепла при протекании электрического тока в проводящей среде, но и такие процессы, как распределение зарядов в электростатическом или магнитостатическом поле и температуры при термоядерных превращениях в реакторах,а также процессы нагрева и распределения температуры в динамических системах за счет вязкого трения в движущихся частях систем.

3.2. Конечно-разностная аппроксимация диффузионных уравнений

Уравнением математической физики, описывающим нестационарные процессы распознаваемой среды, является диффузионное уравнение, известное как уравнение теплопроводности. Выведем это

уравнение для электрической системы как распознаваемого объекта, и осуществим его конечно-разностную аппроксимацию.

Нестационарность распознаваемого процесса, описываемого уравнением диффузии, связана с апериодическим накоплением энергии в течение определенного времени, характерного для данного процесса.

Рассмотрим электрическую систему с распределенным значением сопротивления R и емкости С. Система и ее дифференциальный элемент показаны на рис. 1, *а, б.*

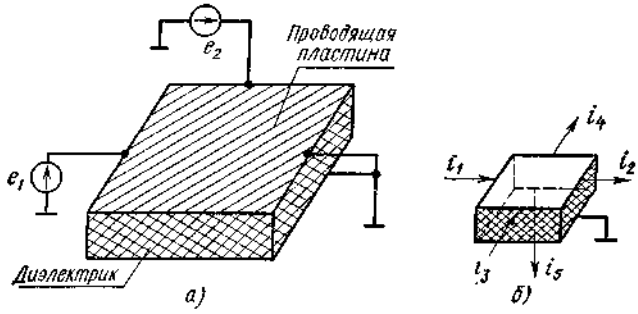


Рис. 1

Если сравнить дифференциальные элементы при моделировании уравнения Лапласа (см. рис. 3 п.3.1) и диффузионного уравнения (рис. 1, *б),* то станет ясно, что анализ последней системы может быть произведен путем изменения уравнений, соответствующих полю, описываемому уравнением Лапласа, так, чтобы учесть ток, текущий через емкость.

При рассмотрении дифференциального элемента, моделирующего уравнение Лапласа в бесконечно малой области, получим соотношение

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{R}{\Delta x \Delta y}(i_1 - i_2 + i_3 - i_4).$$

В этом уравнении полагали сумму токов равной нулю. Для уравнения диффузии эта сумма не равна нулю, а в соответствии с принципом непрерывности

$$i_1 - i_2 + i_3 - i_4 = i_5.$$

Тогда

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \left(\frac{R}{\Delta x \Delta y}\right) i_5.$$

Ток

$$i_5 = C\Delta x\Delta y \frac{\partial u}{\partial t}.$$

Если в предыдущее соотношение подставить полученное значение для i_5, то

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{R}{\Delta x\Delta y} C\Delta x\Delta y \frac{\partial u}{\partial t} = RC \frac{\partial u}{\partial t}. \quad (1)$$

Выведенное уравнение — есть уравнение диффузии в двумерной системе координат.

Наглядное представление трехмерного электрического поля, включающего сопротивление и емкость, затруднительно. Однако, сопоставляя аналогичные уравнения Лапласа и диффузионные уравнения для электрической системы, легко найти, что уравнение диффузии в трехмерном случае будет иметь вид

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = RC \frac{\partial u}{\partial t}, \quad (2)$$

где R — сопротивление между противоположными сторонами единичного куба, Ом·м; C — емкость, характеризующая накопитель потенциальной энергии моделируемой системы, Ф/м^3. Электрические аналоги дифференциального элемента для плоской и объемной конструкции приведены на рис. 2, *а, б.*

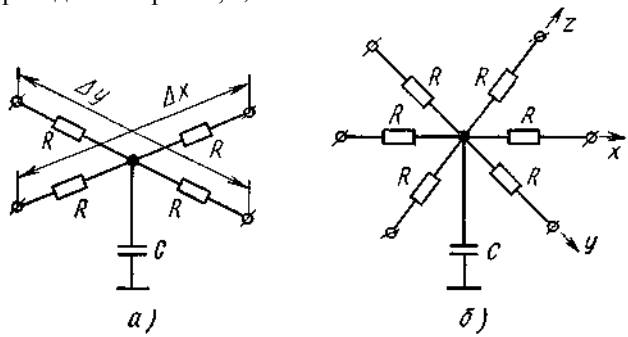
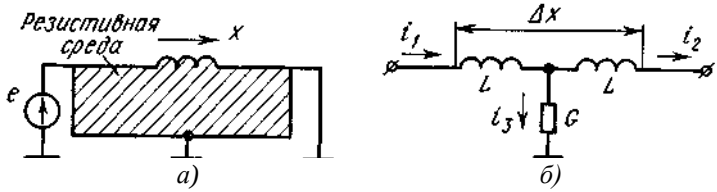


Рис. 2

Граничными условиями являются уравнения типа (15 п.3.1.), а начальное условие $u(x, y, z, 0)$ позволяет, определить напряжение во всех точках внутри поля в начальный момент времени.

В распознаваемых объектах, где в качестве накопителя энергии применяют элемент, накапливающий, главным образом, кинетическую энергию, дифференциальный элемент и одномерный аналог электрической системы имеют вид, показанный на рис. 3, *а, б.* При

этом предполагают, что проводник характеризуется некоторой погонной индуктивностью L(Гн/м) и шунтирующей погонной проводимостью G(См/м).



а) *б)*

Рис. 3

По закону Кирхгофа для дифференциального элемента рис. 3, *б* справедливо выражение

$$i_1 - i_2 - i_3 = 0.$$

Изменение тока вдоль проводника $-\partial i/\partial x = Gu$.

В свою очередь известно, что $u = L\dfrac{\partial i}{\partial t}$. Тогда градиент потенциала $-\dfrac{\partial u}{\partial x} = L\dfrac{\partial i}{\partial t}$.

Продифференцируем соотношение для $\partial i/\partial x$ по времени *t*, а соотношение $\partial u/\partial x$ по координате *x:*

$$-\frac{\partial^2 i}{\partial x \partial t} = G\frac{\partial u}{\partial t}; \qquad -\frac{\partial^2 u}{\partial x^2} = L\frac{\partial^2 i}{\partial x \partial t}.$$

В результате соответствующих подстановок получим окончательный результат в виде равенства $\dfrac{\partial^2 u}{\partial x^2} = LG\dfrac{\partial u}{\partial t}$. Для решения этого уравнения кроме граничных условий необходимо задать начальные условия для момента времени $t = 0$.

В общем случае подобные уравнения решают при граничных условиях, аналогичных условиям для уравнения Лапласа: $u = k_1$; $\partial u/\partial n = k_2$, где k_1 и k_2 — значения источников переменных первого и второго видов; n — направление нормали к граничной области.

Начальные условия должны характеризовать состояние потенциальной и кинетической энергии в распознаваемом объекте в начальный момент времени на накопителях этой энергии. Если в распознаваемом объекте есть источники, накапливающие потенциальную энергию, то для этих элементов необходимо задать начальное распределение потенциальной энергии $u(x, y, z, 0)$. Если в распознаваемом объекте в качестве источников, запасающих энергию, применяют источники

кинетической энергии, то в выражения для начальных условий вводят начальную скорость изменения этой энергии

$$\partial u(x, y, z, 0)/\partial t.$$

В каждом распознаваемом объекте может быть только один источник накопления либо потенциальной, либо кинетической энергии. Таким образом, в модели распознаваемого объекта будет всего два вида элементов: накопитель энергии и демпфер, рассеивающий эту энергию. При этом энергия не может изменяться скачком и мгновенно переходить из одного вида в другой. В распознаваемых объектах, описываемых уравнениями диффузии, энергия изменяется непрерывно во времени и это изменение определяется постоянной времени τ. Так для электрического распознаваемого объекта $\tau = RC = LG$, т. е. постоянная времени распознаваемого объекта зависит от параметров распознаваемого объекта.

В отличие от распознаваемых объектов, описываемых уравнением диффузии, в распознаваемых объектах, поведение поля которых подчиняется уравнениям Лапласа или Пуассона, возможен переход из одного состояния в другое скачком, мгновенно.

Очевидно, что если существуют стационарные условия, т. е. ни один из источников возмущения не является функцией времени, или прошло достаточно времени от начала процесса изменения источника возмущения, то член $\partial u/\partial t$ в уравнении диффузии исчезает и $\nabla^2 u$ становится равным нулю. Поэтому уравнение Лапласа можно рассматривать как предельный или вырожденный случай уравнения диффузии.

Аппроксимация уравнения диффузии является сложной задачей и распадается на несколько простых операций: аппроксимацию функций $u(x, y, z, t)$ и $\partial u(x, y, z, t)/\partial t$; аппроксимацию дифференциального оператора ∇^2; аппроксимацию граничных условий.

Так как все пространство для анализа поля в распознаваемых объектах при конечно-разностной аппроксимации разбивается некоторой квадратной или прямоугольной сеткой на дифференциальные элементы, то исследуемые потенциальные функции можно определить на множестве дискретных точек, являющихся узлами этих дифференциальных элементов. Координаты таких точек представляют собой последовательности целых чисел. Совокупность такого рода точек называют математической сеткой, на которой задана функция, а саму функцию — сеточной функцией.

В частном случае, когда рассматривают функцию одного аргумента, сетка, на которой определена эта функция, представляет собой натуральный ряд чисел. Такие функции называют импульсными.
Координаты пространственной сетки имеют следующие значения:

$$t = m\Delta t;\ x = n\Delta x;\ y = k\Delta y \text{ и } z = l\Delta z,$$

где

$n = 0, \pm 1, \pm 2, \ldots;\ k = 0, \pm 1, \pm 2, \ldots;\ l = 0, \pm 1, \pm 2, \ldots;$
$m = 0, 1, 2, \ldots;\ \Delta t, \Delta x, \Delta y, \Delta z$ — шаги во времени и пространстве.

Сетку называют равномерной, если шаги по всем координатам равны между собой. Для более точных исследований потенциальных функций иногда используют полуцелые координаты:

$$t = (m + 0{,}5)\,\Delta t;\ x = (n \pm 0{,}5)\,\Delta x;$$

$$y = (k \pm 0{,}5)\,\Delta y;\ z = (l \pm 0{,}5)\,\Delta z.$$

Сеточная функция аппроксимирует некоторую непрерывную функцию, если значения функции на сетке и соответствующие значения сеточной функции совпадают.

При расчете линейной, плоской или объемной системы сетку строят согласно рис. 4, *а—в.*

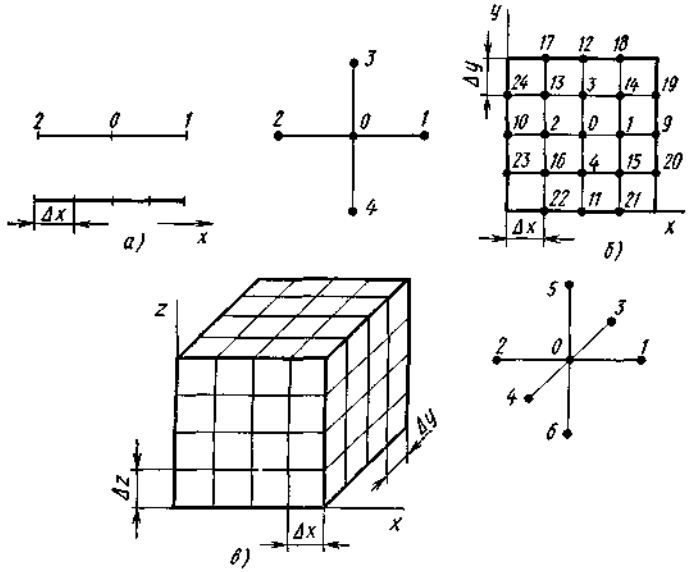


Рис. 4

Каждую ячейку сетки представляют соответствующим дифференциальным элементом, отражающим свойство среды или поля. На рис. 4, *а—в* дифференциальный элемент для среды с одной, двумя и тремя координатами представлен обобщенно в виде соответственно двух-, четырех- или шестилучевой звезды.

Для каждого такого элемента согласно (17) и (20) п.3.1 соотношение $\nabla^2 u$ записывают в виде

$$\nabla^2 u \approx u_1 + u_2 - 2u_0;$$

$$\nabla^2 u \approx u_1 + u_2 + u_3 + u_4 - 4u_0;$$

$$\nabla^2 u \approx u_1 + u_2 + u_3 + u_4 + u_5 + u_6 - 6u_0.$$

При аппроксимации производной по времени в уравнениях для частных производных символ t является еще одной независимой переменной, поэтому операции конечно-разностной аппроксимации для этой производной такие же, как и для пространственных переменных.

Процесс дискретизации времени осуществляют также с помощью масштабной сетки, имеющей шаг Δt.

Рассмотрим пример дискретизации одномерного поля в нестационарном режиме (рис 5, *а).*

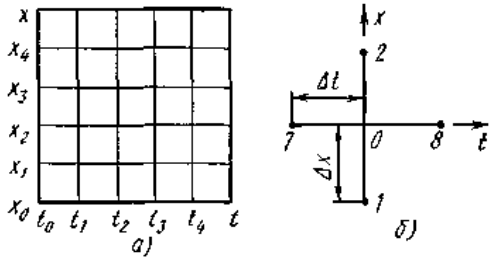


Рис. 5

По одной оси отложим номера узлов сетки, определяющих значения распределения потенциала вдоль координаты, а по другой — временные интервалы, связанные с соответствующими значениями потенциальной функции.

Первая производная от потенциальной функции по времени может быть аппроксимирована тремя способами (рис. 5, *б):*

$$\left(\frac{\partial u}{\partial t}\right)_{0-8} \approx \frac{u_8 - u_0}{\Delta t} \text{ (разность вперед);}$$

$$\left(\frac{\partial u}{\partial t}\right)_{0-7} \approx \frac{u_7 - u_0}{\Delta t} \text{ (разность назад);}$$

$$\left(\frac{\partial u}{\partial t}\right)_0 \approx \frac{u_8 - u_7}{2\Delta t} \text{ (среднее из разностей вперед и назад).}$$

Дискретизация временной переменной приводит к существенным трудностям вычисления из-за необходимости проведения анализа устойчивости (неустойчивости) вычисления.

Поэтому решение полностью дискретных аппроксимаций нестационарных задач сложнее и требует более глубокого изучения природы конечно-разностных уравнений, чем решение стационарных задач.

3.3. Конечно-разностная аппроксимация волновых уравнений

Третьим важным уравнением математической физики, описывающим колебательные процессы в распознаваемом объекте, является волновое уравнение. Как и в предыдущих случаях, рассмотрим аппроксимацию этого вида уравнений на примере электрической системы. Наглядно представить такую систему в двух- или трех-| мерном пространстве сложно, поэтому определим свойства волнового уравнения на примере одномерной электрической системы.

На рис. 1, *а* показана одномерная электрическая система, представляющая собой распределенную относительно Земли индуктивность, изолированную от нее диэлектрической средой.

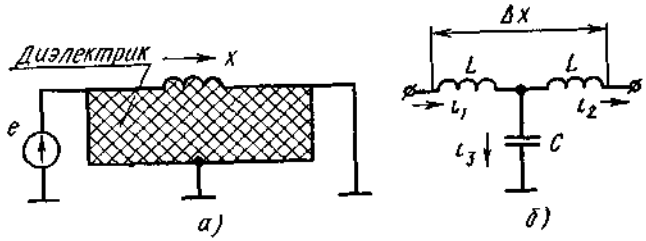


Рис. 1

На рис. 1, *б* приведена модель дифференциального элемента рассматриваемой системы.

Из курса электротехники известно, что градиент напряжения вдоль проводника $-\frac{\partial u}{\partial x} = L\frac{\partial i}{\partial t}$, а градиент тока

$$-\frac{\partial i}{\partial x} = C\frac{\partial u}{\partial t}.$$

Продифференцируем первое соотношение по переменной *x,* а второе — по времени *t:*

$$-\frac{\partial^2 u}{\partial x^2} = L\frac{\partial^2 i}{\partial x\,\partial t};\quad -\frac{\partial^2 i}{\partial x\,\partial t} = C\frac{\partial^2 u}{\partial t^2}.$$

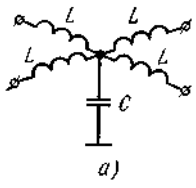
В результате соответствующих подстановок второго из полученных соотношений в первое получим:

$$\frac{\partial^2 u}{\partial x^2} = LC\frac{\partial^2 u}{\partial t^2}. \qquad (1)$$

Решение этого уравнения возможно лишь при задании граничных условий (для данной задачи $x = l$ и $x = 0$, где l — длина моделируемой системы), а также двух видов начальных условий, характеризующих начальный запас кинетической и потенциальной энергий. Для двумерного и трехмерного случаев (рис. 2, *а, б)* волновые уравнения имеют вид:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = LC\frac{\partial^2 u}{\partial t^2};$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = LC\frac{\partial^2 u}{\partial t^2}.$$



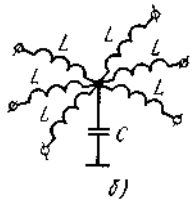


Рис. 2

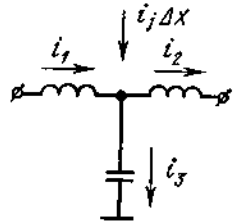


Рис. 3

Наличие двух типов накопителей энергии в таких системах подразумевает взаимный обмен этой энергией, что и определяет колебательный характер процесса, описываемого волновыми уравнениями. Если в распознаваемом объекте имеются равномерно распределенные внутренние источники энергии (рис. 3), то в уравнение вводят дополнительный член:

$$\nabla^2 u = k_1\frac{\partial^2 u}{\partial t^2} - k_2 i_j,$$

где k_1 и k_2 — параметры волнового процесса, определяемые характеристиками распознаваемого объекта.

Часто при решении задач распознавания приходится иметь дело с затухающим колебательным процессом (затухающие механические колебания в распознаваемых объектах, затухающие электромагнитные колебания в линиях связи между блоками распознаваемого объекта и т. д.). Дифференциальный элемент такой одномерного распознаваемого объекта можно представить в виде электрической модели, приведенной на рис. 4.

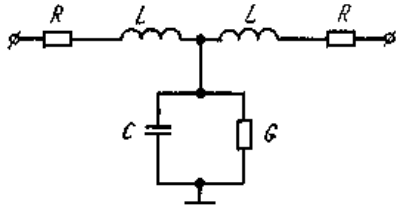


Рис. 4

Полагая, что градиенты напряжения и тока в рассматриваемой цепи соответственно

$$-\frac{\partial u}{\partial x} = iR + L\frac{\partial i}{\partial t} \quad \text{и} \quad -\frac{\partial i}{\partial x} = uG + C\frac{\partial u}{\partial t},$$

получим уравнение для этой модели в виде

$$\frac{\partial^2 u}{\partial x^2} = LC\frac{\partial^2 u}{\partial t^2} + (LG + RC)\frac{\partial u}{\partial t} + RGu. \qquad (2)$$

Волновое уравнение электрической системы с затуханием для одномерного случая называют телеграфным уравнением, оно является более общей модификацией уравнений волнового типа.

Дискретизация подобных уравнений с помощью конечно-разностных аппроксимаций аналогична дискретизации диффузионных уравнений. Лишь вторая производная по времени, например для рис. 5, *а,*п.3.2 аппроксимируется по формуле

$$\left(\frac{\partial^2 u}{\partial t^2}\right)_0 \approx \frac{u_7 + u_8 - 2u_0}{\Delta t^2}. \qquad (3)$$

Для тщательного анализа различных полей в распознаваемых объектах можно воспользоваться более сложными аппроксимациями, предложенными Бикли, в которых учитываются потенциалы более чем в двух соседних точках сеточной модели:

$$\frac{\partial u}{\partial x} \approx \frac{1}{2\Delta x}(-3u_0 + 4u_{+1} - u_{+2});$$

$$\frac{\partial u}{\partial x} \approx \frac{1}{6\Delta x}(u_{-2} - 6u_{-1} + 3u_0 + 2u_{+1});$$

$$\frac{\partial u}{\partial x} \approx \frac{1}{6\Delta x}(-11u_0 + 18u_{+1} - 9u_{+2} + 2u_{+3});$$

$$\frac{\partial u}{\partial x} \approx \frac{1}{24\Delta x}(-50u_0 + 96u_{+1} + 32u_{+3} - 72u_{+2} - 6u_{+4});$$

$$\frac{\partial u}{\partial x} \approx \frac{1}{6\Delta x}(-2u_{-1} - 3u_0 + 6u_{+1} - u_{+2});$$

$$\frac{\partial u}{\partial x} \approx \frac{1}{24\Delta x}(-6u_{-1} - 20u_0 + 36u_{+1} - 12u_{+2} + 2u_{+3}).$$

Вторая производная аппроксимируется путем вычитания разности назад из разности вперед с последующим делением на шаг сетки по формулам:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{\Delta x^2}(2u_0 - 5u_{+1} + 4u_{+2} - u_{+3});$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{24\Delta x^2}(35u_0 - 104u_{+1} + 114u_{+2} - 56u_{+3} + 114u_{+4});$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{24\Delta x^2}(-u_{-2} + 16u_{-1} - 30u_0 + 16u_{+1} - u_{+2});$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{720\Delta x^2}(4u_{-3} - 56u_{-2} + 540u_{-1} - 980u_0 + 540u_{+1} - 54u_{+2} + 4u_{+3}).$$

Отличительной особенностью механических колебаний является возникновение в системах связанных типов волн. Такие колебания описываются уравнениями бигармонического типа

$$\frac{\partial^4 u}{\partial x^4} + 2\left(\frac{\partial^4 u}{\partial x^2 \partial y^2}\right) + \frac{\partial^4 u}{\partial y^4} = k\frac{\partial^2 u}{\partial t^2}. \qquad (4)$$

Специфика в аппроксимации этого вида уравнений состоит в дискретизации $\nabla^4 u$. Соотношение дискретизации этой части уравнения получается в результате двойной аппроксимации $\nabla^2 u$ и $(\nabla^2)^2 u$, после чего осуществляют аппроксимацию члена со смешанными производными. Аппроксимацию второй производной по времени выполняют известными методами

Рассмотрим расчет двумерной сетки (см. рис. 4, *б)* п.3.2. Для тех номеров узлов, которые нанесены на этой сетке, получим:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_1 \approx \frac{u_9 + u_4 - 2u_1}{\Delta x^2} \equiv m_1;$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_2 \approx \frac{u_0 + u_{10} - 2u_2}{\Delta x^2} \equiv m_2;$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_0 \approx \frac{u_1 + u_2 - 2u_0}{\Delta x^2} \equiv m_0.$$

(Номера *5* и *6*, а также *7* и *8* использованы в других сетках для обозначения вертикальной составляющей (оси *z*) и шкалы времени).
Тогда четвертая производная по *x* примет вид

$$\left(\frac{\partial^4 u}{\partial x^4}\right)_0 \approx \left(\frac{\partial^2 m}{\partial x^2}\right) \approx \frac{m_2 + m_1 - 2m_0}{\Delta x^2}$$

или

$$\left(\frac{\partial^4 u}{\partial x^4}\right)_0 \approx \frac{1}{\Delta x^4}(u_{10} + u_9 - 4u_2 - 4u_1 + 6u_0).$$

Аналогично найдем

$$\left(\frac{\partial^4 u}{\partial y^4}\right) \approx \frac{1}{\Delta y^4}(u_{11} + u_{12} - 4u_4 - 4u_3 + 6u_0).$$

Члены со смешанными производными определим в узлах 3, 0, 4:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_4 \approx \frac{u_{15} + u_{16} - 2u_4}{\Delta x^2} \equiv m_4;$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_3 \approx \frac{u_{13} + u_{14} - 2u_3}{\Delta x^2} \equiv m_3;$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_0 \approx \frac{u_1 + u_2 - 2u_0}{\Delta x^2} \equiv m_0.$$

Тогда член выражения $\frac{\partial^4 u}{\partial x^2 \partial y^2}$ получим в виде

$$\left(\frac{\partial^4 u}{\partial x^2 \partial y^2}\right)_0 \approx \frac{m_1 + m_3 - 2m_0}{\Delta y^2}.$$

Подставив соответствующие значения *m,* найдем:

$$2\left(\frac{\partial^4 u}{\partial x^2 \partial y^2}\right)_0 \approx \frac{2}{\Delta x^2 \Delta y^2}\{u_{15} + u_{16} + u_{13} + u_{14} - 2(u_1 + u_2 + u_3 + u_4) + 4u_0\}.$$

Аппроксимация полного уравнения получается необходимой комбинацией найденных выражений.

3.4. Ошибки конечно-разностных аппроксимаций

При решении задач анализа полей в распознаваемых объектах методами конечно-разностных аппроксимаций неизбежны погрешности, присущие всем разностным методам нахождения потенциальных функций. Дело в том, что на модель анализируемого поля распознаваемого объекта мысленно наносят ортогональную координатную сетку, в узлах которой приближенно вычисляют искомую потенциальную функцию. В промежутках между узлами сетки методами интерполяции определяют промежуточные значения функции и строят искомые эквипотенциалы.

Таким образом, погрешности метода конечно-разностных аппроксимаций обусловливаются, главным образом, двумя причинами:

1) заменой дифференциальных уравнений конечно-разностными, приводящей к ошибкам при определении искомых функций в узловых точках;

2) отличием в построении интерполирующих и искомых функций, несмотря на точное определение потенциальных функций в узлах сетки.

Так как в задачах распознавания решение должно иметь заданную точность, то следует предвидеть ошибки, присущие любым способам вычислений.

Казалось бы, точность вычислений должна увеличиваться с уменьшением шага сетки. В общем случае это так, но для некоторых видов аппроксимаций точное значение потенциальной функции можно найти и при грубом шаге. Оказывается, что не только шаг сетки, но и характер анализируемого поля влияет на точность конечно-разностной аппроксимации.

Более точный анализ численных методов вычислений потенциальных функций на основе разностных схем показывает, что если непрерывная функция $\varphi(x)$ такова, что производные выше третьего порядка равны нулю, то ранее приведенные соотношения аппроксимации дают совершенно точное выражение для второй производной, независимо от шага сетки.

Если же потенциальная функция имеет более сложный характер и производные ее, имеющие порядок выше четвертого, отличны от нуля, то конечно-разностная аппроксимация такой функции приводит к ошибкам. Поэтому при выборе необходимой сетки дискретизации должна приниматься во внимание природа потенциальной функции, заданной внутри поля. Однако в более общем случае именно эта потенциальная функция и является целью решения задачи распознавания полей в распознаваемых объектах, поэтому

характеристики такой функции трудно предсказать заранее. Если при анализе поля в распознаваемом объекте четвертая производная потенциальной функции, вычисленная данными методами, окажется велика, то шаг сетки для рассматриваемой задачи распознавания уменьшается и вычисления повторяются.

Определение погрешности, когда потенциальная функция ориентировочно задается, сводится к представлению функции в узловой точке с помощью ряда Тейлора, записанного через значение функции в соседней точке и ее производные. Комбинируя их, можно найти точные выражения для производных искомой функции в виде степенных рядов. В зависимости от вида искомой функции производные высших порядков могут быть приравнены к нулю, а выражение для производных оказывается состоящим из конечного числа членов при этом погрешность легко оценивается.

Ряд Тейлора для функции в некоторой точке x, отстоящей относительно соседней точки x_0, записывается следующим образом:

$$\varphi = \varphi_0 + (x - x_0)\left(\frac{\partial\varphi}{\partial x}\right)_0 + \frac{(x-x_0)^2}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right)_0 + \frac{(x-x_0)^3}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right)_0 + \dots .$$

Чтобы вычислить потенциал в узлах *1* и *2* линейной сетки на рис. 5, *а* п.3.1 через потенциал и производную потенциала в узле О, множитель $x - x_0$ заменим значением Δx, в результате чего получим:

$$\varphi_1 = \varphi_0 + \Delta x\left(\frac{\partial\varphi}{\partial x}\right)_0 + \frac{\Delta x^2}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right)_0 + \frac{\Delta x^3}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right)_0 + \frac{\Delta x^4}{4!}\left(\frac{\partial^4\varphi}{\partial x^4}\right)_0 + \dots;$$

$$\varphi_2 = \varphi_0 - \Delta x\left(\frac{\partial\varphi}{\partial x}\right)_0 + \frac{\Delta x^2}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right)_0 - \frac{\Delta x^3}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right)_0 + \frac{\Delta x^4}{4!}\left(\frac{\partial^4\varphi}{\partial x^4}\right)_0 + \dots .$$

Складывая эти выражения, придем к соотношению

$$\varphi_1 + \varphi_2 = 2\varphi_0 + \Delta x^2\left(\frac{\partial^2\varphi}{\partial x^2}\right)_0 + \frac{\Delta x^4}{12}\left(\frac{\partial^4\varphi}{\partial x^4}\right)_0 + \dots ,$$

откуда

$$\frac{\partial^2\varphi}{\partial x^2} = \frac{1}{\Delta x^2}\left[\varphi_1 + \varphi_2 - 2\varphi_0 - \frac{\Delta x^4}{12}\left(\frac{\partial^4\varphi}{\partial x^4}\right)_0 - \dots\right].$$

При конечно-разностной аппроксимации

$$\frac{\partial^2\varphi}{\partial x^2} \approx \frac{1}{\Delta x^2}(\varphi_1 + \varphi_2 - 2\varphi_0),$$

поэтому ошибка за счет аппроксимации второй производной

$$\varepsilon_2 = -\frac{\Delta x^4}{12}\left(\frac{\partial^4\varphi}{\partial x^4}\right) + \dots .$$

При $\Delta x \to 0$ ошибка $\varepsilon_2 \to 0$. Из формулы для погрешности $\nabla^2\varphi$ следует, что если потенциальная функция имеет производные высших порядков, начиная с четвертого, равные нулю, то конечно-разностная аппроксимация точно описывает $\nabla^2\varphi$.

Для определения ошибок, вносимых аппроксимацией первой производной через разности вперед и назад, вычтем из выражений φ_1 и φ_2 значение φ_0. В результате получим:

$$\varphi_1 - \varphi_0 = +\Delta x\left(\frac{\partial\varphi}{\partial x}\right)_0 + \frac{\Delta x^2}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right)_0 + \frac{\Delta x^3}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right)_0 + \dots;$$

$$\varphi_2 - \varphi_0 = -\Delta x\left(\frac{\partial\varphi}{\partial x}\right)_0 + \frac{\Delta x^2}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right)_0 - \frac{\Delta x^3}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right)_0 + \dots .$$

Из найденных соотношений выразим соответственно:

$$\left(\frac{\partial\varphi}{\partial x}\right)_{0-1} = \frac{\varphi_1 - \varphi_0}{\Delta x} - \frac{1}{\Delta x}\left[\frac{\Delta x^2}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right)_0 + \right.$$

$$\left. + \frac{\Delta x^3}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right)_0 + \dots\right];$$

$$\left(\frac{\partial\varphi}{\partial x}\right)_{2-0} = \frac{\varphi_0 - \varphi_2}{\Delta x} + \frac{1}{\Delta x}\left[\frac{\Delta x^2}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right)_0 - \right.$$

$$\left. - \frac{\Delta x^3}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right) + \dots\right].$$

Таким образом, ошибки, вносимые аппроксимацией первой производной разностью вперед и назад, имеют вид:

$$\varepsilon_{1в} = -\frac{\Delta x}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right) - \frac{\Delta x^2}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right) - \dots;$$

$$\varepsilon_{1н} = \frac{\Delta x}{2!}\left(\frac{\partial^2\varphi}{\partial x^2}\right) - \frac{\Delta x^2}{3!}\left(\frac{\partial^3\varphi}{\partial x^3}\right) + \dots$$

Ошибки, вносимые более сложными видами конечно-разностных аппроксимаций и аппроксимацией производных более высоких порядков, могут быть оценены аналогичным способом.

3.5. Интерполяция, устойчивость и сходимость конечно-разностных аппроксимаций

Для определения закона изменения потенциальной функции, после вычисления значений этой функции в узлах сетки с заданной степенью точности, необходимо решить интерполяционную задачу — найти значения потенциальной функции в промежутках между узлами сетки.

В первом приближении применяют самый простой способ интерполяции, который предполагает линейное изменение потенциала в промежутке между двумя известными точками. Например, если известны значения потенциальной функции φ_0 и φ_1 в точках x_0 и x_1 (см. рис. 4, *а),*п.3.2 то значение этой функции в точке x_k, находящейся в промежутке между точками x_0 и x_1, определим из соотношения

$$\frac{\varphi_k - \varphi_0}{\varphi_1 - \varphi_0} = \frac{x_k - x_0}{x_1 - x_0} \quad \text{или} \quad \varphi_k = \varphi_0 + (\varphi_1 - \varphi_0)\frac{x_k - x_0}{x_1 - x_0}.$$

Однако на практике потенциальная функция даже на небольших интервалах может иметь явно нелинейный характер, особенно при наличии в распознаваемых объектах большого числа источников энергии (источников наводки, нагретых транзисторов и резисторов, точек крепления блоков в конструкциях и т. п.). В этом случае градиент потенциала между точками x_1 и x_0 не остается постоянными, следовательно, линейная интерполяция приводит к существенным ошибкам.

Максимальная абсолютная величина этой ошибки может быть оценена следующим образом:

$$|\delta_{\text{инт}}|_{\max} \leqslant \frac{\Delta x^2}{8}\left|\frac{\partial^2 \varphi}{\partial x^2}\right|.$$

В общем случае ошибка интерполяции максимальна посредине между двумя известными точками и стремится к нулю по мере сближения этих точек.

Для более точной интерполяции разработано много численных методов (Ньютона, Стирлинга, Бесселя, Лагранжа и др.), но все они, как правило, включают аппроксимацию непрерывной потенпиальной функции полинома, степень которого определяется числом используемых точек сетки. Например, если известно значение потенциальной функции в точках x_0, x_1, x_2, x_3 и т. д., то такую функцию, представленную с помощью полинома, записывают в виде

$$\varphi = p_0 + (x - x_0)p_1 + (x - x_0)(x - x_1)p_2 + (x - x_0)(x - x_1)(x - x_2)p_3 + (x - x_0)(x - x_1) \times (x - x_2)(x - x_3)p_4. \qquad (1)$$

Учитывание значений функции в большом числе точек приведет к появлению дополнительных членов полинома более высокого порядка, получаемых в результате перемножения всех сомножителей уравнения (1) и приведения подобных членов, что значительно увеличит время вычисления. Высшая степень полиномов всегда на единицу меньше числа известных точек интерполяции. Поэтому число точек сетки определяет предел объема вычислений или накладывает ограничение на возможности уточнения решения с помощью следующей методики: чем выше порядок, тем большую точность должна иметь интерполяция.

Коэффициенты при неизвестных p_0, p_1, ..., p_m определяют из уравнения (1) путем подстановки в него известных значений потенциалов поля.

При исследовании нестационарных процессов в распознаваемых объектах следует осуществлять проверку на вычислительную устойчивость, так как при определенных соотношениях шагов по пространственным и временным координатам вычислительный процесс может оказаться неустойчивым. Например, при очень малых шагах временной координаты Δt по сравнению с шагом пространственных координат Δx (Δy или Δz) любая малая ошибка (например, при округлении), сделанная на первом шаге интегрирования по времени, будет возрастать на последующих шагах. К концу вычисления эта ошибка может существенно перекрыть любые другие погрешности, связанные с интерполяцией или кусочно-разностным представлением области исследуемого пространства, и, таким образом, привести к неверным результатам. Такое явление называют вычислительной неустойчивостью процесса.

Любой анализ или исследование полей с помощью кусочно-разностной аппроксимации может приводить к заранее устойчивым, неустойчивым или устойчивым при некоторых ограничениях вычислениям. С этой точки зрения желательно сразу же по виду уравнений аппроксимации уметь определять устойчивость вычислений, так как некоторые виды конечно-разностных аппроксимаций, устойчивые в широком интервале значений отношения шагов сетки $\Delta t/\Delta x$, при некоторых условиях имеют критическое значение $\Delta t/\Delta x$. Если эта критическая величина превышена (выбрано очень малое отношение $\Delta t/\Delta x$), то решение становится неустойчивым, хотя в остальных случаях оно устойчиво. Таким образом, критерий устойчивости в подобных задачах имеет важное значение.

Существуют различные аналитические методы определения устойчивости. Рассмотрим метод, предложенный У. Карплюсом. Его идея

заключается в том, что если каждое непрерывное поле можно представить с помощью конечно-разностных аппроксимаций в виде моделей сеток различных сопротивлений, то всякому конечно-разностному уравнению соответствует своя сетка сопротивлений. При этом следует ожидать, что вычислительная неустойчивость в соответствующей сетке сопротивлений будет проявляться в виде электрической неустойчивости. Это значит, что в сетке сопротивлений, соответствующей математически неустойчивому конечно-разностному уравнению, любое флуктуационное отклонение напряжения в узловой точке от номинала, отображающего истинное решение, приводит к лавинообразному нарастанию токов и напряжений. Выясним, при каких условиях этот процесс не может возникнуть.

Рассмотрим элементарный контур сетки сопротивлений (см. рис. 4, *б)* п.3.1. Уравнение по второму закону Кирхгофа имеет вид

$$R_1 (i_0 - i_1) + R_2 (i_0 - i_2) + R_3 (i_0 - i_3) + R_4 (i_0 - i_4) = 0. \quad (2)$$

Очевидно, если все сопротивления контура положительны (в контуре отсутствуют источники напряжения), то данный контур устойчив, так как он содержит элементы, лишь рассеивающие энергию и, следовательно, приводящие к затуханию возникающие флуктуационные напряжения. В теории цепей доказывается, что контур, содержащий и отрицательные сопротивления, также может быть устойчивым, если алгебраическая сумма всех сопротивлений контура положительна. Отсюда вытекает метод проверки уравнений конечно-разностной аппроксимации на вычислительную устойчивость предложенный У. Карплюсом.

1. Записываем конечно-разностное уравнение в виде

$$(\varphi_{i+1,j} - \varphi_{ij}) + a(\varphi_{i-1,j} - \varphi_{ij}) + b(\varphi_{i+2,j} - \varphi_{ij}) + c(\varphi_{i,j+1} - \varphi_{ij}) + d(\varphi_{i,j-1} - \varphi_{ij}) + e(\varphi_{i,j+2} - \varphi_{ij}) + \dots, \quad (3)$$

где i относится к пространственным координатам, и исследуем коэффициенты a, b, c, d, e и т. д.

2. Если все коэффициенты положительны, то уравнение устойчиво.

3. Если некоторые из коэффициентов отрицательны, то достаточным условием устойчивости является наличие положительной алгебраической суммы всех коэффициентов уравнения аппроксимации, преобразованного в соответствии с п. 1.

Если конечно-разностное уравнение является аппроксимацией дифференциального уравнения в частных производных, то в общем случае его можно привести к виду (2) или (3).

Рассмотрим пример исследования устойчивости конечно-разностной аппроксимации диффузионного уравнения

$$\frac{\partial^2\varphi}{\partial x^2} = k\frac{\partial\varphi}{\partial t}.$$

Это уравнение может быть аппроксимировано с помощью способа разности назад или разности вперед, т. е.

$$\frac{\varphi_{i+1,j} + \varphi_{i-1,j} - 2\varphi_{ij}}{\Delta x^2} \approx k\frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{\Delta t} \text{ (разность назад);}$$

$$\frac{\varphi_{i+1,j} + \varphi_{i-1,j} - 2\varphi_{ij}}{\Delta x^2} \approx k\frac{\varphi_{i,j+1} - \varphi_{ij}}{\Delta t} \text{ (разность вперед).}$$

Перепишем уравнения согласно (3):

$$(\varphi_{i+1,j} - \varphi_{ij}) + (\varphi_{i-1,j} - \varphi_{ij}) + k'(\varphi_{i,j-1} - \varphi_{ij}) \approx 0;$$

$$(\varphi_{i+1,j} - \varphi_{ij}) + (\varphi_{i-1,j} - \varphi_{ij}) - k'(\varphi_{i,j+1} - \varphi_{ij}) \approx 0,$$

где $k' = k\Delta x^2/(\Delta t)$.

Применяя критерий устойчивости У. Карплюса, убедимся, что первое уравнение всегда устойчиво, а второе устойчиво при $k' < 2$, т. е. $\frac{k\Delta x^2}{\Delta t} < 2$ или $\Delta t > \frac{k\Delta x^2}{\Delta t}$.

Помимо устойчивости важной проблемой, возникающей при дискретном представлении временной переменной в дифференциальных уравнениях с частными производными, является сходимость вычислительного процесса. Конечно-разностная аппроксимация сходится, если приближенное решение конечно-разностного уравнения стремится к точному решению задачи по мере последовательного измельчения сетки при условии, что отношение пространственных шагов сетки вдоль различных координат сохраняется постоянным. Если шаг по координате *t* слишком велик по сравнению с шагом для пространственных координат, то решение может быть неудовлетворительным. Общего критерия сходимости таких процессов в литературе не имеется. Сходимость можно анализировать с помощью физического рассмотрения, аналогичного обсуждению проблемы устойчивости. В вычислительной математике утверждается, что если конечно-разностная система устойчива, то она также сходится. Устойчивость предполагает сходимость, хотя обратное утверждение не всегда верно. Так как в общем случае конечно-разностная сетка выбирается из условия устойчивости конечно-разностного приближения, то рассмотрение вопроса сходимости не вызывает особых затруднений.

3.6. Методы решения краевых задач на ЭВМ

Чаще всего при исследовании полей в распознаваемых объектах выясняется, что они описываются линейными дифференциальными уравнениями в частных производных. Такие задачи представляют собой класс линейных краевых задач математической физики.

Одним из общих методов решения линейных краевых задач является метод сведения подобной задачи к нескольким со специально подобранными начальными условиями (задачи Коши).

Каждая из этих задач решается одним из численных методов, такими, как метод Эйлера, Рунге—Кутта и т. д. Таким образом, получают ряд частных числовых решений, удовлетворяющих дифференциальному уравнению, но не удовлетворяющих в общем случае краевым условиям. Если краевые условия задаются в n точках, задачу называют n-точечной.

Рассмотрим решение двухточечной краевой задачи для линейного дифференциального уравнения второго порядка

$$a(x)\frac{d^2\varphi}{dx^2} + b(x)\frac{d\varphi}{dx} + c(x)\varphi = f(x). \qquad (1)$$

Краевые условия задаются в двух точках x_0 и x_N:

$$\varphi(x_0) = \varphi_0;\ \varphi(x_N) = \varphi_N.$$

Методы численного интегрирования строятся на решении двух задач Коши для однородного уравнения

$$a(x)\frac{d^2\varphi}{dx^2} + b(x)\frac{d\varphi}{dx} + c(x)\varphi = 0. \qquad (2)$$

Для первой задачи за начальные условия примем $\varphi(x_0) = 1$; $\varphi'(x_0) = 0$; для второй $\varphi(x_0) = 0$; $\varphi'(x_0) = 1$.

Обозначим полученные решения символами $\varphi^*(x)$ и $\varphi^{**}(x)$. Найдем решение задачи Коши для неоднородного уравнения с однородными начальными условиями $\varphi(x_0) = 0$; $\varphi'(x_0) = 0$, которое обозначим символом $\varphi^{***}(x)$.

Решение исходной краевой задачи выразим как линейную комбинацию частных решений:

$$\varphi(x) = c_1\varphi^*(x) + c_2\varphi^{**}(x) + \varphi^{***}(x). \qquad (3)$$

Коэффициенты c_1 и c_2 определим, исходя из краевых условий. Подставляя полученную линейную комбинацию для $\varphi(x)$ в исходные краевые условия, получим систему линейных алгебраических уравнений:

$$\left.\begin{aligned}\varphi(x_0) &= c_1\varphi^*(x_0) + c_2\varphi^{**}(x_0) + \varphi^{***}(x_0) = \varphi_0;\\ \varphi(x_N) &= c_1\varphi^*(x_N) + c_2\varphi^{**}(x_N) + \varphi^{***}(x_N) = \varphi_N.\end{aligned}\right\} \quad (4)$$

Определитель этой системы

$$\begin{vmatrix} \varphi^*(x_0) & \varphi^{**}(x_0) \\ \varphi^*(x_N) & \varphi^{**}(x_N) \end{vmatrix}.$$

Если определитель отличен от нуля, то коэффициенты c_1 и c_2 находим из системы (4) однозначно. Следовательно, получаем и решение всей задачи.

Таким образом, при решении краевой задачи второго порядка методом частных решений выполняют численное интегрирование трех задач с начальными условиями.

Аналогичным образом решают линейные дифференциальные уравнения *n*-го порядка. Для двухточечной задачи общее решение определяют в виде *n* частных линейно-независимых решений соответствующего однородного уравнения плюс *(n*+1)-е частное решение неоднородного уравнения [см. (4)]. В задаче задаются *n* начальными условиями по правилам единичной матрицы. Функция $\varphi_{n+1}(x)$ определяется неоднородным дифференциальным уравнением и однородными начальными условиями:

$$\varphi_{n+1}(x_0)=0;\ \frac{d\varphi_{n+1}(x_0)}{dx} = 0;\ \ldots;\qquad \frac{d^{n-1}\varphi_{n+1}(x_0)}{dx^{n-1}} = 0.$$

Рассмотренный метод пригоден для решения линейных дифференциальных уравнений. При этом краевые условия в общем случае могут быть нелинейными. Когда краевые условия задаются в линейной форме, для определения коэффициентов c_i (где i = 1, 2, ..., *n)* применяют методы решения систем линейных алгебраических уравнений. Если же краевые условия задаются в нелинейном виде, то используют методы решения систем нелинейных уравнений.

Метод частных решений очень удобен при реализации на ЭВМ, однако в отдельных случаях получают невысокую вычислительную точность.

Специально для использования ЭВМ при решении краевых задач был разработан метод прогонки, который успешно применяют для решения различных краевых задач, сводящихся как к решению простых дифференциальных уравнений, так и к решению дифференциальных уравнений в частных производных.

Пример. Методом прогонки решить линейное дифференциальное уравнение второго порядка

$$\frac{d^2\varphi}{dx^2} + a(x)\frac{d\varphi}{dx} + b(x)\varphi = j(x) \quad (5)$$

при граничных условиях $\varphi(x_0) = \varphi_0;\ \varphi(x_N) = \varphi_N$.

Решение. Поскольку условия заданы в двух различных точках, задача сводится к построению интегральной кривой, проходящей через две заданные точки. Такая задача имеет единственное решение. Рассмотрим аналитическое решение этой задачи. Метод прогонки представляет собой одну из разновидностей метода исключения неизвестных Гаусса.

Разделим участок интегрирования $[x_0,\ x_N]$ на n частей.

$$(x_0,\ x_1),\ (x_1,\ x_2),\ \ldots,\ (x_{n-2},\ x_{n-1}),\ (x_{n-1},\ x_N).$$

Длина каждой части $h = (x_N - x_0)/n$.

В точках деления отрезков заменим производные, входящие в исходное уравнение, конечными разностями. В результате получим:

$$\left.\frac{d\varphi}{dx}\right|_{x=x_i} \approx \frac{\varphi_{i+1} - \varphi_i}{h};$$

$$\left.\frac{d^2\varphi}{dx^2}\right|_{x=x_i} \approx \frac{\varphi_{i+2} - 2\varphi_{i+1} + \varphi_i}{h^2},$$

где φ_i — значение искомой функции в i-й точке деления участка $[x_0,\ x_N]$, т. е. $\varphi(x_i) = \varphi_i$. Обозначим

$$a(x_i) = a_i;\ b(x_i) = b_i;\ f(x_i) = f_i.$$

Подставляя выражение для конечных разностей в уравнение (5), получим систему из $(n-1)$-го разностного уравнения с $n - 1$ неизвестными:

$$\begin{cases} \varphi_2 + \varphi_1(a_0h - 2) + \varphi_0(1 - a_0h + b_0h^2) = h^2f_0; \\ \varphi_3 + \varphi_2(a_1h - 2) + \varphi_1(1 - a_1h + b_1h^2) = h^2f_1; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \varphi_{k+2} + \varphi_{k+1}(a_kh - 2) + \varphi_k(1 - a_kh + b_kh^2) = h^2f_k; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \varphi_n + \varphi_{n-1}(a_{n-2}h - 2) + \varphi_{n-2}(1 - a_{n-2}h + b_{n-2}h^2) = h^2f_{n-2}. \end{cases} \quad (6)$$

Таким образом, решение краевой задачи свелось к решению системы разностных уравнений (6) с условиями $\varphi(x_0) = \varphi_0;\ \varphi(x_N) = \varphi_N$.

Из первого уравнения системы (6) выразим φ_1:

$$\varphi_1 = \frac{h^2f_0 - \varphi_0(1 - a_0h + b_0h^2)}{a_0h - 2} - \frac{1}{a_0h - 2}\varphi_2.$$

В результате проделанных преобразований получили линейную зависимость (пропорциональность) φ_1 от φ_2. Обозначим коэффициенты

пропорциональности, называемые прогоночными коэффициентами, через

$$p_{1,2} = -\frac{1}{a_0 h - 2}; \quad q_{1,2} = \frac{h^2 f_0 - \varphi_0 (1 - a_0 h - b_0 h^2)}{a_0 h - 2} =$$

$$= p_{1,2} [-h^2 f_0 + \varphi_0 (1 - a_0 h + b_0 h^2)].$$

Первое уравнение системы (6) запишем короче, т. е. в виде соотношения

$$\varphi_1 = p_{1,2} + q_{1,2} \varphi_2.$$

Подобным же образом представим второе уравнение системы (6):

$$\varphi_2 = p_{2,3} + q_{2,3} \varphi_3,$$

где

$$p_{2,3} = \frac{1}{(a_1 h - 2) + p_{1,2} (1 - a_1 h + b_1 h^2)};$$

$$q_{2,3} = p_{2,3} [h^2 f_1 - (1 - a_1 h + b_1 h^2) p_{1,2}].$$

По аналогии запишем выражение, устанавливающее связь между φ_k и φ_{k+1} при любом φ_k.

$$\varphi_k = p_{k,\,k+1} + q_{k,\,k+1} \varphi_{k+1},$$

где $k = 0, 1, 2, \ldots, n - 2$, а также рекуррентные соотношения для определения прогоночных коэффициентов связи $p_{k,k+1}$ и $q_{k,k+1}$ через $p_{k-1,k}$, $q_{k-1,k}$ и значения коэффициентов уравнения для промежуточных узлов интегрирования.

Метод прогонки осуществляют в два этапа. На первом этапе *(прямая прогонка),* используя граничное условие

$$\varphi(x_0) = \varphi_0$$

по рекуррентным формулам вычисляют коэффициенты связи $q_{k,k+1}$ $p_{k,k+1}$, для каждого $k (k = 0, 1, 2, \ldots, n - 2)$.

Второй этап *(обратная прогонка)* связан с вычислением по формулам для φ_k, используя второе граничное условие $\varphi(x_N) = \varphi_N$ и найденные на первом этапе коэффициенты $p_{k,k+1}$, $q_{k,k+1}$, последовательных значений искомой функции для каждого узла интегрирования: x_{n-1}, x_{n-2}, ..., x_1.

Метод по своей структуре несложен в программировании и очень эффективен при использовании ЭВМ. Отметим, что для вычисления

методом прогонки системы (6), состоящей из $n + 1$ уравнений, нужно проделать арифметические операции, количество которых лишь в конечное число раз больше, чем число неизвестных. При решении произвольной линейной системы уравнений с n неизвестными обычными методами исключения приходится выполнять порядка n^3 арифметических операций. Такого сокращения числа арифметических операций при решении системы (6) прогонки удалось достигнуть, удачно использовав специфику этой системы.

4. Использование теории графов для описания распознаваемых объектов

Применение вычислительных средств при распознавании объектов по-новому ставит задачи разработки математических моделей и методов их анализа и оптимизации. Отличительной чертой в постановке задач распознавания объектов является максимальная формализация математических описаний и использование для отыскивания оптимальных решений аппарата математического программирования.

В общем случае под математической моделью распознаваемого объектов понимают систему математических соотношений, описывающих с требуемой точностью распознаваемый объект и его поведение в реальных условиях. Процесс составления математических моделей называют математическим моделированием. В основу математического моделирования положен принцип идентичности формы уравнений и одозначности соотношений между переменными в уравнениях оригинала и модели, т. е. принцип аналогии распознаваемого объектов с моделью. При составлении математических моделей могут использоваться различные математические средства описания объекта — дифференциальные или интегральные уравнения, теория множеств, теория графов, теория вероятностей, математическая логика и др. Особое место в математическом моделировании занимает квазианалоговое моделирование, суть которого состоит в изучении не распознаваемого объекта, а объекта иной физической природы, но описываемого математическими соотношениями, эквивалентными относительно получаемого результата.

В данном разделе рассмотрены вопросы применения теории множеств и теории графов, для описания распознаваемых объектов и моделирования протекающих в них процессов.

4.1. Основные понятия теории множеств и теории графов

Определения. Математические методы, положенные в основу алгоритмических процессов распознавания объектов, а также процессы организации входной и выходной информации о распознаваемом объекте широко используют понятия и символы теории множеств.

Под множеством понимают совокупность объектов любой природы, называемых элементами данного множества, обладающих каким-либо общим для множества свойством. Как основное понятие теории понятие множества не подлежит логическому определению.

Элементы множества могут иметь самую различную природу. Например, можно говорить о множестве микросхем, входящих в определенную конструкцию РЭА, или о множестве чертежей, входящих в полный комплект конструкторской документации для производства какого-либо изделия, и т. д.

Множества обозначают заглавными буквами латинского алфавита: *X, Y, Z,* а элементы множеств — соответствующими строчными буквами того же алфавита: *x, y, z* или строчными буквами с индексами: $x_1, x_2, \ldots; y_1, y_2, \ldots$. Равенство $X = \{x_1, x_2, \ldots, x_n\}$ свидетельствует о том, что элементы $x_1, x_2, \ldots, x_n$ являются элементами множества *X.*

Множество можно задавать не только перечислением его элементов, но и с помощью описательного способа, указывающего характерное свойство, которым обладают все элементы этого множества. Например, если во всем множестве *X* микросхем электронного блока сложной радиоаппаратуры есть некоторое множество *A* гибридных интегральных схем, то это можно записать следующим образом: $A = \{x \in X : x$ — гибридная интегральная схема$\}$, что читается так: множество *A* состоит из элементов *x* множества *X,* обладающих тем свойством, что *x* является гибридной интегральной схемой. Здесь введено новое обозначение $\in$, означающее, что объект *x* является элементом множества *X.* Если же некоторый объект *y* не принадлежит множеству *X,* то это условие записывают в виде $y \notin X$.

В том случае, когда не вызывает сомнения, из какого множества берутся элементы *x,* принадлежность их к множеству *X* можно не указывать. Например, если известно, что множество гибридных интегральных схем входит во множество микросхем того же самого электронного блока, то можно записать $A = \{x : x$ — гибридная интегральная схема$\}$.

Число элементов множества $X = \{x_1, x_2, \ldots, x_n\}$ называют мощностью этого множества и обозначают прямыми скобками, например $|X| = n$. Если число элементов множества *X* конечно, то такое множество называют конечным. В противном случае множество будет бесконечным. В теории множеств вводится понятие пустого множества, в котором не содержится ни одного элемента. Пустое множество обозначают специальным символом Ø. Так, например, если множество *X* пусто, то пишут $X = \varnothing$.

Последовательность из *n* элементов множества называют *n*-строкой. В отличие от обычного множества, где порядок элементов безразличен, в *n*-строке обязательно задается их определенная последовательность.

Множество *X* равно множеству *Y,* если оба эти множества состоят из одних и тех же элементов. Если множество *X* полностью содержится во множестве *Y* и при этом $|X| \leqslant |Y|$, то говорят, что множество *X* является подмножеством множества $Y : X \subset Y$. В случае когда $X \subset Y$ и одновременно $Y \subset X$, имеет место равенство *X= Y,* т е. множества *X* и *Y* совпадают. Символическая запись $X \neq Y$ означает, что множество *X* не совпадает с множеством *Y.*

Действия над множествами. Над множествами, как и над другими математическими величинами, можно производить некоторые действия, например выполнять пересечение множеств, их объединение, вычитание, находить дополнение, декартово произведение и др.

Пересечением множеств *X* и *Y* называют новое множество *P,* которое образуется из элементов, одновременно общих и множеству *X,* и множеству *Y.* На рис. 1, а множество *P* показано заштрихованной областью.

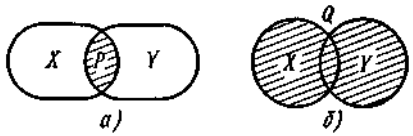


Рис. 1

Пересечение множеств *X* и *Y* записывают следующим образом: $P = X \cap Y$. Если рассматривают пересечение нескольких множеств $X_1, X_2, \ldots, X_i, \ldots, X_r$,

то математическая записьимеет вид

$$P = \bigcap_{i=1}^{r} X_i,$$

где *r* — число пересекающихся множеств

Операция пересечения множеств подчиняется переместительному закону, т. е. $P = X \cap Y = Y \cap X$. Если множества X и Y не пересекаются, то $P = X \cap Y = \varnothing$.

С помощью операции пересечения множеств можно, например, выявить множество типоразмеров элементов распознаваемых объектов, общих распознаваемым объектам X и Y, или множество межобъектных соединений для распознаваемых объектов X и Y, т. е. выявить любые множества, обладающие какими-либо общими свойствами.

Объединение множеств X и Y приводит к образованию нового множества Q, которое получается из всех тех и только тех элементов, которые принадлежат хотя бы одному из множеств X или Y. На рис. 1, б такое множество показано заштрихованной областью.

Математически объединение множеств X и Y записывают следующим образом: $Q = X \cup Y$. Если рассматривают объединение нескольких множеств, то запись примет вид

$$Q = \bigcup_{i=1}^{r} X_i,$$

где r — число объединяемых множеств. Операция объединения множеств, так же как и операция пересечения, подчиняется переместительному закону.

С помощью этой операции можно подсчитать, например, число типоразмеров элементов для распознаваемых объектов X и Y или общее число внешних соединений распознаваемых объектов X и Y.

Разность множеств X и Y есть новое множество R, которое образуется из элементов множества X, за исключением элементов, принадлежащих одновременно множеству Y. На рис. 2, *а* множество R показано в виде заштрихованной области. Математически разность множеств X и Y записывают следующим образом: $R = X/Y$.

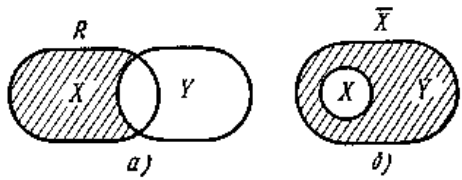


Рис. 2

С помощью этой операции можно выявить сугубо индивидуальные признаки распознаваемого объекта, например число типоразмеров элементов, принадлежащих только распознаваемому объекту X.

Дополнением множества X по отношению к множеству Y называют множество $\overline{X}$, состоящее из элементов множества Y, не принадлежащих множеству X. На рис. 2, *б* множество $\overline{X}$ показано в виде заштрихованной области. С помощью операции дополнения множества можно выявить все дополнительные, недостающие признаки распознаваемого объекта и подвергнуть их анализу.

Декартовым произведением множеств X и Y называют множество Z упорядоченных пар *(x, y),* образованных элементами множеств X и Y: $Z = X \times Y$. На рис. 3 декартово произведение множеств X_1 и Y_2 показано в виде заштрихованной области множества паросочетаний.

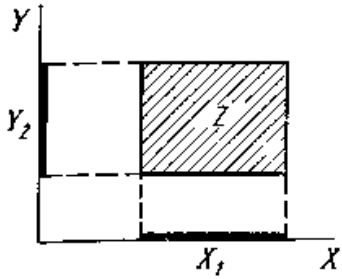


Рис. 3

Декартово произведение двух множеств используют для исследования всевозможных паросочетаний. Декартово произведение нескольких множеств

$$Z = X_1 \times X_2 \times \ldots \times X_r = \prod_{i=1}^{r} X_i$$

представляет собой множество r-строчек, каждая из которых образуется упорядоченной композицией элементов исходных множеств, т. е. $z_s = (x_{1j}, x_{2j}, \ldots, x_{rk})$. Операция декартова произведения множеств не обладает переместительным свойством, т. е. $X \times Y \neq Y \times X$

Разбиением множествах называют такое множество множеств $\{X_j\}$, где $j \in J$, а J — некоторое множество индексов j, при котором:

1) $X_j \subset X$ при всех $j \in J$;
2) $X_j \neq \varnothing$ при всех $j \in J$;
3) $X_i \cap X_j = \varnothing$ при $i \neq j$;
4) $\bigcup_{j \in J} X_j = X$.

Ряд задач разбиения множества элементов высокого уровня на элементы более низкого уровня (например, задача разбиения множества блоков распознаваемого объекта на отдельные субблоки) сводится к операциям разбиения множеств.

Понятие пустого множества $\varnothing$ аналогично нулю в алгебре чисел. Действительно, если для любого числа a справедливо $a \times 0 = 0$ и $a + 0 = a$, то для любого множества X справедливо

$$X \cap \varnothing = \varnothing$$

и

$$X \cup \varnothing = X.$$

Введем понятие множества I, соответствующее единице в алгебре чисел. Такое множество должно обладать тем свойством, что пересечение с ним любого множества X дает в результате это же множество X, т. е. $X \cap I = X$ по аналогии с $a \times 1 = a$.

Множество I, обладающее этим свойством называют универсальным или единичным множеством. В общем случае, если при некотором рассмотрении участвуют только подмножества некоторого фиксированного множества I, то это самое большое множество и является универсальным.

В конкретных приложениях в качестве универсального множества могут использоваться различные общие подмножества. Например, среди множества комплектов конструкторских документов на изготовление изделий РЭА полный комплект конструкторских документов является универсальным множеством этих документов или когда при рассмотрении множеств микросхем отдельных субблоков РЭА выделяют универсальное множество таких микросхем на всю данную радиоэлектронную аппаратуру в целом.

Универсальное множество обладает свойством, не имеющим аналога в алгебре чисел, а именно, для любого множества X справедливо соотношение $X \cup I = I$.

В объединение этих множеств должны входить как элементы множества X, так и дополняющие элементы множества I. Но, в свою очередь, все элементы множества X входят в универсальное множество I, поэтому и объединение $X \cup I$ равно универсальному множеству I.

На основании этих рассуждений легко определить дополнение множества X как $\overline{X} = I \setminus X$. Двойное дополнение $\overline{\overline{X}} = X$.

С помощью операции дополнения можно в удобном виде представить разность множеств

$$X \setminus Y = \{ x : x \in X \text{ и } x \notin Y \} = \{ x : x \in X \text{ и } x \in \overline{Y} \},$$

т. е.

$$X \backslash Y = X \cap \overline{Y}.$$

Многие определения теории множеств удобно записывать в виде математических выражений, содержащих некоторые логические символы. К числу таких символов относится символ следствия (импликации) $\Rightarrow$. Например, запись $X \subset Y$ и $Y \subset Z \Rightarrow X \subset Z$ (транзитивность) читают так: если $X \subset Y$ и $Y \subset Z$, то $X \subset Z$. Другие символы связаны с применением кванторов общности и существования. Квантор общности — это операция, которая сопоставляет *P(x)* высказыванию: «Все *x* обладают свойством *P(x)*». Для этой операции употребляют знак $\forall$ (перевернутое латинское *A*). Например, запись $\forall x(P(x) \Rightarrow Q(x))$ свидетельствует о том, что все объекты, обладающие свойством *P(x),* обладают и свойством *Q(x).*

Наряду с квантором общности в теории множеств существует понятие квантора существования, обозначаемого $\exists$ (перевернутая латинская буква *E).* Например, запись

$$\exists x\,(P\,(x) \cap Q\,(x))$$

утверждает, что существует по крайней мере один объект *x,* обладающий одновременно свойствами $P(x)$ и $Q(x)$, т. е. $P(x)$ и $Q(x)$ пересекаются : $P(x) \cap Q(x) \neq \varnothing$.

В теории множеств часто пользуются понятием логической єквивалентности (в смысле то же самое, что ...), обозначаемой $\Leftrightarrow$. Например, запись

$$X \subset Y \text{ и } Y \subset X \Leftrightarrow X = Y$$

нужно читать: «Выполнение условий $X \subset Y$ и $Y \subset X$ то же самое, что $X = Y$».

Пример 1. Доказать с помощью тождественных преобразований равенство $(X \cup Y) \cap Z = (X \cap Z) \cup (Y \cap Z)$ и показать с помощью диаграмм его коммутативные свойства.

Решение. Это равенство известно как тождество дистрибутивности операций над множествами. Чтобы убедиться в справедливости этого тождества, положим $a \in (X \cup Y) \cap Z$. Тогда одновременно $a \in X \cup Y$ и $a \in Z$, что возможно в случае, когда $a \in X \cap Z$ или $a \in Y \cap Z$, т. е. $a \in (X \cap Z) \cup (Y \cap Z)$. Отсюда можно заключить, что $(X \cup Y) \cap Z \subset (X \cap Z) \cup (Y \cap Z)$. Аналогично доказывается соотношение $(X \cap Z) \cup (Y \cap Z) \subset (X \cap Y) \cap Z$. В соответствии с определением равенства множеств приходим к требуемому тождеству. На рис. 4, *а* показан набор исходных множеств *X, Y* и *Z,* а на рис. 4, *б, в* — комбинация множеств в соответствии с выражениями $(X \cup Y) \cap Z$ и $(X \cap Z) \cup (Y \cap Z)$.

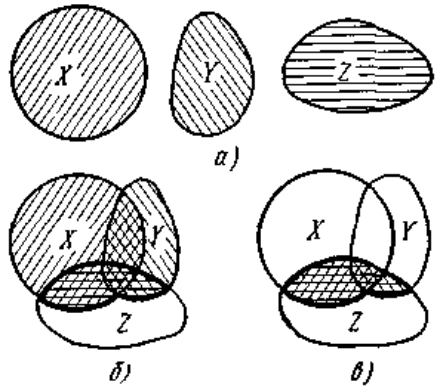


Рис. 4

Внутренние области, ограниченные жирными линиями, совпадают. Можно проследить, что операции над множествами по их объединению или пересечению обладают также коммутативностью и ассоциативностью.

Отношения множеств. Виды отношений и их свойства. Элементы множества, как правило, находятся в каком-либо отношении друг относительно друга. Эти отношения можно задать в виде неполных предложений — предикатов, например, «меньше, чем...», «больше, чем ...», «эквивалентно», «конгруэнтно» и т. п.

Тот факт, что некоторый элемент $x_i \in X$ находится в каком-либо отношении к элементу того же множества x_j, математически записывают как $x_i R x_j$, где R — символ отношения.

Отношение из двух элементов множества X называют бинарным. Бинарные отношения множеств X и Y представляют собой некоторое множество упорядоченных пар (x, y), образованных декартовым произведением $X \times Y$. В общем случае можно говорить не только о множестве упорядоченных пар, но и о множестве упорядоченных троек, четверок элементов и т. д., т. е. о n-арных отношениях, получаемых в результате декартова произведения $X_1 \times X_2 \times \times \ldots \times X_n$, где n — размерность n-строчек. Рассмотрим основные виды отношений — отношения эквивалентности, порядка и доминирования.

Некоторые элементы множеств можно считать эквивалентными в том случае, когда любой из этих элементов при определенных условиях можно заменить другим, т. е. данные элементы находятся в отношении эквивалентности. Примерами отношений эквивалентности являются отношения параллельности на множестве прямых какой-либо

плоскости; подобия на множестве треугольников; принадлежности к одной функциональной группе микросхем или к одному классу типоразмеров и т. д.

Термин «отношение эквивалентности» будем применять при выполнении следующих условий:

1) каждый элемент эквивалентен самому себе;

2) высказывание, что два элемента являются эквивалентными, не требует уточнения того, какой из элементов рассматривается первым, а какой вторым;

3) два элемента, эквивалентные третьему, эквивалентны между собой.

Введем для обозначения эквивалентности символ $\sim$, тогда рассмотренные условия можно записать следующим образом:

1) $x \sim x$ (рефлективность);

2) $x \sim y \Rightarrow y \sim x$ (симметричность);

3) $x \sim y$ и $y \sim z \Rightarrow x \sim z$ (транзитивность).

Следовательно, отношение R называют отношением эквивалентности, если оно рефлексивно, симметрично и транзитивно.

Пусть некоторому элементу $x \in X$ эквивалентно некоторое подмножество элементов $A \subset X$, тогда это подмножество образует класс эквивалентности, эквивалентный x. Очевидно, что все элементы одного и того же класса эквивалентности эквивалентны между собой (свойство транзитивности). Тогда всякий элемент $x \in X$ может находиться в одном и только одном классе эквивалентности, т. е. в этом случае множество X разбивается на некоторое непересекающееся подмножество классов эквивалентности

$$\{A_j \subseteq X : j \in J\},$$

где J — некоторое множество индексов.

Таким образом, каждому отношению эквивалентности на множестве X соответствует некоторое разбиение множества X на классы A_j.

В теории распознавания часто сталкиваются с отношениями, которые определяют некоторый порядок расположения элементов множества. Например, в процессе автоматизированного распознавания требуется вводить множество одних исходных данных *раньше* или *позже,* чем множество других. При этом может оказаться, что элементы одного множества больше или меньше элементов другого и т. д. Во всех этих случаях можно расположить элементы множества X или группы элементов в некотором порядке (например, в виде убывающей или возрастающей последовательности), т. е. ввести отношение порядка на множестве X.

Различают отношения строгого порядка, для которых применяют символы $<$, $\subset$, $\Rightarrow$, и отношения нестрогого порядка, где используют символы $\leqslant$ и $\subseteq$. Эти отношения характеризуются следующими свойствами:

для отношения строгого порядка:

$x < x$ — ложно (антирефлексивность);

$x < y$ и $y < x$ — взаимоисключаются (несимметричность);

$x < y$ и $y < z \Rightarrow x < z$ — (транзитивность);

для отношения нестрогого порядка:

$x \leqslant x$ — истинно (рефлексивность);

$x \leqslant y$ и $y \leqslant x \Rightarrow x = y$ — (антисимметричность);

$x \leqslant y$ и $y \leqslant z \Rightarrow x \leqslant z$ — (транзитивность).

Множество X называют упорядоченным, если любые два элемента x и y этого множества сравнимы, т. е. если для них выполняется одно из условий: $x < y$, $x = y$, $y < x$.

Упорядоченное множество называют кортежем. В общем случае кортеж — это последовательность элементов, т. е. совокупность элементов, в которой каждый элемент занимает вполне определенное место. Элементы упорядоченного множества называются компонентами кортежа. Примерами кортежа может служить упорядоченная последовательность чисел арифметической или геометрической прогрессий, последовательность операций распознавания при распознавании какого-либо объекта, упорядоченная последовательность установочных позиций печатной платы для закрепления конструктивных элементов.

Во всех этих множествах место каждого элемента вполне определено и не может произвольно изменяться.

При обработке информации на ЭВМ часто используют отношения доминирования. Говорят, что $x \in X$ доминирует над $y \in X$, т. е. $x \gg y$, если элемент x в чем-либо превосходит (имеет приоритет) элемент y того же множества. Например, под x можно понимать один из списков данных, который должен поступить на обработку первым. При распознавании нескольких объектов какой-либо из них должен быть отдан приоритет, так как этот объект обладает лучшими, с нашей точки зрения, свойствами, чем другие, т. е. объект x доминирует над объект y.

Свойство транзитивности при этом не имеет места. Действительно, если, например, объект x по каким-либо одним параметрам предпочли объекту y, а объект y по каким-либо другим параметрам предпочли

объекту *z*, то отсюда еще не следует, что объекту *x* должно быть отдано предпочтение по сравнению с объектом *z*.

Отображение множеств. Одним из основных понятий теории множеств является понятие отображения. Если заданы два непустых множества *X* и *Y*, то закон, согласно которому каждому элементу $x \in X$ ставится в соответствие элемент $\Gamma x \in Y$, называют однозначным отображением *X* в *Y* или функцией, определенной на *X* и принимающей значение на *Y*.

На практике приходится иметь дело и с многозначными ототображениями множества *X* на множестве *Y*, которые определяют закон, согласно которому каждому элементу $x \in X$ ставится в соответствие некоторое подмножество $\Gamma x \subseteq Y$, называемое образом элементах. Возможны случаи, когда $\Gamma x = \varnothing$.

Пусть задано некоторое подмножество $A \subseteq X$. Для любого $x \in A$ образом *x* является подмножество $\Gamma x \subseteq Y$. Совокупность всех элементов *Y*, являющихся образами для всех $x \in A$, назовем образом множества *A* и будем обозначать Γ*A*. В этом случае

$$\Gamma A = \bigcup_{x \in A} \Gamma x.$$

Рассмотрим некоторые свойства отображений. Если заданы два подмножества $A_1 \subset X$ и $A_2 \subset X$, то для отображения объединения этих подмножеств

$$\Gamma (A_1 \cup A_2) = \bigcup_{x \in A_1 \cup A_2} \Gamma x = \left(\bigcup_{x \in A_1} \Gamma x \right) \cup \left(\bigcup_{x \in A_2} \Gamma x \right) = \Gamma A_1 \cup \Gamma A_2.$$

При отображении пересечений этих подмножеств $\Gamma(A_1 \cap A_2)$ соотношение $\Gamma(A_1 \cap A_2) = \Gamma A_1 \cap \Gamma A_2$ справедливо только в том случае, когда отображение является однозначным. В общем случае имеет место выражение $\Gamma(A_1 \cap A_2) \subseteq \Gamma A_1 \cap \Gamma A_2$.

В использованиии теории множеств теории распознавания широко распространены многократные отображения, получаемые на одном и том же множестве элементов. Пусть Γ и Δ — отображения множества *X* в *X*. Произведением (композицией) этих отображений назовем отображение $\Gamma\Delta$, которое согласно свойству ассоциативности композиции определим следующим образом:

$$(\Gamma\Delta)\, x = \Gamma (\Delta x).$$

Для многократного отображения множества *X* в *X*, когда $\Gamma = \Delta$,

$$\Gamma^2 x = \Gamma (\Gamma x); \quad \Gamma^3 x = \Gamma(\Gamma^2 x) = \Gamma(\Gamma(\Gamma x)) \text{ и т. д.}$$

В общем случае $\Gamma^\alpha x = \Gamma(\Gamma^{\alpha-1} x)$. Тогда

$$\Gamma^0 x = \Gamma(\Gamma^{-1}x) = \Gamma\Gamma^{-1}x = x.$$

Приведенная запись означает, что $\Gamma^{-1}x$ представляет собой обратное отображение, а $\Gamma^{-2}x = \Gamma^{-1}(\Gamma^{-1}x)$ и т. д.

В заключение отметим, что при более строгом рассмотрении между отображением и функцией все же имеется некоторое различие, характеризуемое способом определения этих отношений на множестве *X*, причем отображение следует рассматривать как частный случай функции.

Функциональное отношение $A \subset X \times Y$ называют отношением множества *X* в *Y*, если это отношение всюду определено на *X*, т. е. его область определения $D_0(A)$ совпадает с множеством *X*.

Отношение $A \subset X \times Y$ называют функциональным, если все его элементы (упорядоченные пары) имеют различные первые координаты, т. е. каждому элементу $x \in X$, такому, что $(x, y) \in A$, соответствует один и только один элемент $y \in Y$. При этом первая координата *x* упорядоченной пары $(x, y) \in A$ является аргументом (переменной), а вторая *y* — образом (значением) функции.

Пример 2. Во множестве $N = \{1, 2, 3, 4, 5, 6\}$ заданы отношения:

$$\{(1, 3), (2, 4), (2, 6), (3, 5), (3, 2)\}, \quad \text{(а)}$$

$$\{(1, 6), (2, 2), (3, 5), (4, 5), (5, 6)\}. \quad \text{(б)}$$

Какие из этих отношений являются функциями и какие отображениями?

Решение. В выражениях (а) и (б) первое отношение является отображением, второе — функцией, так как для второго отношения все первые координаты отличны друг от друга, а для первого это условие не выполняется.

Рассмотрим пример конструирования печатной платы. Пусть *x* — некоторое исходное расположение конструктивных элементов на плате; *X* — множество различных расположений таких элементов на плате. Тогда Γx для любого $x \subset X$ —множество положений, которые можно получить из *x*, например с помощью парных перестановок конструктивных элементов, делая один шаг перестановок в направлении улучшения некоторого показателя качества размещения. При этом $\Gamma^4 x$ —множество перестановок конструктивных элементов, которые можно выполнить из состояния *x* четырьмя шагами; $\Gamma^{-1}x$ — множество положений (состояний) конструктивных элементов, из которых данное положение может быть получено за один шаг. Если из положения *x* перестановками с другими элементами не удается улучшить показатель качества размещения (достичь локальный оптимум показателя качества), то $\Gamma x = \varnothing$.

Рассмотрим основные определения из теории графов.

Теорию графов применяют для решения таких задач, как анализ электронных схем, распределение и размещение конструктивных элементов, проектирование проводного и печатного монтажа, сетевое планирование и многих других. Это объясняется тем, что использование графов сокращает объем вычислений по сравнению с обычными методами и, сохраняя наглядность описания распознаваемых объектов, позволяет строить компактные и удобные для реализации на ЭВМ алгоритмы преобразований и оптимизации.

Основные понятия. Понятие графа опирается на понятие множества. Под абстрактным графом или просто графом $G(X, U)$ понимают совокупность непустого множества *X* и изолированного от него подмножества *U* (возможно пустого), представляющего собой множество всех упорядоченных пар (x_i, x_j), где $x_i; x_j \in X$. Элементы множеств *X* и *U* называют соответственно вершинами и дугами (ребрами) графа.

Геометрически граф можно представить в виде множества точек $X = \{x_i\}$ $(i = 1, 2, \ldots, n)$ в *n*-мерном евклидовом пространстве E^n, и множества простых, направленных, самонепересекающихся кривых $\vec{U} = \{\vec{u}_k\}$ $(k = 1, 2, \ldots, r)$, соединяющих $x_i, x_j \in X$, которые находятся в некотором отношении друг к другу. То что элемент $x_j \in X$ находится в отношении T_{ij} к элементу $x_i \in X$, отображается на графе соединением элементов x_t и xj линией со стрелкой в направлении от x_i к x_j. Такие соединения вершин графа с указанием направления называют ориентированными ребрами или дугами и записывают как $\vec{u}_k = (x_i, x_j) \sim x_i T_{ij} x_j$. Граф, в котором все вершины соединены дугами, называют ориентированным, направленным или несимметрическим графом (рис. 5, *а* и *б*).

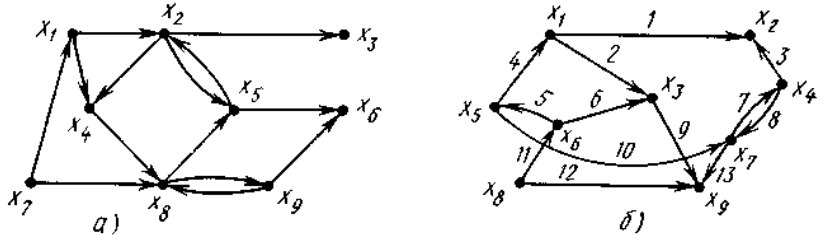


Рис. 5

Аналитически любой ориентированный граф описывается системой алгебраических уравнений, связывающих параметры $x_i \in X$, и

наоборот, любая система алгебраических уравнений может быть представлена в виде направленного графа. Например, граф на рис. 5, *а* определяет следующую систему уравнений:

$$\begin{cases} x_1 = T_{71}x_7; \\ x_2 = T_{12}x_1 + T_{52}x_5; \\ x_3 = T_{23}x_2; \\ x_4 = T_{14}x_1 + T_{24}x_2; \\ x_5 = T_{25}x_2 + T_{85}x_8; \\ x_6 = T_{56}x_5 + T_{96}x_9; \\ x_8 = T_{78}x_7 + T_{48}x_4 + T_{98}x_9; \\ x_9 = T_{89}x_8. \end{cases}$$

Граф, в котором для любых двух вершин $x_i,\ x_j \in X$ справедливо $T_{ij} = T_{ji}$, называют неориентированным, ненаправленным или симметрическим графом. В таком графе вершины x_i и x_j соединены ненаправленной кривой, называемой неориентированным ребром или просто ребром графа (рис. 6).

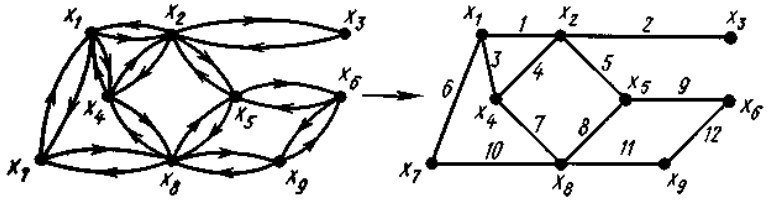


Рис. 6

В дальнейшем рассматриваются в основном неориентированные графы, так как при решении большинства задач распознавания статических объектов существенным является лишь наличие или отсутстине связей между отдельными элементами распознаваемого объекта. При распознавании динамических объектов, наоборот, пользуются только направленными графами.

Две вершины $x_i,\ x_j \in X$ считаются смежными, если они определяют ребро (дугу), и, соответственно, два различных ребра (дуги) смежны, если они имеют общую вершину. Иными словами, вершина x_j смежна x_i, если $x_j \in \Gamma x_i$, где Γx_i—отображение x_i на множестве X.

Так как отображение Γx_i представляет собой совокупность всех игршин графа $X_i \subset X$, смежных x_i, получаем еще один способ зада-

ния графа: граф задан, если задано непустое множество X и отображение Г множества X в X. Обозначим его $G(X, \Gamma)$. При геометричесеой реализации такого графа каждую вершину $x_i \in X$ соединяют со всеми вершинами $x_j \in \Gamma x_i$. Например, для графа $G(X, \Gamma)$ на рис. 6, можно записать

$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$; $\Gamma x_1 = \{x_2, x_4, x_7\}$;
$\Gamma x_2 = \{x_1, x_3, x_4, x_5\}$; $\Gamma x_3 = \{x_2\}$; $\Gamma x_4 = \{x_1, x_2, x_8\}$;
$\Gamma x_5 = \{x_2, x_6, x_8\}$; $\Gamma x_6 = \{x_5, x_9\}$; $\Gamma x_7 = \{x_1, x_8\}$;
$\Gamma x_8 = \{x_4, x_5, x_7, x_9\}$;
$\Gamma x_9 = \{x_6, x_8\}$.

Вершина x_i инцидентна ребру (дуге) u_j, если она является началом или концом ребра (дуги). Аналогично утверждение, что ребро (дуга) u_j инцидентно вершине x_i если оно входит или выходит из этой вершины. Число ребер (дуг), инцидентных некоторой вершине x_i, называют степенью вершины и обозначают $\rho(x_i)$. Для графа нарис. 6 можно записать $\rho(x_1) = \rho(x_4) = \rho(x_5) = 3$; $\rho(x_2) = \rho(x_8) = 4$;
$\rho(x_3) = 1$; $\rho(x_6) = \rho(x_7) = \rho(x_9) = 2$.

Учитывая, что каждое ребро неориентированного графа инцидентно двум вершинам, получим выражение, связывающее число ребер графа со степенями вершин:

$$\sum_{i=1}^{n} \rho(x_i) = 2|U|,$$

где $n = |X|$— число вершин графа; $|U|$— число ребер графа. Из этого выражения следует, что число вершин с нечетной степенью в графе четное, так как при опускании всех вершин x_i с четными степенями $\rho(x_i)$ сумма слева остается четной.

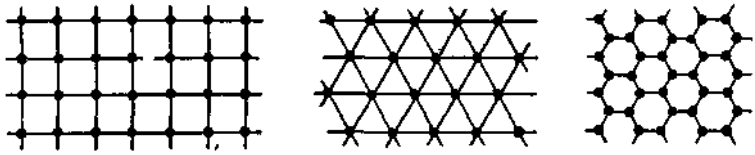
Вершину, неинцидентную никакому ребру (дуге), называют изолированной. Граф, состоящий только из изолированных вершин $(U = \varnothing)$, называют нуль - графом и обозначают G_0.

При использовании графов для распознавания объектов в отдельных случаях целесообразно введение связи вершины самой с собой, т. е. $u = (x_i, x_i) \sim x_i T_{ii} x_i$. Такую связь называют петлей. Если граф $G(X, U)$ имеет петлю при вершине $x_i \in X$, то $x_i \in \Gamma x_i$. Отсюда следует, что необходимым и достаточным условием отсутствия петель в графе является $\forall x_i \in X[x_i \notin \Gamma x_i]$. При геометрической реализации графа петля представляется замкнутой дугой, начинающейся и оканчивающейся в одной и той же вершине x_i и не проходящей через другие вершины графа. Так как концевые точки петли совпадают, то петлю считают неориентированной.

Граф называют конечным, если число его ребер конечно, и бесконечным, если число его ребер бесконечно. Конечный граф, у которого отсутствуют петли и изолированные вершины, называют регулярным.

Граф называют однородным степени *t*, если степени всех его вершин равны *t*, т. е. $\rho(x_1) = \rho(x_2) = \ldots = \rho(x_n) = t$. Число ребер в однородном графе степени *t* равно $|U| = 0{,}5|X|t$.

На рис. 7, *а—в* приведены примеры трех бесконечных однородных графов.



а) *б)* *в)*

Рис. 7

Эти графы находят широкое применение в задачах трассировки соединений распознаваемых объектов, так как их использование позволяет разбивать коммутационное поле объектов на элементарные ячейки одинаковой формы.

Граф, все вершины которого попарно смежны, называют сильносвязным или полным графом (рис. 8, *а, б*).

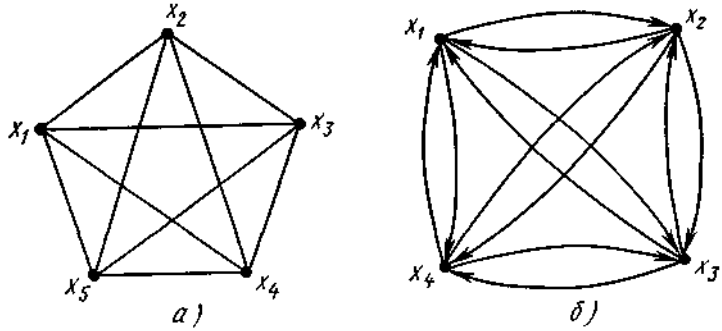


Рис. 8

Так как для полного неориентированного графа степень каждой вершины $\rho(x_i)$ равна n — 1, то число его ребер определяют из соотношения

$$|U| = n\,\frac{n-1}{2}.$$

Полный граф, у которого при каждой вершине имеется петля, называют плотным графом.

Граф, в котором, перемещаясь по ребрам из вершины в вершину, можно попасть в каждую вершину, называют связным (рис. 6 и 8, *а)*. Граф, состоящий из отдельных фрагментов, называют несвязным, состоящим из отдельных компонент связности (рис. 9).

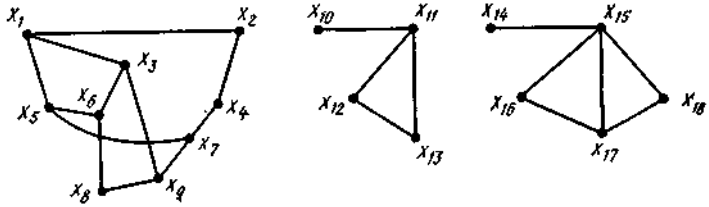


Рис. 9

Число, характеризующее разность между числом вершины графа n и числом компонент связности p, называют рангом графа и обозначают $R(G)$, т. е. $R(G) = n - p$. Для графа на рис. 9 $R(G) = 15$.

При изображении графа в виде геометрической фигуры существует большая свобода в размещении вершин графа в пространстве и в выборе формы соединяющих их ребер (дуг). Следовательно, один и тот же граф может иметь различную геометрическую реализацию. Два графа G и G' изоморфны, если они имеют одинаковое число вершин и если каждой паре вершин, соединенных ребром (дугой), в одном графе соответствует такая же пара вершин, соединенных ребром (дугой), в другом графе (рис. 10).

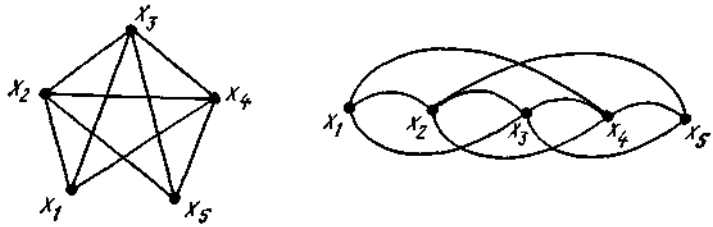


Рис. 10

Решение ряда задач распознавания связано с изоморфными преобразоваииями графа, т. е. с построением графа, изоморфного заданному. Например, с целью сокращения числа пересечений ребер графа, изображенного на плоскости, уменьшения суммарной длины ребер графа и т. п.

Граф, у которого существует хотя бы одна пара вершин, соединенная m ребрами (дугами в одном направлении), называют мультиграфом (рис. 11, *а*).

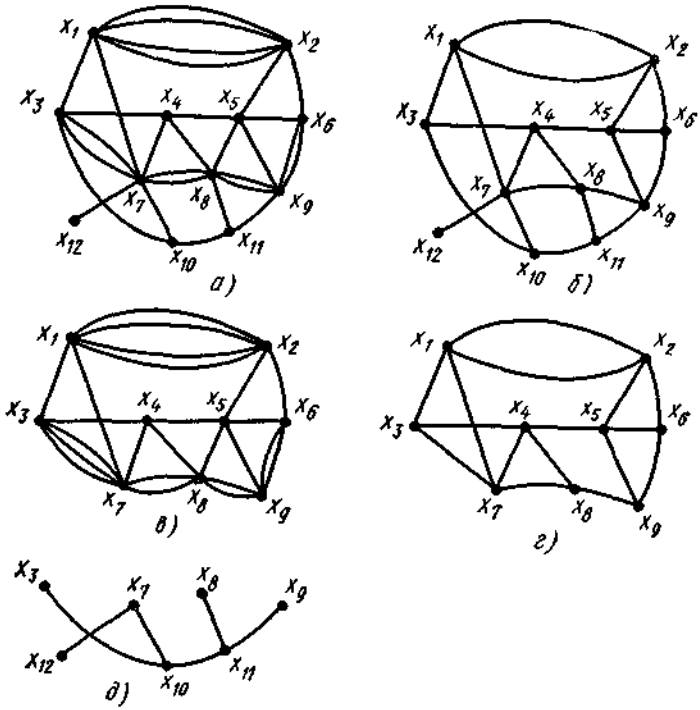


Рис. 11

При этом ребра (дуги), связывающие одну и ту же пару вершин, считают кратными, а максимальное число кратных ребер (дуг) в графе — мультичислом графа. Для мультиграфа на рис. 11, *а* мультичисло равно четырем.

Если в графе $G(X, \Gamma)$ опущены некоторые ребра, а число вершин осталось прежним, то полученный граф $G(X, \Gamma_p)$ называют частичным графом $G(X, \Gamma)$. На рис. 11, *б* показан частичный граф мультиграфа рис. 11, *а*.

Граф $G(X_1, \Gamma_a)$ называют подграфом $G(X, \Gamma)$, если он получается из $G(X, \Gamma)$ опусканием некоторых веошин и инцидентных им ребер, т.е.

$X_1 \subset X;\ \forall x_i \in X_1\ [\Gamma_a x_i = \Gamma x_i \cap X_1]$. На рис. 11, *в* приведен приведен подграф мультиграфа рис. 11, *а*

Частичным подграфом $G(X, \Gamma)$ считают граф $G(X_1, \Gamma_{pa})$, который получается из графа $G(X, \Gamma)$ с помощью операций свойственных одновременно и частичным графам, и подграфам, т. е. когда имеет место опускание в $G(X, \Gamma)$ и вершин с инцидентными им ребрами, и некоторых отдельных ребер: $X_1 \subset X;$ $\forall x_i \in X_1 [\Gamma_{pa} x_i \subset \Gamma x_i \cap X_1]$. На рис. 11, *г* изображен частичный подграф мультиграфа рис. 11, *а*.

Для части графа $G(X, \Gamma)$ существует единственная дополняющая часть (дополнение), состоящая из всех ребер графа $G(X, \Gamma)$, которые не принадлежат этой части. Например, граф $G(X', \Gamma'_a)$ является дополнением подграфа $G(X_1, \Gamma_a)$ графа $G(X, \Gamma)$, если $X = X_1 \cup X';\ \forall x_i \in X [\Gamma x_i = \Gamma_a x_i \cup \Gamma_a x_i]$. На рис. 11, *д* показано дополнение подграфа $G(X_1, \Gamma_a)$ мультиграфа $G(X, \Gamma)$, изображенных на рис. 11, *б* и *а* соответственно. Аналогично получают дополнения частичного графа и частичного подграфа.

Циклом называют последовательность ребер $u_1 = (x_1, x_i), \ldots,$ $u_k=(x_j, xi)$, при которой в результате обхода иершин графа $x_1, x_i, \ldots, x_j$ по этим ребрам возвращаются в исходную вершину x_1. Каждое ребро графа встречается в цикле не более одного раза, в то время как вершины могут повторяться и несколько раз. Цикл считают простым, если в нем нет повторяющихся иершин, и сложным, если такие имеются. Цикл называют элементарным, если он не содержит в себе никаких других циклов. Цикл считают минимальным, если он включает минимальное число ребер, и максимальным, если он содержит максимальное число ребер графа.

Последовательность ребер, получаемая при переходе от одной вершины графа к другой, называют цепью. Таким образом, цикл — это замкнутая цепь.

Большое значение в задачах распознавания имеют эйлеровы и гамильтоновы циклы.

Эйлеров цикл — это цикл, в котором содержатся все ребра графа. Граф, имеющий такой цикл, называют эйлеровым графом (рис. 12).

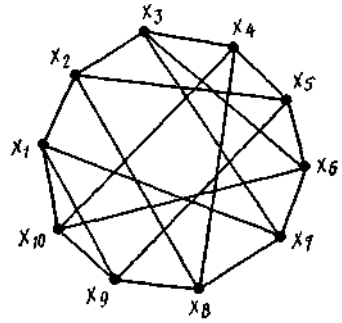


Рис. 12

Необходимым и достаточным условием наличия в конечном связном графе $G(X, U)$ эйлерова цикла является четность степеней всех его вершин.

Задача отыскания эйлерова цикла имеет большое прикладное значение, например, при выборе наиболее рационального пути движения головки копира, резца и т. п., позволяя осуществлять обход контура сложной геометрической фигуры без отведения рабочего инструмента, исключив таким образом холостые ходы.

Гамильтонов цикл определяют для конечных связных графов аналогичным образом, но только по отношению к вершинам: цикл называют гамильтоновым, если он проходит через каждую вершину графа один раз. На рис. 13, *а—в* приведены примеры гамильтоновых циклов для нескольких простых графов.

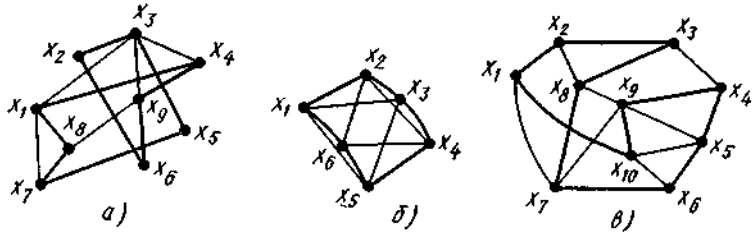


Рис. 13

Несмотря на некоторое сходство в определениях эйлеровых и гамильтоновых циклов, используемые для их отыскания методы имеют мало общего. Критерий существования эйлерова цикла был установлен просто, для гамильтонова цикла такого общего правила не известно. Более того, иногда для конкретного графа бывает затруднительно сказать, имеет ли он такой цикл или нет. Существуют лишь частные

критерии наличия в графе гамильтонова цикла, например критерий Дирака: граф имеет гамильтонов цикл, если сумма локальных степеней двух любых вершин графа больше или равна числу вершин, т. е.

$$\forall x_i,\ x_j \in X\ [\rho(x_i) + \rho(x_j) \geqslant n].$$

Задача отыскания гамильтонова цикла в графе имеет большое прикладное значение, например, для распознавания и построения плоского изображения графа, при решении коммуникационных задач.

Связный неориентированный граф, не содержащий циклов, называют деревом. Несвязный граф без циклов, отдельные компоненты связности которого являются деревьями, называют лесом. Очевидно, что любое дерево, построенное на n вершинах, содержит $r_t = n - 1$ ребер, а лес, состоящий из p компонент связности имеет $r_f = n - p$ ребер. Следовательно, для любого графа, содержащего циклы, имеет место соотношение $r > n - p$.

Число различных деревьев, которые можно построить на n вершинах, $t_n = n^{n-2}$. Заметим, что среди деревьев, учитываемых данным выражением, многие являются изоморфными, т. е. отличаются только нумерацией вершин. Так, при n=20 число деревьев t_n=20^{18}=232144·10^{18}, среди которых только 823 065 неизоморфны. Число неизоморфных деревьев определяют путем использования методов комбинаторики. На рис. 14, *а—в* показаны три различных дерева, которые можно построить на множестве из трех вершин. Все приведенные деревья изоморфны друг другу, т. е. на трех вершинах можно построить только одно неизоморфное дерево.

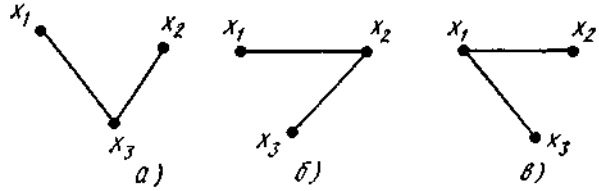


Рис. 14

Представление распознаваемых объектов с помощью графов. При распознавании таких объектов, как отдельные виды конструкций РЭА, применяют в основном ненаправленные графы. Например, принципиальная электрическая схема интерпретируется графом, в котором каждому конструктивному элементу ставится в однозначное соответствие вершина, а электрическим связям — ребра графа. Это позволяет абстрагироваться от конкретных схем и, перейдя к их

математическим моделям — графам, разрабатывать эффективные методы поиска оптимальных конструктивных решений.

Рассмотрим схему транзисторного усилителя на рис. 15.

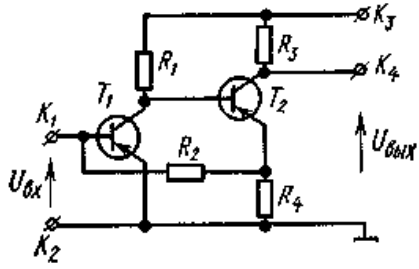


Рис. 15

Представим конструкцию в виде произвольного неориентированного графа *G(X, U),* у которого

$$X = \{K_1, K_2, K_3, K_4, T_1, T_2, R_1, R_2, R_3, R_4\}.$$

U — множество всех электрических связей элементов конструк ции (рис 16, *а*).

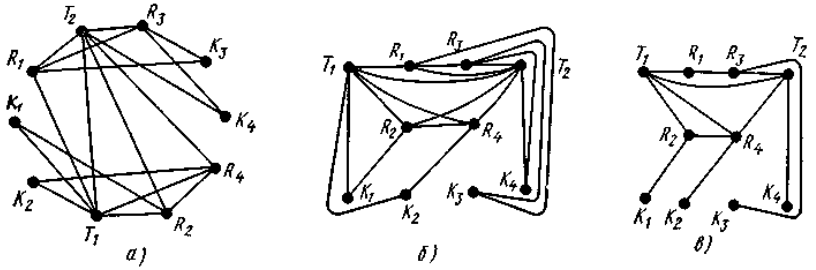


Рис. 16

Выполним изоморфные преобразования, используя в качестве целевой функции минимум числа пересечений ребер графа, вложенного в плоскость. Смещаем вершины графа в фиксированные позиции на плоскости (отдельные вершины графа могут быть предварительно закреплены исходя из конструктивных соображений. В нашем случае такими вершинами являются K_1, K_2, K_3 и K_4, взаимное местоположение которых определяется контактами разъема) (рис.16,*б)* и исключаем те из дублирующих ребер, которые имеют максимальную протяженность или большое число пересечений с другими ребрами (рис. 16, *в).* Заметим, что в получившемся графе число вершин с локальной степенью, большей числа выводов

соответствующих им элементов, не должно превышать числа узлов принципиальной схемы рис. 15. Найденное в результате преобразований местоположение вершин графа (рис. 16, *б*) соответствует рациональному взаимному расположению элементов на плате. Печатная плата усилителя с установленными на ней конструктивными элементами приведена на рис. 17.

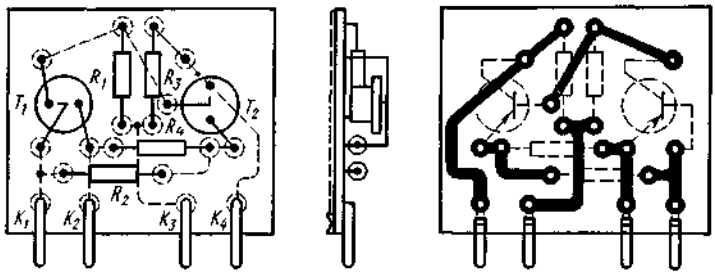


Рис. 17

Рассмотренный пример иллюстрирует основные принципы интерпретации принципиальной электрической схемы модуля неориентированным графом, выполнения над ним операций удаления отдельных ребер и изоморфных преобразований, направленных на оптимизацию размещения элементов на печатной плате, и последующего перехода от полученного в результате этих преобразований графа к рациональному расположению элементов схемы на коммутационном поле модуля.

4.2. Способы задания графов

Уже отмечалось, что произвольный граф можно задать совокупностью двух множеств: X —множества вершин и U —множества ребер (дуг) графа или множества X и отображением Γ множества X в X. Условные обозначения таких графов — $G(X, U)$ и $G(X, \Gamma)$ соответственно. Другой удобной формой описания графов является представление их с помощью матриц, методика формальной обработки которых хорошо разработана.

Матрица смежности. Если задан граф $G(X, U)$, то ему можно поставить в соответствие квадратную матрицу смежности $\mathbf{A} = \|a_{ij}\|_{n \times n} (n = |X|)$, общий элемент которой

$$a_{ij} = \begin{cases} m(x_i, x_j), \text{ если } (x_i, x_j) \in U; \\ 0, \text{ если } (x_i, x_j) \notin U, \end{cases}$$

где $m(x_i, x_j)$— кратность ребер между вершинами x_i и x_j.

Для графа на рис. 6 матрица смежности

$$\mathbf{A} = \begin{array}{c|ccccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \\ \hline x_1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ x_2 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ x_3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_4 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ x_5 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ x_6 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ x_7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ x_8 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ x_9 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}$$

Правильность составления матрицы смежности легко проверить: лля неориентированного графа сумма элементов в каждом i-ом столбце или строке соответствует степени вершины x_i. Если элемент матрицы a_{ii}, расположенный на главной диагонали, отличен от нуля, то это свидетельствует о наличии петель в вершине x_i. Заметим, что для неориентированного графа матрица смежности **A** симметрична относительно главной диагонали,так как $a_{ij} = a_{ji}$.

При решении целого класса задач распознавания приходится оперировать матрицами, которые строятся аналогично матрицам смежности, но значения их элементов определяются мерой (весом), связанной с ребром (дугой) графа. В теории распознавания широко используют две разновидности таких матриц: матрицу весовых соотношений и матрицу длин.

Матрица весовых соотношений. Это квадратная матрица $\mathbf{C} = ||c_{ij}||_{n \times n}$, общий элемент которой

$$c_{ij} = \begin{cases} T_{ij}, \text{ если } x_j \text{ смежна } x_i; \\ 0, \text{ если } x_j \text{ не смежна } x_i, \end{cases}$$

где T_{ij} — вес связи (x_i, x_j).

Применение матриц весовых соотношений позволяет учитывать различные требования к сокращению длины тех или иных электричесеих соединений в конструкциях РЭА, условия тепловой и электромагнитной совместимости отдельных элементов схемы.

Матрица длин. Это квадратная матрица $\mathbf{D} = ||d_{ij}||_{n \times n}$, общий элемент которой

$$d_{ij} = \begin{cases} l_{ij}, \text{ если } x_i \text{ и } x_j \text{ смежны;} \\ 0, \text{ если } x_j \text{ не смежна } x_i, \end{cases}$$

где l_{ij} —длина ребра (x_i, x_j). В евклидовой метрике расстояние между двумя точками на плоскости

$$l_{ij} = \sqrt{(s_i - s_j)^2 + (t_i - t_j)^2},$$

где s_i, t_i и s_j, t_j —координаты вершин x_i и x_j соответственно.

Это выражение неудобно для использования в машинных программах, так как извлечение квадратного корня на ЭВМ требует больших затрат времени. Поэтому часто пользуются линейной метрикой, тем более, что при решении многих задач распознавания она вполне оправдана. Например, при проектировании многослойных печатных плат трассировка соединений ведется в каждом из слоев во взаимно перпендикулярных направлениях, в связи с чем длина электрических связей между элементами с координатами (s_i, t_i) и (s_j, t_j)

$$l_{ij} = |s_i - s_j| + |t_i - t_j|.$$

Линейная метрика оказывается непригодной для решения задач, в которых имеет место вычисление производных по координатам (оптимизация нелинейных функций). В этом случае расстояние между двумя точками представляется в виде степенной функции:

$$l_{ij} = (s_i - s_j)^k + (t_i - t_j)^k,$$

где k — показатель степени (как правило, $k = 2$).

Матрицу длин используют при решении задач оптимизации размещения конструктивных элементов на плате, когда одним из критериев качества является минимум суммарной длины соединений.

Матрица инцидентности. Данная матрица представляет собой прямоугольную матрицу $\mathbf{S} = ||s_{ij}||_{n \times r} (n = |X|, r = |U|)$, строки которой соответствуют вершинам, а столоцы — реорам (дугам) графа $G(X, U)$. Общий элемент этой матрицы для неориентированного графа

$$s_{ij} = \begin{cases} 1, \text{ если } u_j \text{ инцидентно } x_i; \\ 0, \text{ если } u_j \text{ неинцидентно } x_i. \end{cases}$$

Для графа на рис. 6

$$S = \begin{array}{c|cccccccccccc} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} & u_{12} \\ \hline x_1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_2 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ x_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ x_7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ x_8 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ x_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

Правильность составления матрицы **S** легко проверить: число единиц в i-й строке матрицы соответствует степени вершины x_i графа, а число единиц в каждом столбце —двум, так как каждое ребро соединяет две вершины графа. Единственное исключение составляет петля, дважды инцидентная одной и той же вершине. Столбец, соответствующий петле, состоит из нулей, в результате чего матрица **S** не указывает на существование петель. Поэтому при изучении свойств графа с помощью этой матрицы необходимо исключить из него петли.

Для несвязного графа рис. 9

$$S = \begin{array}{c|c|c|c} S_{11} & 0 & 0 \\ \hline 0 & S_{22} & 0 \\ \hline 0 & 0 & S_{33} \end{array}$$

где S_{11}, S_{22} и S_{33} — подматрицы, соответствующие трем связным компонентам несвязного графа $G(X, U)$. Иными словами, для несвязного графа все подматрицы S_{ij}, кроме тех, которые находятся на главной диагонали $(i = j)$, являются нулевыми.

Граф однозначно задается матрицами смежности и инцидентности. В свою очередь, каждая из этих матриц полностью определяет граф. Существуют простые приемы перехода от одной матрицы к другой.

Кроме матриц смежности и инцидентности, для описания графа можно воспользоваться матрицами контуров, сечений и т. п.

4.3. Действия над графами

Над графами, так же как и над множествами, можно выполнять операции объединения, пересечения, вычитания, произведения, декартова произведения и сложения графов. Среди них чаще в теории распознавания используют первые три операции, которые и рассмотрим более подробно.

Объединение графов условно записывают следующим образом:

$$G(X, \Gamma) = G_1(X_1, \Gamma_1) \cup G_2(X_2, \Gamma_2),$$

где $G(X, \Gamma)$ — граф, полученный в результате объединения исходных графов $G_1(X_1, \Gamma_1)$ и $G_2(X_2, \Gamma_2)$.

Объединение осуществляют по следующим правилам:

1) вершинами графа $G(X, \Gamma)$ является объединение вершин исходных графов: $X = X_1 \bigcup X_2$;

2) отображение для каждой вершины графа $G(X, \Gamma)$ получают путем объединения отображений этой вершины для исходных графов: $\Lambda x_i \in X[\Gamma x_i = \Gamma_1 x_i \cup \Gamma_2 x_i]$.

Например, в результате объединения графов на рис. 18, *а, б* находим граф рис. 18, *в*, у которого

$$X = X_1 \cup X_2 = \{x_1, x_2, x_3, x_4, x_5, x_6\};$$

$$\Gamma x_1 = \Gamma_1 x_1 \cup \Gamma_2 x_1 = \{x_2, x_6\} \cup \{x_2, x_4\} = \{x_2, x_4, x_6\};$$

$$\Gamma x_2 = \Gamma_1 x_2 \cup \Gamma_2 x_2 = \{x_1, x_3, x_4, x_6\} \cup \{x_1, x_3\} = \{x_1, x_3, x_4, x_6\};$$

$$\Gamma x_3 = \Gamma_1 x_3 \cup \Gamma_2 x_3 = \{x_2, x_4\} \cup \{x_3, x_4\} = \{x_2, x_4\};$$

$$\Gamma x_4 = \Gamma_1 x_4 \cup \Gamma_2 x_4 = \{x_2, x_3, x_5, x_6\} \cup \{x_1, x_3\} = \{x_1, x_2, x_3, x_5, x_6\};$$

$$\Gamma x_5 = \Gamma_1 x_5 \cup \Gamma_2 x_5 = \{x_4, x_6\} \cup \{\varnothing\} = \{x_4, x_6\};$$

$$\Gamma x_6 = \Gamma_1 x_6 \cup \Gamma_2 x_6 = \{x_1, x_2, x_4, x_5\} \cup \{\varnothing\} = \{x_1, x_2, x_4, x_5\}.$$

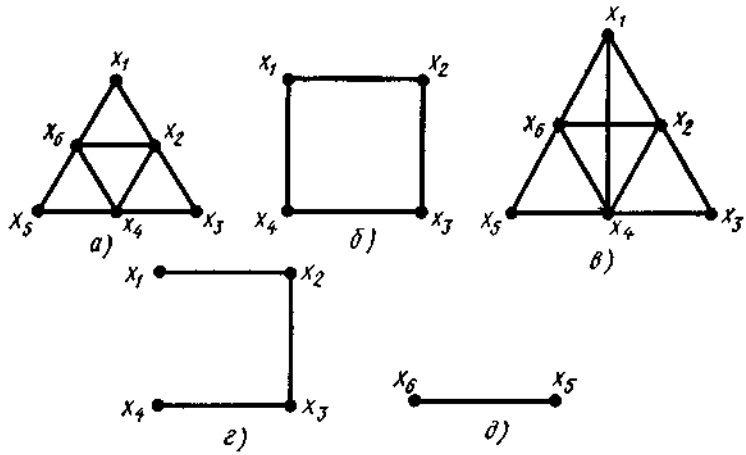


Рис. 18

Пересечение графов условно записывают в виде

$$G(X, \Gamma) = G_1(X_1, \Gamma_1) \cap G_2(X_2, \Gamma_2),$$

где $G(X, \Gamma)$ — граф, полученный в результате пересечения исходных графов $G_1(X_1, \Gamma_1)$ и $G_2(X_2, \Gamma_2)$.

Пересечение выполняют по следующим правилам:

1) вершинами графа $G(X, \Gamma)$ является пересечение вершин исходных графов: $X = X_1 \cap X_2$;

2) отображение для каждой вершины графа $G(X, \Gamma)$ получают в результате пересечения отображений этой вершины для исходных графов: $\forall x_i \in X[\Gamma x_i = \Gamma_1 x_i \cap \Gamma_2 x_i]$. Например, в результате пересечения графов на рис. 18, *а, б* получаем граф рис. 18, *г,* у которого

$$X = X_1 \cap X_2 = \{x_1, x_2, x_3, x_4\};$$

$$\Gamma x_1 = \{x_2\}; \Gamma x_2 = \{x_1, x_3\}; \Gamma x_3 \{x_2, x_4\};$$

$$\Gamma x_4 = \{x_3\}.$$

Вычитание графов условно записывают следующим образом:

$$G(x, \Gamma) = G_1(X_1, \Gamma_1) \setminus G_2(X_2, \Gamma_2),$$

где $G(X, \Gamma)$ — граф, полученный в результате вычитания исходных графов $G_1(X_1, \Gamma_1)$ и $G_2(X_2, \Gamma_2)$.

Вычитание производят по следующим правилам:

1) вершинами графа $G(X, \Gamma)$ являются вершины графа $G_1(X_1, \Gamma_1)$, за исключением вершин, общих для исходных графов: $X = X_1 \setminus X_2$;

2) отображением для каждой вершины графа $G(X, \Gamma)$ является пересечение множества вершин этого графа и отображения той же вершины в графе $G_1(X_1, \Gamma_1) : \forall x_i \in X[\Gamma x_i = X \cap \Gamma_1 x_i]$. Например, в результате вычитания графов на рис. 18, *а, б* получаем граф (рис. 18, *д),* у которого

$$X = X_1 \backslash X_2 = \{x_5, x_6\};$$

$$\Gamma x_5 = X \cap \Gamma_1 x_5 = \{x_6\}; \ \Gamma x_6 = X \cap \Gamma_1 x_6 = \{x_5\}.$$

4.4. Характеристические числа графа и их применение

Рассмотрим некоторые характеристические числа графа, позволяющие в ряде случаев упростить решение задач исследования топологических свойств графа. К таким числам, не зависящим от изоморфных преобразований графа, относятся: цикломатическое $\nu(G)$ и хроматическое $k(G)$ числа, числа внутренней $\alpha(G)$ и внешней $\beta(G)$ устойчивости графа.

Цикломатическое число. При рассмотрении произвольного неориентированного графа $G(X, U)$ без петель с $|X| = n$ и $|U| = r$, состоящего из p компонент связности, величину $\nu(G) = r - n + p$ называют цикломатическим числом графа. Иногда вводят понятие ранга графа $R(G) = n - p$. В этом случае цикломатическое число $\nu(G) = r - R(G)$.

Цикломатическое число графа указывает то наименьшее число ребер, которое необходимо удалить из данного графа, чтобы получить дерево (для связного графа) или лес (для несвязного графа), т. е. добиться отсутствия у графа циклов. Нетрудно показать, что цикломатическое число всегда неотрицательно.

Определим основное свойство цикломатического числа, сформулировав его с помощью теоремы.

Теорема 1. *Цикломатическое число мультиграфа равно максимальному числу независимых циклов.* (Циклы считают независимыми, если в каждом из них содержится по крайней мере одно ребро, не входящее в другие циклы графа).

Доказательство. Осуществим операцию последовательного удаления ребер мультиграфа $G(X, U)$, инцидентных вершинам $x_i \in X \subset \rho(x_i) = 1$. Так как эти ребра графа не могут входить ни в один из циклов $G(X, U)$, общее число циклов и значение $\nu(G)$ не изменятся.

Выделим для полученного частичного графа $G_p(X, U_p)$ какой-либо каркас $G_F\ (X_F, U_F)$, каждая компонента связности которого есть дерево, содержащее все вершины соответствующей компоненты связности $G_p\ (X, U_p)$, т. е. $p(G_F) = p(G_p)$; $|U_F| = |U_p| - \nu(G_p)$; $\nu(G_F) = 0$. При этом каждое ребро, не принадлежащее $G_F(X_F, U_F)$, образует с отдельными ребрами каркаса графа простой цикл.

Действительно, для каждой пары вершин x_i и x_j графа $G_p(X, U_p)$, соединенных ребром u_q, можно найти компоненту $G'_F(X'_F, U'_F)$ каркаса $G_F(X_F, U_F)$, которая содержит эти вершины.

Так как $G'_F(X'_F, U'_F)$—дерево, то в нем имеется только одна цепь, притом простая, соединяющая x_i с x_j, и она вместе с ребром u_q образует простой цикл, который является единственным независимым циклом. Общее число таких циклов определяется числом ребер частичного графа $G_p(X, U_p)$, не вошедших в его каркас, т. е. числом ребер исходного мультиграфа $G(X, U)$, не вошедших в произвольный каркас мультиграфа, так как дополнение частичного графа $G_p(X, U_p)$ до $G(X, U)$ представляет собой лес.

Все эти циклы независимы, причем каждое из ребер каркаса $G_F(X_F, U_F)$ входит в один из этих циклов. Любой другой цикл будет содержать только ребра, рассмотренные в простых циклах мультиграфа $G(X, U)$.

Следовательно, максимальное число независимых циклов в $G(X, U)$ равно числу простых циклов (числу ребер, не вошедших в каркас мультиграфа) т.е. $\nu(G)$.

Знание цикломатического числа оказывается полезным при распознавании топологии объектов. Например, если рассмотреть конструкцию печатной платы, то схему электрических соединений можно интерпретировать графом $G(X, U)$, где X — множество областей контактных площадок, внутри которых проведение проводников запрещено, а U —множество трасс (рис. 1).

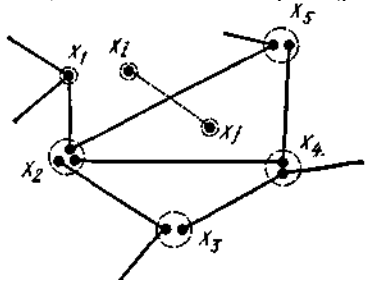


Рис. 1

При этом все коммутационное пространство разбивается проведенными соединениями на отдельные, локально замкнутые области $Q = \{q_1, q_2, \ldots, q_g\}$. Очевидно, что любые две вершины $x_i, x_j \in X$, находящиеся в различных областях $q_s, q_t \in Q$, не могут быть соединены ребром $u_k \in U$ без пересечения ребер (соединений), ограничивающих области q_s и q_t. В связи с этим возникает необходимость распознавания непопадания смежных вершин в различные изолированные области. Цикломатическое число $\nu(G)$ позволяет распознать число таких локально замкнутых областей и перейти к решению задачи рационального перераспределения ребер графа $G(X, U)$.

Хроматическое число. Пусть задан граф $G(X, U)$ без петель. Разобьем множество его вершин на k непересекающихся подмножеств

$$X_1, X_2, \ldots, X_k, X = \bigcup_{i=1}^{k} X_i, \forall X_i, X_j \subset X [X_i \cap X_j = \varnothing] \quad (1)$$

так, чтобы любые две смежные вершины $x_s, x_t \in X$ принадлежали разным подмножествам, т. е. чтобы ребра графа $G(X, U)$ соединяли вершины из разных подмножеств:

$$\forall x_s \in X [x_s \in X_i \rightarrow \Gamma x_s \not\subset X_i]. \quad (2)$$

Отметим все вершины X индексами $1, 2, \ldots, k$ (т. е. раскрасим вершины k цветами), причем вершины внутри каждого подмножества X_i помечают одним индексом (раскрашивают одним цветом). Подмножества формируют таким образом, чтобы концы любого ребра графа имели различные индексы.

Наименьшее возможное число подмножеств, получаемое в результате такого разбиения вершин графа $G(X, U)$, называют хроматическим числом графа $k(G)$, граф $G(X, U)$ именуют k-хроматическим, а выражения (1) и (2) — хроматическим разложением X.

Особое значение имеет частный вид k-хроматического графа — дихроматический или двудольный граф, для которого множество вершин X можно разбить на два непересекающихся подмножества X_1 и X_2 так, чтобы никакое ребро не соединяло бы вершины одного и того же подмножества, т. е.

$$X = X_1 \cup X_2, X_1 \cap X_2 = \varnothing,$$

$$\forall x_i, x_j \in X [(x_i, x_j) = u_f \in U \Rightarrow (x_i \in X_1 \,\&\, x_j \in X_2) \vee (x_i \in X_2 \,\&\, x_j \in X_1)].$$

Такие графы называют графами Кенига и обозначают

$$G(X_1,\ X_2,\ U).$$

Критерий бихроматичности произвольного графа формулируется теоремой Кенига, согласно которой, обыкновенный граф *G(X, U)* является бихроматическим тогда и только тогда, когда он не содержит циклов нечетной длины.

Если граф *G(X, U)* — дерево, т. е. в нем полностью отсутствуют циклы (существуют лишь циклы нулевой длины), то он является бихроматическим графом и может быть представлен в виде двудольного графа (рис. 2).

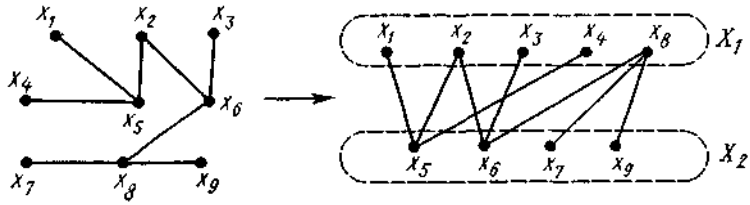


Рис. 2

На рис. 3 приведен пример бихроматического графа $G(X_1,\ X_2,\ U)$, у которого $X_1 = \{x_1,\ x_2,\ x_3,\ x_4\}$, $X_2 = \{x_5, x_6, x_7, x_8\}$.

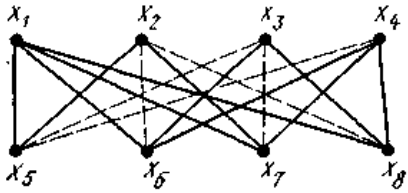


Рис. 3

Граф Кенига $G(X_1,\ X_2,\ U)$ называют полным, если каждая вершина $x_i \in X_1$ смежна с каждой вершиной $x_j \in X_2$, и наоборот. При добавлении к графу рис. 3 ребер $(x_2,\ x_6)$, $(x_2,\ x_8)$, $(x_3,\ x_5)$, $(x_3,\ x_7)$ и $(x_4,\ x_5)$ получим полный бихроматический граф. Очевидно, что подмножество X_1 множества вершин X можно раскрасить в один цвет, а X_2 — в другой. Число ребер полного бихроматического графа

$$G(X_1,\ X_2,\ U)\ r = |U| = |X_1||X_2|.$$

При составлении целого ряда алгоритмов распознавания используют операцию удаления некоторых вершин и ребер графа. При этом особое значение приобретает понятие критического графа. Граф $G(X, U)$ называют критическим, если удаление любой вершины $x_i \in X$ с инцидентными ей ребрами уменьшает хроматическое число графа. Критическим 1-хроматическим графом является одна вершина; критическим 2-хроматическим — две вершины, соединенные ребром; критическим 3-хроматическим — простой цикл нечетной длины, так как при удалении из него любой вершины с инцидентными ей ребрами получим двудольный граф. Очевидно, что полный граф всегда является критическим, причем его хроматическое число равно $k(G) = |X| = n$.

В отличие от цикломатического числа определение хроматического числа осуществляется с помощью сравнительно сложных алгоритмов, в основу большинства которых положены методы целочисленного линейного программирования.

Оценка хроматического числа через число вершин графа $G(X, U)$ очевидна: $1 \leqslant k(G) \leqslant |X| = n$. При этом нижняя граница соответствует пустым, а верхняя — полным графам.

Более точную оценку границ, в которых находится значение $k(G)$ для произвольного связного графа $G(X, U)$ с $|X| = n$; $|U| = r$ дает теорема, сформулированная А. П. Ершовым и Г. И. Кожухиным, согласно которой при отсутствии в графе петель и кратных ребер

$$-\left[-\frac{n}{\left[\frac{n^2-2r}{n}\right]}\left(1-\frac{\left\{\frac{n^2-2r}{n}\right\}}{1+\left[\frac{n^2-2r}{n}\right]}\right)\right] \leqslant$$

$$\leqslant k(G) \leqslant \left[\frac{3+\sqrt{9+8(r-n)}}{2}\right],$$

где квадратные и фигурные скобки соответственно означают взятие целой и дробной части заключенного в них числа.

Особый интерес в теории распознавания представляет оценка хроматического числа через локальные степени графа

$$K(G) \leqslant \rho_{\max}(x) + 1, \quad \rho_{\max}(x) = \max_{x_i \in X} \rho(x_i),$$

которая является точной, так как для полных графов $k(G) = \rho(x) + 1$. Исключив из рассмотрения графы с компонентами связности, представляющими собой полный $[\rho_{\max}(x) + 1]$-вершинный граф, эту оценку можно улучшить: $k(G) \leqslant \rho_{\max}(x)$.

Для простых связных графов оценить величину $k(G)$ можно следующим образом. Сначала выбираем вершину с минимальной локальной степенью и пометим (раскрасим) ее, затем произведем хроматическую раскраску вершин, смежных с данной, и т. д. Например, для графа на рис. 6 п.4.2 сначала выбираем вершину x_3, для которой $\rho(x_3) = 1$, и раскрасим ее в красный цвет. Тогда вершину x_2 можно раскрасить в синий цвет, а вершины x_1 и x_5 — в красный (они не смежны с x_3). Вершины x_6 и x_8 можно раскрасить в синий цвет, а оставшиеся вершины x_4, x_7 и x_9 — в желтый. На раскраску вершин графа пошло три краски $k(G) = 3$.

Знание хроматического числа графа позволяет в ряде случаев упростить алгоритмы, используемые теории распознавания.

Рассмотрим задачу оценки числа слоев многослойной печатной платы. Пусть граф $G(X, U)$ интерпретирует фрагмент коммутационного поля платы, где X — множество областей контактных площадок конструктивных элементов, внутри которых проведение проводников запрощено, а U — множество трасс платы (рис. 4, *а*).

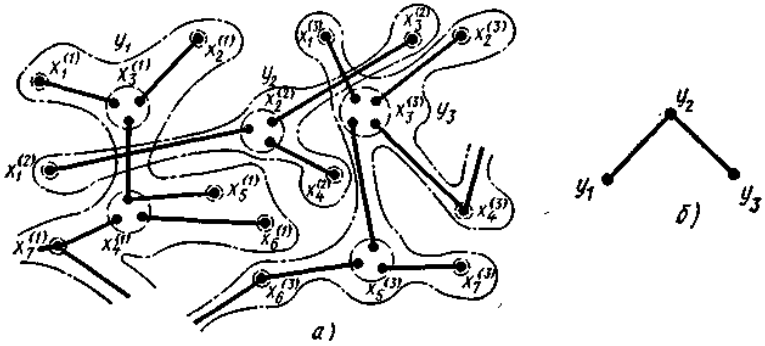


Рис. 4

Построим граф пересечений $Q(Y, V)$, в котором вершинами являются отдельные мпоненты связности (электрические цепи) графа $G(X, U)$, причем y_i, $y_i \in Y$ смежны, если соответствующие им компоненты связности $G_i(X_i, U_i)$ и $G_j(X_j, U_j)$ перекрывают друг друга (рис. 4, *б*). При этом хроматическое число $k(Q)$ графа определит $Q(Y, V)$ верхнюю границу числа коммутационных слоев рассматриваемого фрагмента платы (в нашем примере $k(Q) = 2$). В этом случае цепи, соответствующие вершинам одного цвета графа $Q(Y, V)$, можно располагать в одном слое печатной платы.

Число внутренней устойчивости. Если в графе $G(X, U)$ имеется

некоторое подмножество $F_i \subset X$ несмежных между собой вершин, т. е.

$$\forall x_i \in F_i \, [F_i \cap \Gamma x_j = \varnothing], \qquad (3)$$

то такое подмножество F_i называют внутренне устойчивым.

Рассмотрим семейство $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$ всех внутренне устойчивых подмножеств графа *G(X, U)* и выделим из него наибольшее внутреннее устойчивое подмножество $F_g \subset \mathcal{F}$, мощность которого

$$|F_g| = \max_{F_i \subset \mathcal{F}} |F_i|.$$

Величину, равную мощности наибольшего внутренне устойчивого подмножества, называют числом внутренней устойчивости графа $\alpha(G)$, т. е.

$$\alpha(G) = \max_{F_i \subset \mathcal{F}} |F_i|.$$

Внутреннее устойчивое подмножество считают непополнимым, если его нельзя дополнить ни одной вершиной $x_i \in X$ без потери свойств внутренней устойчивости (3). Таким образом, наибольшее внутренне устойчивое подмножество всегда непополнимо.

В графе рис. 6 п.4.2 можно выделить следующее семейство $\mathcal{F} = \{F_1, F_2, \ldots, F_7\}$ непополнимых внутреннее устойчивых подмножеств:

$F_1 = \{x_3, x_4, x_5, x_7, x_9\}; F_2 = \{x_2, x_6, x_7\}; F_3 = \{x_3, x_4, x_6, x_7\};$
$F_4 = \{x_2, x_7, x_9\}; F_5 = \{x_2, x_6, x_8\}; F_6 = \{x_1, x_3, x_5, x_9\};$
$F_7 = \{x_1, x_3, x_6, x_8\}$. Наибольшее внутренне устойчивое подмножество F_1 содержит пять элементов, следовательно, $\alpha(G) = |F_1| = 5$.

Между хроматическим числом *k(G)* и числом внутренней устойчивости $\alpha(G)$ графа существует связь, характеризующаяся соотношением

$$k(G)\,\alpha(G) \geqslant |X| = n.$$

Действительно, если произвести хроматическую раскраску всех вершин графа *G(X, U),* то вершины, окрашенные в один цвет, представляют собой внутренне устойчивые подмножества. Допустим, что в каждое такое подмножество входит n_i вершин, где $i = 1, 2, \ldots, k(G)$. Так как $n_i \leqslant \alpha(G)$, то

$$n = \sum_{i=1}^{k(G)} n_i \leqslant k(G)\,\alpha(G).$$

Оценку верхней границы числа внутренней устойчивости $\alpha(G)$ для произвольного графа $G(X, U)$ с $|X| = n$ и $|U| = r$ при отсутствии в нем петель и кратных ребер можно определить из соотношения

$$\alpha(G) \leqslant \left[1/2 + \sqrt{\left(n - 1/2\right)^2 - 2r} \right],$$

где квадратные скобки означают взятие целой части заключенного в них числа.

В основу составления многих алгоритмов выделения внутренне устойчивых подмножеств положены методы Х. Магу и Дж. Уэйсмана, использующие вычисление составленного по матрице инциденций $\mathbf{S} = ||s_{ij}||_{n \times r}$ произведения $\Pi_G = \prod_{j=1}^{r} \sum_{i=1}^{n} s_{ij} x_i$, в котором j-й сомножитель представляет собой сумму двух слагаемых, соответствующих вершинам, инцидентным ребру $u_j \in U$ графа $G(X, U)$.

Теорема 2. *Подмножество вершин F_q является внутренне устойчивым тогда и только тогда, когда* Π_G=1 *для системы переменных* $\{x_i^0\}$, *определяемых следующим образом:*

$$x_i^0 = \begin{cases} 1, & \text{если } x_i \notin F_g; \\ 0, & \text{если } x_i \in F_g. \end{cases}$$

Доказательство. Если F_q —внутренне устойчивое подмножество, то каждое ребро $u_j \in U$ должно быть инцидентно хотя бы одной вершине из $X \backslash F_q$. При этом каждый из сомножителей Π_G одержит по крайней мере одно слагаемое $x_i \notin F_g$, а замена x_i на x_i^0 обратит Π_G в единицу.

Если, используя дистрибутивный, ассоциативный и коммутативный законы, сначала раскрыть все скобки в выражении для Π_G, а затем, применив соотношение $x_i^2 = x_i$ и закон поглощения, привести всю сумму к минимальной форме записи, то все слагаемые этой суммы в качестве сомножителей будут иметь вершины из $X \backslash F_q$ (только в этом случае Π_G=1). Следовательно, каждому слагаемому Π_G соответствует внутренне устойчивое подмножество графа $G(X, U)$.

Рассмотрим пример получения непополнимых внутренне устойчивости подмножеств для графа рис. 6 п.4.2. Чтобы не записывать матрицу **S**, заметим, что каждый сомножитель Π_G представляет собой сумму смежных вершин. В нашем примере

$$\Pi_G = (x_1 + x_2)(x_1 + x_4)(x_1 + x_7)(x_2 + x_3)(x_2 + x_4) \times$$
$$\times (x_2 + x_5)(x_4 + x_5)(x_5 + x_6)(x_5 + x_8)(x_6 + x_9) \times$$
$$\times (x_7 + x_8)(x_8 + x_9).$$

Будем последовательно раскрывать скобки, помня, что в силу закона поглощения $(a + b)(a + c) \ \ldots \ (a + q) = a + bc \ldots q$. Тогда, учитывая, что $x_i^2 = x_i$, получим

$$\Pi_6 = (x_1 + x_2\ x_4\ \ x_7)\,(x_2 + x_3\,x_4\,x_5)\,(x_6 + x_5\,x_9)\,(x_8 + x_4\,x_5\,x_7\,x_9) =$$
$$= (x_1\,x_2 + x_1\,x_3\,x_4\,x_5 + x_2\,x_4\,x_7 + x_2\,x_4\,x_5\,x_7) \times$$
$$\times (x_6\,x_8 + x_4\,x_5\,x_7\,x_9 + x_4\,x_5\,x_6\,x_7\,x_9 + x_5\,x_8\,x_9) =$$
$$= (x_1\,x_2\,x_6\,x_8 + x_1\,x_2\,x_4\,x_5\,x_7\,x_9 + x_1\,x_3\,x_4\,x_5\,x_6\,x_8 + x_1\,x_3\,x_4\,x_5\,x_7\,x_9 +$$
$$+ x_1\,x_2\,x_5\,x_8\,x_9 + x_1\,x_3\,x_4\,x_5\,x_8\,x_9 + x_2\,x_4\,x_6\,x_7\,x_8 + x_2\,x_4\,x_5\,x_7\,x_9 + x_2\,x_4\,x_5\,x_7\,x_8\,x_9).$$

В результате получим следующее семейство непополнимых внутренне устойчивых подмножеств:

$$\mathcal{F} = \{F_1,\ F_2,\ \ldots,\ F_7\},\ \text{где}\ F_1 = \{x_3,\ x_4,\ x_5,\ x_7,\ x_9\};$$

$$F_2 = \{x_2,\ x_7,\ x_9\};\ F_3 = \{x_2,\ x_6,\ x_8\};\ F_4 = \{x_3,\ x_4,\ x_6,\ x_7\};$$

$$F_5 = \{x_2,\ x_6,\ x_7\};\ F_6 = \{x_1,\ x_3,\ x_5,\ x_9\};\ F_7 = \{x_1,\ x_3,\ x_6,\ x_8\}.$$

Изложенный метод Ж. Магу для выделения всех непополнимых внутренне устойчивых подмножеств позволяет найти и минимальную раскраску графа. Для этого строят матрицу

$$\mathbf{V} = ||v_{ij}||_{n \times m},$$

где $n = |X|$; $m = |\mathcal{F}|$, каждый элемент которой

$$v_{ij} = \begin{cases} 1, & \text{если } x_i \in F_j; \\ 0, & \text{если } x_i \notin F_j. \end{cases}$$

В нашем случае

$$\mathbf{V} = \begin{array}{c|ccccccc} & F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 \\ x_1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ x_2 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ x_3 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ x_4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ x_5 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ x_6 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ x_7 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ x_8 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ x_9 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

Поставим в соответствие каждой вершине графа *G(X, U)* сумму тех $F_j \subset \mathcal{F}$, в которые она входит, и запишем произведение этих сумм:

$$\Pi_V = (F_6 + F_7)(F_2 + F_3 + F_5)(F_1 + F_4 + F_6 + F_7) \times$$
$$\times (F_1 + F_4)(F_1 + F_6)(F_3 + F_4 + F_5 + F_7)(F_1 + F_2 + F_4 + F_5) \times$$
$$\times (F_3 + F_7)(F_1 + F_2 + F_6).$$

Раскрыв скобки по правилам булевой алгебры, получим:

$$\Pi_V = F_1F_2F_7 + F_1F_3F_6 + F_1F_3F_7 + F_1F_5F_7 + F_3F_4F_6 +$$
$$+ F_2F_4F_6F_7 + F_4F_5F_6F_7.$$

Каждое из слагаемых содержит в неявном виде все вершины графа *G(X, U).* Поэтому для нахождения всевозможных минимальных раскрасок оставим в произведении Π_V только слагаемые, содержащие минимальное число сомножителей F_i (в нашем случае первые пять), и, раскрыв их, устраним всеми возможными способами дублирующие вершины. В общем случае каждое слагаемое дает 2^k вариантов раскраски, где k — число дублирующих вершин в сомножителях слагаемого. Например, пятое слагаемое дает четыре варианта хроматической раскраски графа при $k = 2$ (табл. 1).

Таблица 1

Номер варианта	Красный цвет	Синий цвет	Зеленый цвет
1	x_3, x_4, x_6, x_7	x_2, x_8	x_1, x_5, x_9
2	x_3, x_4, x_7	x_2, x_6, x_8	x_1, x_5, x_9
3	x_4, x_6, x_7	x_2, x_8	x_1, x_3, x_5, x_9
4	x_4, x_7	x_2, x_6, x_8	x_1, x_3, x_5, x_9

Знание числа внутренней устойчивости графа оказывается полезным при распознавании размещений компонентов на коммутационной плате, связанным с итеративными операциями перестановки компонентов (вершин графа), когда в качестве основного критерия оптимальности решения используют минимум суммарной длины связей между компонентами.

Наибольшего эффекта достигают при перестановке элементов непополнимого внутренне устойчивого подмножества F_q.

Число внешней устойчивости. Если в графе *G(X, U)* имеется некоторое подмножество вершин $R_l \subset X$, не смежных остальным вершинам графа, т. е.

$$\forall x_i \in (X \backslash R_l)\, \exists x_j \in R_l\, [(x_i, x_j) \in U], \qquad (4)$$

то такое подмножество называют внешне устойчивым.

Рассмотрим семейство $\mathscr{B} = \{R_1, R_2, \ldots, R_s\}$ всех внешне устойчивых подмножеств графа *G(X, U)*. Каждое подмножество $R_l \subset \mathscr{B}$ характеризуется мощностью $|R_l|$. Величину, равную мощности внешне устойчивого подмножества, содержащего наименьшее число элементов, называют числом внешней устойчивости $\beta(G)$ графа: $\beta(G) = \min\limits_{R_l \subset \mathscr{B}} |R_l|$.

По аналогии со свойствами внутренне устойчивого подмножества будем считать подмножество $R_l \subset \mathscr{B}$ неуменьшаемым, если из него нельзя выбросить ни одной вершины без потери свойства внешней устойчивости (4). Внешне устойчивое подмножество R_l называют наименьшим, если $|R_l| = \beta(G)$. Следовательно, наименьшее внешне устойчивое подмножество графа всегда неуменьшаемо.

В графе рис. 6 п. 4.2 можно выделить семейство $\mathscr{B} = \{R_1, R_2, \ldots, R_7\}$ неуменьшаемых внешне устойчивых подмножеств:

$$R_1 = \{x_3, x_4, x_5, x_7, x_9\};\ R_2 = \{x_3, x_4, x_6, x_7\};$$
$$R_3 = \{x_2, x_7, x_9\};\ R_4 = \{x_2, x_6, x_8\};$$
$$R_5 = \{x_2, x_6, x_7\};\ R_6 = \{x_1, x_3, x_5, x_9\};$$
$$R_7 = \{x_1, x_3, x_6, x_8\}.$$

Наименьшими внешне устойчивыми множествами являются R_3, R_4 и R_5, мощность которых равна трем. Следовательно, $\beta(G) = 3$.

Каждое неуменьшаемое внешне устойчивое подмножество для неориентированного графа *G(X, U)* является одновременно и непополнимым внутренне устойчивым (для ориентированных графов данное утверждение неверно). Таким образом, в данном случае для нахождения числа внешней устойчивости $\beta(G)$ можно использовать описанный ранее метод выделения всех непополнимых внутренне устойчивых подмножеств (метод Х. Магу).

На практике для простых графов оценить величину $\beta(G)$ можно следующим образом. Выбирают вершину $x_i \in X$ с максимальной локальной степенью $\rho(x_i)$ и включают ее в подмножество R_l. Процесс повторяют для подграфа с $X \setminus \Gamma x_i$ вершинами и т. д. до тех пор, пока в X остается хотя бы одна вершина, несмежная с R_l. Например, для графа рис. 6 п. 4.2 первой выбираем вершину x_2, у которой $\rho(x_4) = 4$. В подмножестве $X \setminus \Gamma x_2 = \{x_6, x_7, x_8, x_9\}$ максимальную локальную степень имеют вершины x_8 и x_9. Пусть второй вершиной определяемого внешне устойчивого подмножества R_l будет x_8. Тогда

единственной несмежной с x_2 и x_3 вершиной является x_6. В результате получим $R_1 = \{x_2, x_6, x_8\}$, следовательно, $\beta(G) = 3$.

Если в графе $G(X, U)$ некоторое подмножество вершин $N_l \subset X$ является одновременно внутренне и внешне устойчивым, то его называют ядром графа. Семейство $N = \{N_1, N_2, \ldots, N_t\}$ всех ядер графа определяют следующим образом:

$$N = \mathcal{F} \cup \mathcal{B}.$$

Необходимым условием наличия в графе $G(X, U)$ ядра N_l является

$$\alpha(G) = \max_{F_l \in \mathcal{F}} |F_l| \geqslant |N_l| \geqslant \min_{R_l \in \mathcal{B}} |R_l| = \beta(G).$$

В неориентированных графах все непополнимые (неуменьшаемые) внутренне (внешне) устойчивые подмножества являются ядрами.

Знание числа внешней устойчивости $\beta(G)$ графа $G(X, U)$ необходимо при проектировании многослойных печатных плат, когда требуется отыскать минимальное число слоев платы с минимальным числом переходов из слоя в слой (каждый такой переход должен обеспечивать максимально возможное покрытие коммутационного пространства).

4.5. Плоские графы и их свойства

При проектировании электрических соединений печатных плат к печатному монтажу предъявляются требования распознания полного отсутствия или минимального числа пересечений проводников при однослойном (многослойном) печатном монтаже. Если рисунок электрических соединений интерпретировать графом, вершины которого соответствуют контактным площадкам и переходам из слоя в слой, а ребра — печатным проводникам платы, то для выполнения указанных требований необходимо, чтобы граф (частичные графы, представляющие отдельные слои коммутации многослойных печатных плат) полностью размещался на плоскости, а его ребра пересекались только в вершинах при минимизации числа вершин, соответствующих контактным переходам, т. е. граф (частичные графы) должен быть плоским. В связи с этим задача распознавания планарности графа и построения его плоского изображения приобретает особое значение.

Основные понятия. Граф $G(X, U)$ называют плоским (планарным) тогда и только тогда, когда он имеет геометрическую реализацию в двухмерном евклидовом пространстве, т. е. может быть расположен на плоскости так, что все его ребра пересекаются только в вершинах X графа.

Из определения плоского графа следует, что планарность является внутренним свойством графа и не обязательно проявляется при произвольном его изображении. Геометрическая реализация графа в евклидовой плоскости, если она вообще возможна, имеет место лишь при определенном расположении его вершин и ребер на плоскости.

Рассмотрим общее условие существования плоского графа. Пусть граф *G(X, U)* — плоский (рис. 1, *а).*

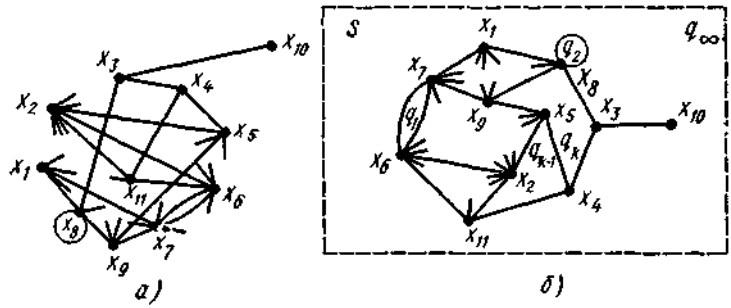


Рис. 1

Обозначим через $G_s(X,\ U)$ некоторую геометрическую реализацию *G(X, U)* в евклидовой плоскости *S* (рис. 1, *б).* Граф $G_s(X,\ U)$ разбивает плоскость *S* на отдельные компоненты связности — внешнюю и внутренние грани графа, образующие некоторое множество $Q = \{q_\infty, q_1, q_2, \ldots, q_k\}$ замкнутых областей плоскости *S*. При этом внутренние грани $q_1, q_2, \ldots, q_k$ ограничены минимальными (простыми) циклами графа, а внешняя (бесконечная) грань q_∞ представляет собой часть плоскости *S*, лежащую вне пределов максимального цикла $G_s(X,\ U)$.

Например, для графа рис. 1, *б* внутренние грани ограничены следующими минимальными циклами:

$$q_1 — (x_6,\ x_7),\ (x_7,\ x_6);\ q_2 — (x_8,\ x_8);\ \ldots;$$

$$q_{k-1} — (x_2,\ x_5),\ (x_5,\ x_4),\ (x_4,\ x_{11}),\ (x_{11},\ x_2);$$

$$q_k — (x_4,\ x_5),\ (x_5,\ x_9),\ (x_9,\ x_8),\ (x_8,\ x_3),\ (x_3,\ x_4).$$

Последовательность ребер $(x_1,\ x_8),\ (x_8,\ x_3),\ (x_3,\ x_4),\ (x_4,\ x_{11}),\ (x_{11},\ x_6),\ (x_6,\ x_7),\ (x_7,\ x_1)$ образует максимальный цикл.

Лемма. *Связный плоский граф с n вершинами, r ребрами и k гранями (включая внешнюю или бесконечную грань) удовлетворяет формуле Эйлера* $n — r + k = 2$.

Доказательство. Общее число минимальных циклов, содержащихся внутри максимального цикла, равно цикломатическому числу $\nu(G)$ графа (см. теорему в п. 4.4). Следовательно, число граней такого плоского графа равно цикломатическому числу $\nu(G)$ плюс единица (еще одна грань q_∞). Таким образом, $k = \nu(G) + 1 = (r - n + 1) + 1 = r - n + 2$.

Формула Эйлера позволяет показать, что в плоском графе $G(X, U)$ хотя бы одна вершина $x_i \in X$ имеет локальную степень $\rho(x_i) \leqslant 5$.

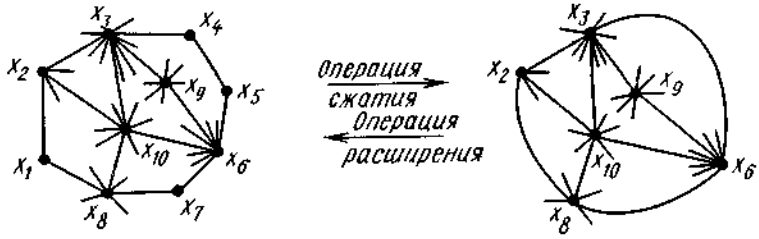


Рис. 2

Очевидно, что свойство планарности графа $G(X, U)$ не изменится, если на некоторых ребрах графа введем новые вершины степени два (рис. 2) или элементарные цепи, содержащие промежуточные вершины степени два, заменим ребрами, удалив промежуточные вершины из графа (обратная операция). Приведенные операции называют расширением и сжатием графа $G(X, U)$, а графы $G'(X', U')$, полученные в результате этих операций, называют изоморфными графу $G(X, U)$ с точностью до вершин степени два.

Заметим, что введение (удаление) петель и кратных ребер также не изменяет планарности графа.

Критерии планарности графов. При распознавании принципиальной электрической схемы радиоэлектронного устройства с точки зрения разработки рекомендаций по возможности ее реализации с помощью печатного монтажа ЛРО важно знать ответ на следующие вопросы: 1) является ли граф, соответствующий рассматриваемой принципиальной схеме, плоским? 2) если граф плоский, то как получить его изображение на плоскости без пересечения ребер?

Ответим на эти вопросы. Прежде всего определим максимально возможное число ребер плоского графа. Пусть задан граф $G(X, U)$, имеющий гамильтонов цикл. С помощью изоморфных преобразований перейдем к графу $G'(X, \hat{U})$, в котором ребра гамильтонова цикла не пересекаются. Тогда во внутренней и внешней областях выпуклой

фигуры, образованной ребрами гамильтонова цикла $G'(X, U)$, можно провести без пересечений не более $(n - 3)$ ребер (рис. 3).

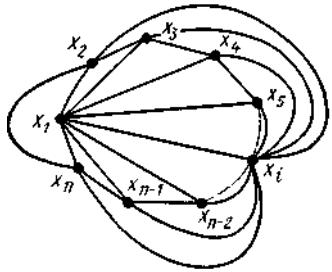


Рис. 3

Следовательно, максималь ное число некратных ребер у плоского графа

$$r_{\max} = n + (n - 3) + (n - 3) = 3(n - 2)$$

Кроме того, известно, что если число некратных ребер графа $r \leqslant n + 2$, то такой граф заведомо плоский. Таким образом, можно записать следующие условия для предварительного распознавания планарности графа:

$$r > 3(n - 2) \text{ — граф заведомо неплоский,}$$

$$r \leqslant n+2 \text{ — граф заведомо плоский.}$$

Рассмотрим критерии планарности графов с числом некратных ребер

$$n + 2 < r \leqslant 3(n - 2).$$

Теорема 1 (теорема Понтрягина — Куратовскогд).
Граф является плоским тогда и только тогда, когда он не содержит подграфа, изоморфного с точностью до вершин степени два одному из графов Понтрягина — Куратовского.

Графы Понтрягина — Куратовского первого (полный пятивершинный граф) и второго (полный граф Кенига с $|X_1| = |X_2| = 3$) типов приведены на рис. 4, *а, б.*

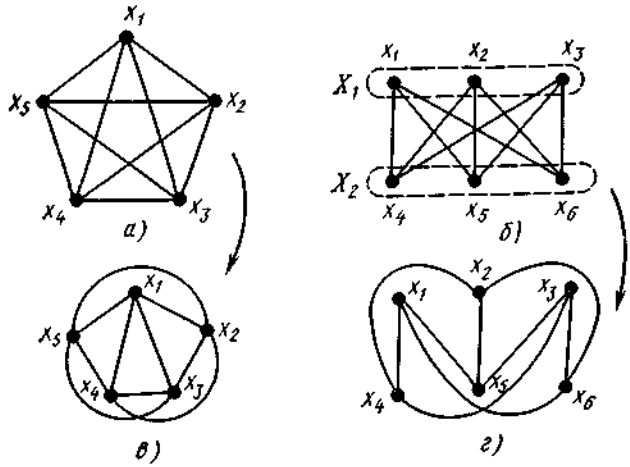


Рис. 4

Эти графы заведомо неплоские В имеют минимум одно пересечение ребер (рис. 4, *в, г).*

Критерий Понтрягина — Куратовского позволяет указать общие условия существования плоского графа. Однако практическая проверка этих условий для произвольного графа не всегда возможна.

Чтобы проверить наличие в произвольном графе *G(X, U)* подграфов Понтрягина — Куратовского, необходимо всевозможными способами удалять из *G (X, U)* ребра и вершины. Однако, как показал Кениг, такие подграфы могут входить в *G(X, U)* скрытно и их при этом невозможно опознать.

Мак-Лейн, связав граф с электрической цепью, в которой ребрам соответствовали конечные активные сопротивления цепи, доказал, что граф будет плоским тогда и только тогда, когда для него существует такая система контурных токов, в которой ни через какое ребро не протекает более двух токов. С помощью подобной системы можно получить плоское изображение графа. К сожалению, неясна возможность получения такой системы для графа, имеющего пересекающиеся ребра и, быть может, вообще неплоского.

С практической точки зрения более эффективным является использование критерия Уитни, который формулируется в виде следующей теоремы.

Теорема 2 (теорема Уитни). *Необходимым и достаточным условием планарности графа является наличие у него двойственного графа.*

Граф $G_d(Y, V)$ называют двойственным для плоского графа *G(X, U),*

если он получен из $G(X, U)$ в результате следующего построения. Каждой грани $q_i \in Q$ (рис. 5, *а)* графа *G(X, U)* ставим в соответствие вершину $y_i \in Y$.

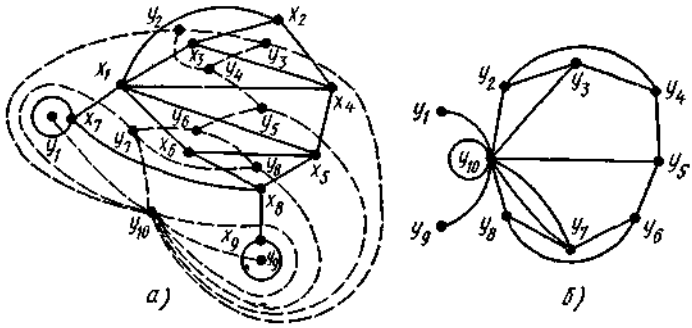


Рис. 5

Если две грани q_i и q_j смежны, то соединяем y_i и y_j ребром $v_f = (y_i, y_j)$, пересекающим один раз общую границу граней q_i, q_j и не проходящим через другие границы графа *G(X, U).* При этом, если некоторое ребро $u_k \in U$ не является границей *G(X, U),* т. е. находится внутри какой-либо грани, то в графе $G_d(Y, V)$ (рис. 5, *б)* ребру u_k соответствует петля.

Практическая эффективность использования критерия Уитни обусловлена существованием сравнительно простых методов, которые позволяют, преобразовав матрицу инцидентности графа *G(X, U)* в его цикломатическую матрицу **P**, судить о возможности реализации двойственного графа $G_d(Y, V)$, т. е. о планарности $G(X, U)$.

(Под цикломатической матрицей **P** понимают матрицу, в которой строки соответствуют всем независимым циклам, а столбцы — ребрам графа. Любой элемент p_{ij} матрицы **P** соответственно равен единице, если *j*-е ребро принадлежит *i*-му циклу и нулю — в противном случае. Подобную матрицу **P** называют матрицей контуров.)

Рассмотренные критерии планарности графов позволяют определить только потенциальную возможность плоской реализации конкретного графа в евклидовой плоскости, но не указывают способа фактического нахождения какого-либо из плоских его изображений.

Методы определения планарности графов и получения их плоских изображений. В основу большинства известных алгоритмов отыскания и построения плоского изображения графа положен метод, предложенный Бадером, суть которого в следующем.

Удалим из графа *G(X, U)* изолированные вершины, петли и кратные ребра, а также вершины степени два и ребра, являющиеся «перешейками», т. е. ребра, удаление каждого из которых приводит к увеличению числа компонент связности *G(X, U)* на единицу. Если *G(X, U)* — разделяющийся граф, т. е. граф, имеющий вершины $x_i \in X$, $i = 1, 2, \ldots, k$, где его можно разделить на отдельные компоненты связности $G_1(X_1, U_1), G_2(X_2, U_2), \ldots, G_m(X_m, U_m)$, каждая из которых $G_j(X_j, U_j)$ содержит вершину x_i^j, образованную из первоначальной вершины x_i, то осуществим разделение *G(X, U)* в вершинах $x_i \in X$.

Граф $G'(X', U')$, полученный в результате указанных операций над исходным графом (рис. 6, а), приведен на рис. 6, *б*.

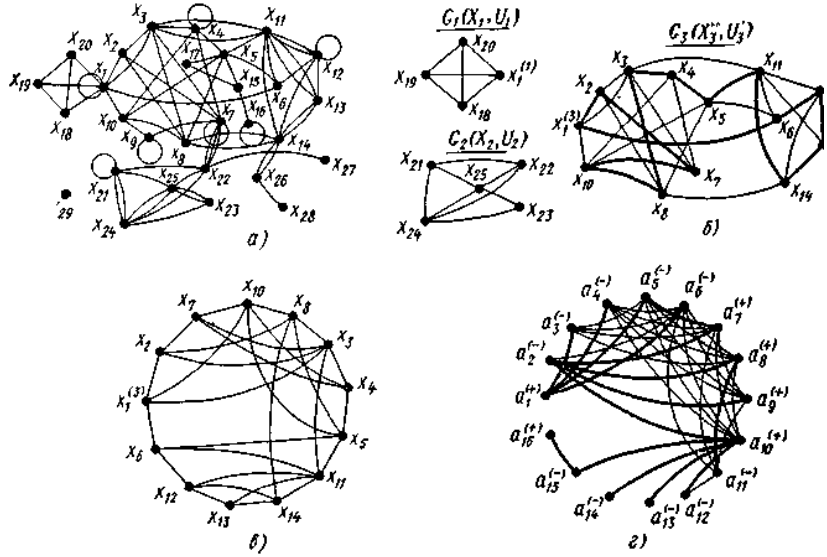


Рис. 6

Если, переместив в каждом из подграфов $G_j(X_j, U_j)$ вершины $x_i^{(1)}$ во внешний цикл фигуры $G_j(X_j, U_j)$, выполнить операцию объединения єтих компонент связности в единый граф *G(X, U)*, то последний будет и плоским тогда и только тогда, когда плоскими являются все компоненты связности $G_j(X_j, U_j)$. Таким образом, при определении планарности графа *G(X, U)* и построения его плоского изображения достаточно решить эту задачу для каждой компоненты связности $G_j(X_j, U_j)$.

Рассмотрим случай, когда в связном графе $G(X, U)$ известен гамильтонов цикл. К сожалению, неизвестен общий критерий наличия у графа этого цикла и способ его определения [в литературе приводятся лишь частные критерии наличия гамильтонова цикла у графа; например, если для любой пары вершин $x_i, x_j \in X$ графа $G(X, U)$ сумма локальных степеней $\rho(x_i) + \rho(x_j) \geqslant n$, то в графе существует гамильтонов цикл]. Но если для данного связного графа можно найти гамильтонов цикл, что на практике почти всегда имеет место, то определить, является ли $G(X, U)$ плоским графом, и, если да, получить его плоское изображение можно следующим образом.

Так как при геометрической реализации плоского графа гамильтонов цикл всегда можно получить в виде непересекающейся линии, то, выполнив соответствующие изоморфные преобразования, перейдем к графу с непересекающимися ребрами гамильтонова цикла. При этом плоскость разбивается на две части: внешнюю и внутреннюю. Все ребра графа, не вошедшие в гамильтонов цикл, при его плоском изображении целиком находятся либо внутри, либо вне этого цикла. Два таких ребра называют противопоставленными, если они пересекутся, будучи помещенными во внутреннюю или внешнюю часть. Ребра $u_k = (x_i, x_j)$, $u_l = (x_f, x_h) \in U$ тогда и только тогда будут непротивопоставленными, когда $x_f, x_h \in X$ находятся в одной из частей деления гамильтонова цикла вершинами $x_i, x_j \in X$, т. е. $x_f, x_h \in \{x_i, x_j\} \cup X_k \setminus x_f, x_h \in \{x_l, x_j\} \cup (X \setminus X_k)$, где X_k — множество вершин цикла, отсеченных ребром u_k.

Построим вспомогательный граф $G_v(A, B)$, множеству вершин которого $A = \{a_1, a_2, \ldots, a_s\}$ поставлено в однозначное соответствие множество ребер графа $G(X, U)$, не принадлежащих гамильтонову циклу, а множеству ребер $B = \{b_1, b_2, \ldots, b_t\}$ — пары вершин $\{a_k, a_l\}$, соответствующие противопоставленным ребрам $u_k, u_l \in U$ графа $G(X, U)$. На вершинах графа $G_v(A, B)$ строим произвольное дерево $H_T(A, B_T)$ и помечаем его вершины, начиная с произвольной, знаками плюс и минус, означающими, что данные ребра графа $G(X, U)$ находятся соответственно вне или внутри гамильтонова цикла. При этом никакие две смежные вершины дерева $H_T(A, B_T)$ не должны иметь одинаковые пометки.

Проверим непротиворечивость расстановки пометок для остальных ребер $b_j \in B \setminus B_T$ графа $G_v(A, B)$. Если эти ребра соединяют вершины с противоположными пометками, то исходный граф $G(X, U)$ — плоский; пометки дают информацию о том, в какой части плоскости необходимо располагать противопоставленные ребра для плоского изображения $G(X, U)$. В случае неплоского графа $G(X, U)$

анализ вершин графа $G_v(A, B)$ с противоречивыми пометками позволяет выявить те ребра, удаление которых из *G(X, U)* превращает оставшийся частичный граф в плоский.

Рассмотрим граф $G(X, U)$ рис. 6, *а.* Применив к нему операции разделения и упрощения, которые не изменяют свойств планарности исходного графа, получим несвязный граф $G'(X', U')$, состоящий из трех компонент связности: $G_1(X_1, U_1)$, $G_2(X_2, U_2)$ и $G_3(X_3, U_3)$, первые две из которых планарны (рис. 6, б). Следовательно, планарность графа *G(X, U)* полностью определяется свойствами подграфа $G_3(X_3, U_3)$, гамильтонов цикл в котором отмечен жирной линией (в рассматриваемом графе имеются и другие гамильтоновы циклы). Изображение данного подграфа без пересечений ребер гамильтонова цикла показано на рис. 6, *в.*

Построим вспомогательный граф $G_v = (A, B)$, в котором

$$A = \{a_1 = (x_1^{(3)}, x_{10}),\ a_2 = (x_1^{(3)}, x_3),\ a_3 = (x_2, x_8),$$
$$a_4 = (x_2, x_3),\ a_5 = (x_7, x_3),\ a_6 = (x_7, x_4),$$
$$a_7 = (x_{10}, x_4),\ a_8 = (x_{10}, x_5),\ a_9 = (x_8, x_5),$$
$$a_{10} = (x_8, x_{14}),\ a_{11} = (x_3, x_{11}),\ a_{12} = (x_5, x_6),$$
$$a_{13} = (x_{11}, x_6),\ a_{14} = (x_{11}, x_{12}),\ a_{15} = (x_{11}, x_{13}),$$
$$a_{16} = (x_{14}, x_{12})\}.$$

Произвольное дерево $H_T(A, B_T)$, построенное на вершинах этого графа (рис. 6, *г),* отмечено жирной линией. Пометим все вершины $H_T(A, B_T)$ знаками плюс и минус, начиная, например, с a_1.Так как ребра графа $G_v(A, B)$: (a_2, a_6), (a_3, a_6), (a_3, a_5), (a_4, a_6), (a_6, a_{11}), (a_7, a_{10}), (a_7, a_9), (a_8, a_{10}) и (a_{11}, a_{12}), не принадлежащие множеству B_T, соединяют вершины с одинаковыми пометками, подграф $G_3(X_3, U_3)$ является неплоским и может быть изображен с минимальным пересечением ребер (рис. 7, *а).*

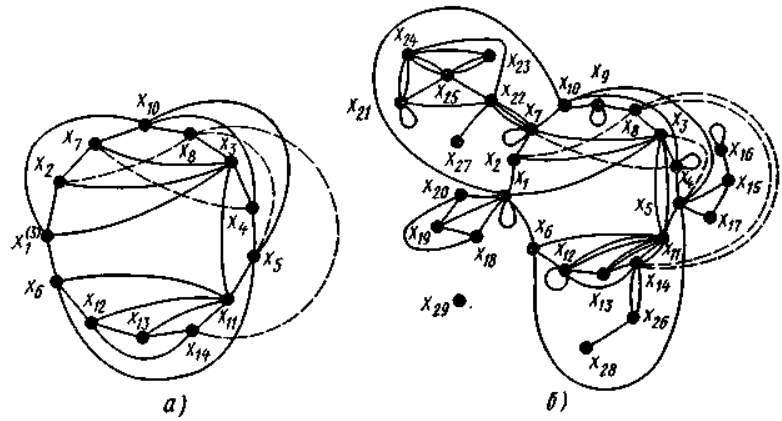


Рис. 7

Удаление из $G_b(A,\ B)$ вершин a_3, a_6, a_9 и a_{10} с инцидентными им ребрами позволяет непротиворечиво проставить пометки для всех оставшихся вершин вспомогательного графа. Следовательно, удаление из $G_3(X_3,\ U_3)$ ребер $(x_2,\ x_8)$, $(x_7,\ x_4)$, $(x_8,\ x_5)$ и $(x_8,\ x_{14})$ превратит оставшийся частичный подграф в пленарный.

На рис. 7, *а* данные ребра показаны пунктиром. Исходный граф *G(X, U)* с минимально возможным числом пересечений ребер показан на рис. 7, *б*.

Если в связном графе *G(X, U)* гамильтонов цикл неизвестен или не существует, то необходимо найти такой цикл, который включал бы в себя максимально возможное число вершин *G(X, U)*. Для этого воспользуемся методом последовательного удлинения цикла в графе, суть которого состоит в следующем. Найдем произвольный цикл L_1, проходящий только один раз через некоторое число вершин.

Проверим, имеются ли между какими-либо двумя соседними вершинами этого цикла еще по крайней мере одна цепь, проходящая только через вершины и ребра, не принадлежащие L_1. Если такая цепь ΔL_1 существует, то строим новый цикл L_2, в котором в отличие от L_1 непосредственное соединение между данными парами соседних вершин заменим на ΔL_1. Этот цикл содержит по меньшей мере на одну вершину и одно ребро больше, чем цикл L_1. Таким образом, начав с L_1, получим последовательность циклов $L_1,\ L_2,\ \ldots,\ L_f,\ \ldots,\ L_h = L_{\max}$. Удлинение L_f возможно, если при опускании L_f из исходного графа в нем остается хотя бы одна компонента связности, для которой вершины примыкания к L_f являются соседними.

Выполнив изоморфные преобразования, перейдем к графу с непересекающимися ребрами найденного максимального цикла L_{max}. После удаления этих ребер из *G(X, U)* получим отдельные компоненты связности $G_1(X_1, U_1)$, $G_2(X_2, U_2)$, ..., $G_m(X_m, U_m)$. По аналогии с предыдущим случаем будем называть два подграфа $G_i(X_i, U_i)$ и $G_j(X_j, U_j)$ противопоставленными, если они пересекутся, будучи оба помещенными во внутреннюю или внешнюю часть плоскости, ограниченной L_{max}. Очевидно, что $G_i(X_i, U_i)$ и $G_j(X_j, U_j)$ тогда и только тогда не будут противопоставленными, когда на одном из отрезков L_{max}, образованных вершинами примыкания первого подграфа, находятся все вершины примыкания второго, и наоборот.

Построим вспомогательный граф $G_v(A, B)$, множеству вершин которого $A = \{a_1, a_2, ..., a_m\}$ поставлено в однозначное соответствие множество подграфов $G_1(X_1, U_1)$, $G_2(X_2, U_2)$, ..., $G_m(X_m, U_m)$, а множеству ребер $B = \{b_1, b_2, ..., b_t\}$ — пары вершин $\{a_k, a_l\}$, соответствующие противопоставленным подграфам $G_k(X_k, U_k)$ и $G_l(X_l, U_l)$.

С помощью метода пометок проверим, можно ли противопоставленные подграфы расположить в различных частях плоскости, ограниченных циклом L_{max}. Если это возможно, то необходимо определить, является ли плоским каждый отдельно взятый подграф $G_i(X_i, U_i)$ вместе с циклом L_{max}. Решение этой задачи осуществляют в зависимости от того, известен или нет гамильтонов цикл. При этом $G_i(X_i, U_i)$ вместе с L_{max} рассматриваем как исходный граф. Этот граф имеет меньшее число вершин и ребер, чем *G(X, U),* и процесс отыскания плоского изображения закончится через конечное число шагов.

Рассмотрим граф *G(X, U)* на рис. 8, *а.* Применив к данному графу операции разделения и упрощения, получим несвязный подграф $G'(X', U')$, состоящий из двух компонент связности: $G_1(X_1, U_1)$ и $G_2(X_2, U_2)$, последняя из которых планарна (рис 8, *б).* Следовательно, планарность *G(X, U)* полностью определяется свойствами подграфа $G_1(X_1, U_1)$. Процесс последовательного расширения цикла в $G_1(X_1, U_1)$ показан на рис. 8, *б* и *в.* Цикл, найденный на каждом этапе, отмечен жирной линией.

(Рассматриваемый граф имеет гамильтонов цикл, получаемый при следующей последовательности обхода вершин: $x_1, x_2, x_3, x_{13}, x_4, x_5, x_7, x_{18}, x_{23}, x_{12}, x_{11}, x_{15}, x_{16}, x_{17}, x_{22}, x_{10}, x_9, x_6, x_1$. Однако первоначально выбранная последовательность обхода вершин

подграфа (рис. 8, *б)* исключила возможность его нахождения с помощью последовательного расширения найденного цикла.)

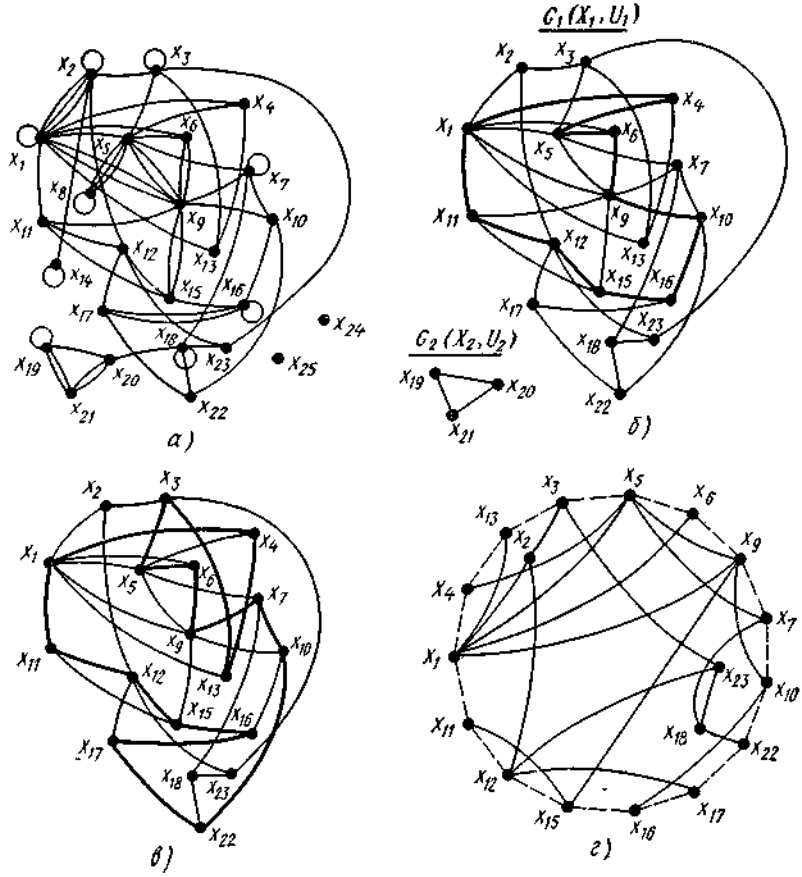


Рис. 8

Изображение данного подграфа без пересечения ребер найденного максимального цикла L_{max} приведено на рис. 8, *г.* Построим вспомогательный граф $G_v(A, B)$, в котором

$$A = \{a_1 = \{(x_3,\ x_{23}),\ (x_7,\ x_{18}),\ (x_{12},\ x_{23}),\ (x_{18},\ x_{23}),$$
$$(x_{18},\ x_{22})\},\ a_2 = \{(x_1,\ x_2),\ (x_2\ x_3),\ (x_2,\ x_{12})\},$$
$$a_3 = (x_1,\ x_{13}),\ a_4 = (x_4,\ x_5),\ a_5 = (x_1,\ x_5),\ a_6 = (x_1,\ x_6),$$
$$a_7 = (x_1,\ x_9),\ a_8 = (x_5,\ x_9),\ a_9 = (x_9,\ x_{15}),\ a_{10} = (x_9,\ x_{10}),\ a_{11} = (x_{10},\ x_{16}),$$
$$a_{12} = (x_{12},\ x_{17}),\ a_{13} = (x_{14},\ x_{15}),\ a_{14} = (x_5,\ x_7)\}.$$

Произвольное дерево $H_T(A,\ B_T)$, построенное на вершинах этого графа (рис. 9, а), отмечено жирной линией.

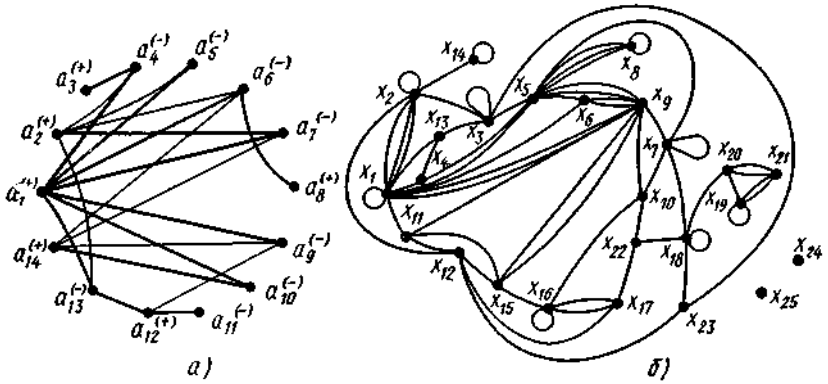


Рис. 9

Пометим все вершины $H_T(A,\ B_T)$ знаками плюс и минус, начиная, например,с a_1. При этом все ребра из подмножества $B \setminus B_T$ соединяют вершины с противоположными пометками. Так как каждая компонента связности вместе с циклом L_{max} планарна, то исходный граф — $G(X,\ U)$ плоский и может быть изображен без пересечения ребер (рис. 9, б).

При машинной реализации метода Бадера графы обычно задают с помощью матриц. Наиболее удобной формой представления информации является использование матриц смежности. Тогда последовательность определения планарности графа будет следующей.

После нахождения максимального (гамильтонова) цикла в графе *G(X, U)* исходная матрица смежности $\mathbf{A}(G) = ||a_{ij}||_{n\times n}$ преобразуется в $\mathbf{A}'(G) = ||a'_{ij}||_{n\times n}$, в которой строки и столбцы, соответствующие вошедшим в цикл вершинам $x_i,\ x_j \in X$ располагаются в той же последовательности, что и в цикле. Так как рассматриваем неориентированные графы, то матрица $\mathbf{A}'(G)$ симметрична относительно главной диагонали. Следовательно, без потери информации можно

использовать треугольную матрицу $\mathbf{A}^*(\mathbf{G}) = ||a^*_{ij}||_{n\times n}$, элементы которой

$$a^*_{ij} = \begin{cases} *, \text{ если } (j = i+1) \vee (i = 1 \& \quad j = m), \\ 0, \text{ если } i \geqslant j; \\ a'_{ij}, \text{ если } i < j-1, \end{cases} \tag{1}$$

где m —число вершин, вошедших в максимальный цикл; $m \leqslant n$ (знак равенства соответствует гамильтонову циклу). Выражение (1) показывает, что ребра, составляющие цикл, помечают знаком «*»; такая запись позволяет упростить ручное оперирование с матрицами (для машинной реализации метода такую пометку не делают).

Разобьем все множество компонент связности, получаемое после удаления из $G(X, U)$ ребер максимального цикла, на два подмножества. Первое подмножество состоит из компонент связности, содержащих только вершины максимального цикла (ребра, не вошедшие в цикл); второе — из всех оставшихся компонент связности.

Две компоненты связности $G_p(X_p, U_p)$ и $G_q(X_q, U_q)$ из первого подмножества будут противопоставленными тогда и только тогда (рис. 10, а), когда

$$[(i+1 \leqslant g \leqslant j-1) \& (j+1 \leqslant f \leqslant m)] \vee [1 \leqslant g \leqslant i-1) \& (i + 1 \leqslant f \leqslant j-1)];$$
$$u_{ij} \in U_p,\ u_{gf} \in U_q, \tag{2}$$

т. е. ребро $u_{gf} = (x_g, x_f)$ противопоставлено $u_{ij} = (x_i, x_j)$, если элемент a^*_{qf} находится в подматрице $\mathbf{A}_{ij}$ или $\mathbf{A}'_{ij}$ (рис. 10, *б*).

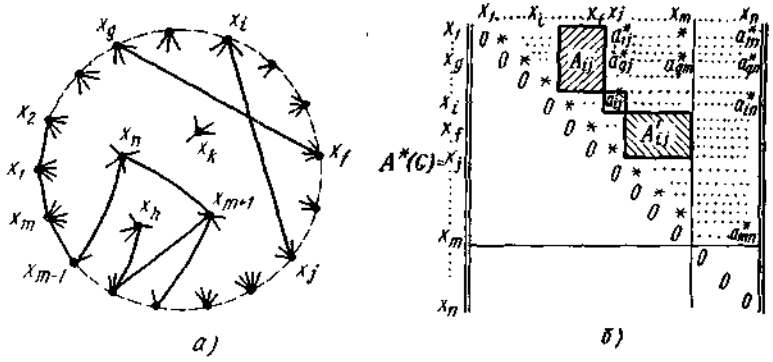


Рис. 10

Две компоненты связности, одна из которых $G_p(X_p, U_p)$ принадлежит первому подмножеству, а другая $G_q(X_q, U_q)$ — второму, будут противопоставленными (рис. 11, *а*), если

$$\sum_{g=t+1}^{j-1} \sum_{f \in F} a^*_{gf} \left(\sum_{g=1}^{t+1} \sum_{f \in F} a^*_{gf} + \sum_{g=j+1}^{m} \sum_{f \in F} a^*_{gf} \right) > 0; \qquad (3)$$

$$u_{ij} \in U_p,$$

где F — множество индексов вершин $G_q(X_q, U_q)$, не принадлежащих максимальному циклу ($f>m$).

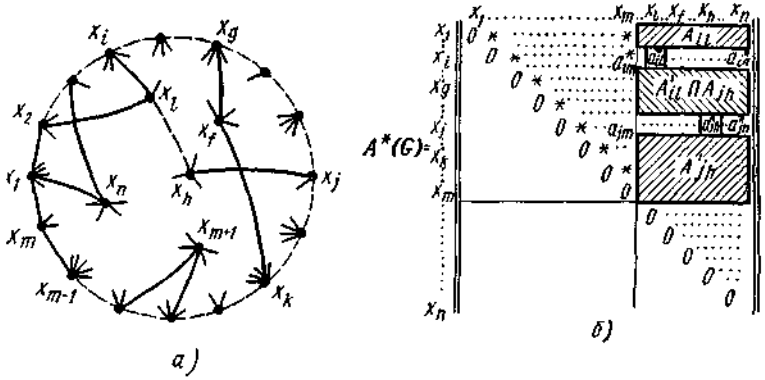


Рис. 11

Иначе $G_q(X_q, U_q)$ противопоставлено ребру u_{ij}, если элементы a^*_{gf}, соответствующие $u_{gf} \in u_q$, находятся одновременно как в подматрице $\mathbf{A}'_{ij}$, так и в подматрице $\mathbf{A}_{ij}''$ или $\mathbf{A}_{ij}'''$ (рис. 12, *б*).

Две компоненты связности $G_p(X_p, U_p)$ и $G_q(X_q, U_q)$ из второго подмножества противопоставлены друг другу (рис. 13, *а*), если

$$\exists\, u_{il},\ u_{jh} \in U_p \left[\sum_{g=i+1}^{j-1} \sum_{f \in F} a^*_{gf} \left(\sum_{g=1}^{i+1} \sum_{f \in F} a^*_{gf} + \right. \right.$$

$$\left. \left. + \sum_{g=j+1}^{m} \sum_{f \in F} a^*_{gf} \right) > 0 \right], \qquad (4)$$

где $l,\ h \in H$; H и F — множества индексов вершин соответственно $G_p(X_p, U_p)$ и $G_q(X_q, U_q)$, не принадлежащих максимальному циклу ($h > m, f > m$). Иначе $G_p(X_p, U_p)$ противопоставлено $G_q(X_q, U_q)$, если для какой-либо пары $a^*_{ih} > 0$ и $a^*_{jh} > 0$ элементы a^*_{gf},

отличные от нуля, находятся одновременно как в подматрице $\mathbf{A}'_{ih} \cap \mathbf{A}_{jh}$, так и в подматрице $\mathbf{A}_{ih}$ или $\mathbf{A}'_{jh}$ (рис. 12, *б*).

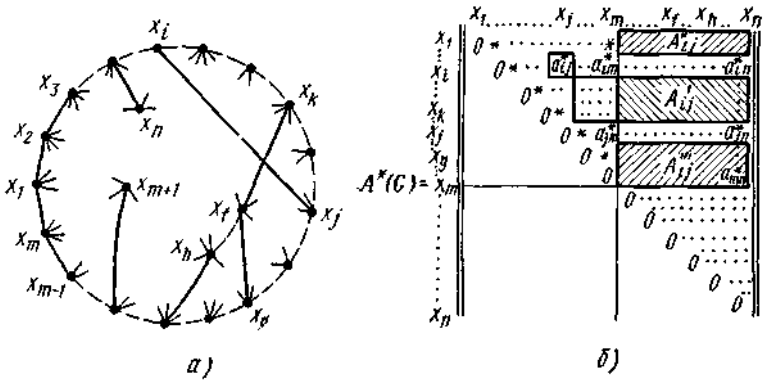


Рис. 12

Соотношения (2) — (4) позволяют по $\mathbf{A}^*(G)$ построить матрицу смежности $\mathbf{B}(G_v)$ вспомогательного графа $G_v(A, B)$, в которой строки и столбцы соответствуют отдельным компонентам связности, а элементы b_{ij} матрицы $\mathbf{B}(G_v)$ равны единице, если компоненты $G_i(X_i, U_i)$ м$G_j(X_j, U_j)$ противопоставлены, и нулю в противном случае.

По матрице $\mathbf{B}(G_v)$ определяют минимальную раскраску вершин нспомогательного графа. Если $k(G_v) = 2$, то исходный граф *G(X, U)* — планарен, и для его плоского изображения достаточно компоненты связности $G_i(X_i, U_i)$ и $G_j(X_j, U_j)$, соответствующие внутренне устойчивым подмножествам $G_v(A, B)$ с различной раскраской, расположить в различных областях деления плоскости максимальным циклом. Если $k(G_v) > 2$, то граф *G(X, U)* — непланарен и для нахождения минимального числа ребер, удаление которых превращает исходный граф в плоский, необходимо у всех внутренне устойчивых подмножеств, обеспечивающих минимальную раскраску $G_v(A, B)$, определить число входящих в них ребер *G(X, U)* и выделить из этих множеств два таких, которые содержат максимальное число ребер. Остальные ребра из графа *G(X, U)* следует удалить. Для компонент связности, содержащих вершины, не входящие в максимальный цикл (из второго подмножества), находят минимальное число ребер, при удалении которых выражения (3) и (4) становятся тождественно равными нулю.

Разбиение графа на плоские су графы. Если граф *G(X,U)* неплоский, то для его геометрической реализации удаляют из *G(X, U)* отдельные ребра $u_{ij} \in U$ (переносят на другую плоскость). Минимальное число ребер, которое необходимо удалить из графа для получения его плоского изображения, называют числом планарности графа $\Theta(G)$. При вынесении этих ребер на вторую плоскость получают частичный граф (суграф), состоящий из $\Theta(G)$ ребер, который также может оказаться неплоским. Тогда вновь решают задачу вынесения отдельных ребер на следующую плоскость и т. д. Минимальное число плоскостей *m,* при котором граф *G(X, U)* разбивается на плоские суграфы $G_1(X, U_1), G_2(X, U_2), \ldots, G_m(X, U_m)$, называют толщиной графа *t(G).* Если переходы между плоскостями совершать не по одноименным вершинам, а в любой точке ребра путем его подразбиения (рис. 13), то для любого графа *G(X, U)* можно построить гомеоморфный ему граф $G'(X', U')$ толщиной $t(G') \leqslant 2$.

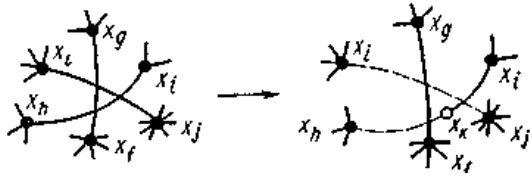


Рис. 13

На рис. 13 ребра, вынесенные на вторую плоскость, изображены пунктиром.

Наибольший интерес представляет разбиение *G(X, U)* на плоские суграфы, когда переходы между плоскостями осуществляют только по вершинам графа (по ножкам конструктивных элементов, установленных на плате). При использовании в аппаратуре двусторонних печатных плат важное значение приобретает задача поиска критерия бипланарности произвольного графа, т. е. возможности разбиения его на два плоских суграфа.

На практике при разработке конструкций на печатных платах возможны два случая.

1. Местоположение конструктивных элементов на плате заранее неизвестно, и его требуется определить, минимизировав либо число перемычек, выполняемых в местах пересечения печатных проводников при однослойном монтаже, либо число слоев коммутации при использовании многослойных печатных плат. Представив схему электрических соединений конструктивных элементов в виде неориентированного графа *G(X, U),* переходят к поиску рационального распо-

ложения вершин и ребер *G(X, U)* на плоскости, обеспечивающему минимум пересечений ребер графа. При этом можно воспользоваться методом Бадера. Найденное взаимное расположение вершин графа будет соответствовать рациональному размещению элементов на плате, а число удаляемых ребер — необходимому числу перемычек или электрических цепей, переносимых из первого слоя во второй.

2. Местоположение конструктивных элементов получено на предыдущем этапе размещения. Требуется найти рациональное расположение ребер графа *G(X, U),* интерпретирующего схему электрических соединений, при котором суммарная длина и общее число пересечений ребер *G(X, U)* минимальны.

5. Математическая логика

Одной из основных задач математической логики является анализ оснований математики. Но она уже вышла из рамок этой задачи и оказала существенное влияние на развитие самой математики. Из ее идей возникло точное определение понятия алгоритма, что позволило решать многие вопросы, которые без этого остались бы в принципе неразрешенными. Возникший в математической логике аппарат нашел приложение и в вопросах теории распознавания.

В начале этого раздела излагаются основные положения, относящиеся к логическим функциям. Подробно исследуются булевы функции двух переменных, зависимости между ними и методы построения функционально полных систем. Наряду с булевой алгеброй, рассматривается алгебра Жегалкина, что позволяет глубже проникнуть в структуру логических функций.

Аппарат математической логики в значительной степени сложился под влиянием прикладных проблем, в рамках которых развились его специфические особенности.

Триумфом сотрудничества математики и техники явилось создание вычислительных машин с программным управлением. К тому времени, когда электроника, магнитная техника и электромеханика смогли предложить эффективные методы построения логических элементов и устройств преобразования информации, математическая логика уже располагала в общих чертах аппаратом для проектирования схем, реализующих сложные логические функции.

Дальнейшие обобщения привели к развитию теории автоматов, основной задачей которой является математическое моделирование физических или абстрактных процессов, технических устройств и некоторых сторон поведения живых организмов. Автоматы ис-

пользуются в качестве универсальной модели в самых разнообразных областях, в том числе и в теории распознавания.

При рассмотрении конечных автоматов, контактных и логических схем используются различные способы представления логических функций: многомерные кубы, карты Карно, символика *s*-кубов. На основе таких представлений излагаются основные методы минимизации булевых функций.

Наряду с двоичными функциональными элементами, разработаны и находят практическое применение многозначные элементы, характеризующиеся рядом положительных особенностей. В связи с этим сильно возросло значение многозначной логики, изложению основных положений которой посвящен специальный параграф. Там же кратко представлены другие логики, развившиеся в связи с техническими и биологическими проблемами: пороговая, мажоритарная, нейронная, потенциально-импульсная и фазоимпульспая.

5.1. Логические функции

1. Логические функции как отображения. Отличительная особенность логических функций состоит в том, что они принимают значения в конечных множествах. Иначе говоря, область значений логической функции всегда представляет собой конечную совокупность чисел, символов, понятий, свойств и, вообще, любых объектов. Если область значений функции содержит k различных элементов, то она называется *k-значной функцией.*

Чтобы различать элементы области значений функции, их необходимо как-то отметить. Удобнее всего элементы перенумеровать числами от 1 до k или обозначить какими-нибудь символами (например, буквами). Перечень всех символов, соответствующих области значений, называют *алфавитом,* а сами символы — *буквами* этого алфавита (буквами могут служить как собственно буквы латинского, русского или другого алфавита, так и порядковые числа или любые другие символы).

Логические функции могут зависеть от одной, двух и, вообще, любого числа переменных (аргументов) $x_1, x_2, \ldots, x_n$. В отличие от самой функции, аргументы могут принимать значения из элементов как конечных, так и бесконечных множеств.

В теоретико-множественном смысле логическая функция n переменных $y = f(x_1, x_2, \ldots, x_n)$ представляет собой отображение множества *наборов* (n-мерных векторов, кортежей, последовательностей)

вида $(x_1, x_2, \ldots, x_n)$, являющегося областью ее определения, на множество ее значений $N = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$. Логическую функцию можно также рассматривать как операцию, заданную законом композиции $X_1 \times X_2 \times \ldots \times X_n \to N$, где $X_1, X_2, \ldots, X_n$ — множества, на которых определены аргументы $x_1 \in X_1, x_2 \in X_2, \ldots, x_n \in X_n$.

2. Однородные функции. Если аргументы принимают значения из того же множества, что и сама функция, то ее называют *однородной функцией.* В этом случае $X_1 = X_2 = \ldots = X_n = N$ и однородная функция, рассматриваемая как закон композиции $N^n \to N$, определяет некоторую *n-местную операцию* на конечном множестве *N.*

Областью определения однородной функции $y = f(x_1, x_2, \ldots, x_n)$ служит множество наборов $(x_1, x_2, \ldots, x_n)$, называемых *словами,* где каждый из аргументов $x_1, x_2, \ldots, x_n$ замещается буквами *k*-ичного алфавита $\{0, 1, \ldots, k-1\}$. Количество *n* букв в данном слове определяет его *длину.*

Очевидно, число всевозможных слов длины *n* в *k*-ичном алфавите равно k^n. Так как каждому такому слову имеется возможность предписать одно из *k* значений множества *N,* то общее количество однородных функций от *n* переменных выражается числом $k^{(kn)}$.

Если буквами алфавита служат числа от 0 до $k-1$, то каждое слово $(x_1, x_2, \ldots, x_n)$ символически представляется упорядоченной последовательностью *n* таких чисел и рассматривается как запись *n*-разрядного числа в позиционной системе счисления с основанием *k,* т. е. $x_1 k^{n-1} + x_2 k^{n-2} + \ldots + x_{n-1} k^1 + x_n k^0 = q$. Числа $q = 0, 1, \ldots, k^n - 1$ служат *номерами слов* и тем самым на множестве всех слов вводится естественная упорядоченность (отношение строгого порядка). Аналогично *номерами функций* можно считать k^n-разрядные числа в той же системе счисления.

Различные слова длины *n* в данном алфавите образуются как *n*-перестановки с повторениями (2. 10. 1). Так, в трехзначном алфавите $\{0, 1, 2\}$ словами длины 4 будут все четырехразрядные числа с основанием $k = 3$, т. е. 0000, 0001, 0002, 0010, 0011, ..., 2221, 2222, которые соответствуют десятичным числам от 0 до $80 = 2 \cdot 3^3 + 2 \cdot 3^2 + 2 \cdot 3^1 + 2 \cdot 3^0$. Поставив каждому такому четырехразрядному числу в соответствие одну из букв алфавита $\{0, 1, 2\}$, получим некоторую функцию четырех переменных $f_i(x_1, x_2, x_3, x_4)$, причем количество таких функций выражается огромным числом 3^{81}.

Пусть алфавит состоит из трех букв русского алфавита {о, п, т}. Множество пятибуквенных слов в этом алфавите состоит из $3^5 = 243$ элементов. Наряду с такими имеющими прямой смысл словами, как

«топот» и «потоп», оно также включает все другие 5-перестановки, например: «ооппт», «поппп», «тттоп» и др.

Примерами однородных логических функций двух переменных могут служить операции сложения и умножения одноразрядных m-значных чисел по модулю m (2. 8. 7), внутренние операции поля Галуа (2. 8. 9) с четырехзначным алфавитом $\{0, 1, A, B\}$ и т. п.

3. Табличное задание функций. Как и бинарный закон композиции (2. 7. 2), однородная функция двух переменных может быть задана таблицей соответствия (матрицей), строки и столбцы которой соответствуют буквам алфавита. Таким способом представлялись функции одной и двух переменных в (1. 5. 2),(1. 5. 8) и (1. 5. 10). Для представления функций трех и большего числа переменных потребовались бы трехмерные и, вообще, n-мерные таблицы. Этого можно избежать, если столбцы матрицы поставить в соответствие не буквам алфавита, а словам, т. е. образовать k^n столбцов. Для каждой функции отводится строка, клетки которой заполняются буквами из данного алфавита. Матрица всех функций n переменных в k -значном алфавите содержит $k^{(k^n)}$ строк и называется *общей таблицей соответствия.* Например, для k — 3 и n — 2 такая матрица имеет вид:

x_1	0	0	0	1	1	1	2	2	2
x_2	0	1	2	0	1	2	0	1	2
y_0	0	0	0	0	0	0	0	0	0
y_1	0	0	0	0	0	0	0	0	1
y_2	0	0	0	0	0	0	0	0	2
...	.	.	.	.	.	.	.	.	.
y_{2361}	0	1	0	0	1	2	2	0	1
...	.	.	.	.	.	.	.	.	.
y_{19682}	2	2	2	2	2	2	2	2	2

Номера столбцов определяются расположенными над ними n-разрядными числами с основанием k, каждое из которых читается сверху вниз. Номера функций отождествляются с k^n-разрядными числами, которые соответствуют строкам матрицы в той же системе счисления.

4. Двузначные однородные функции. Наиболее простым и в то же время важнейшим классом однородных функций являются *двузначные (булевы) функции.*

Областью определения булевых функций от n переменных служит множество слов длины n. Они представляют собой всевозможные наборы из n двоичных цифр и их общее колпчесшо равно 2^n.

Число всевозможных булевых функций n переменных $v = 2^n$ быстро возрастает с увеличением n (при n = 3 оно равно 256, а при n = 5 превышает 4 миллиарда). Но функции одной и двух переменных еще

можно перечислить и подробно исследовать, так как их количество сравнительно невелико ($v = 4$ при $n = 1$ и $v = 16$ при $n = 2$).

5. Булевы функции одной переменной. Общая таблица соответствия для булевых функций одной переменной имеет вид (справа указаны обозначения функций):

x	0	1	y
y_0	0	0	0
y_1	0	1	x
y_2	1	0	$\bar{x}$
y_3	1	1	1

Две функции $y_0 = 0$ и $y_3 = 1$ представляют собой *функции-константы* (тождественный нуль и тождественная единица), так как они не изменяют своих значений при изменении аргумента. Функция $y_1 = x$ повторяет значения переменной x и потому просто совпадает с ней. Единственной нетривиальной функцией является $y_2 = \bar{x}$, называемая *отрицанием,* или *инверсией* ($\bar{x}$ читается «не x»). Она равна 1, когда аргумент принимает значение 0, и равна 0 при аргументе 1.

6. Булевы функции двух переменных. Все 16 функций двух переменных приведены в табл. 1, где указаны условные обозначения, названия и чтения функций (в скобках даны встречающиеся в литературе варианты).

Шесть из приведенных функций не зависят от x_1 или x_2 (или от обоих вместе). Это две константы ($y_0 = 0$ и $y_{15} = 1$), повторения ($y_3 = x_1$ и $y_5 = x_2$) и отрицания ($y_{10} = \bar{x}_2$, $y_{12} = \bar{x}_1$), являющиеся функциями одной переменной (x_1 или x_2). Из остальных десяти функций две (y_4 и y_{11}) отличаются от соответствующих им (y_2 и y_{13}) лишь порядком расположения аргументов и поэтому не являются самостоятельными. Поэтому из 16 булевых функций двух переменных только восемь являются оригинальными (y_1, y_2, y_6, y_7, y_8, y_9, y_{13}, y_{14}).

Рассмотрение булевых функций одной, двух и большего числа переменных показывает, что всякая функция от меньшего числа переменных содержится среди функций большего числа переменных. Функции, которые сводятся к зависимости от меньшего числа переменных, называют *вырожденными,* а функции, существенно зависящие от всех переменных, являются *невырожденными.* Так, среди функций одной переменной имеются две вырожденные (константы 0 и 1, которые можно рассматривать как функции от нуля переменных), функции двух переменных содержат те же константы и четыре функции одной переменной и т. д.

Таблица 1

Булевы функции двух переменных

x_1 / x_2	0 0 1 1 / 0 1 0 1	Обозначения	Названия	Чтение
y_0	0 0 0 0	0	Константа 0 (тождественный нуль, всегда ложно)	*Любое 0*
y_1	0 0 0 1	x_1x_2; $x_1 \wedge x_2$ (x_1 & x_2; $x_1 \bigcap x_2$)	Конъюнкция (совпадение, произведение, пересечение, логическое «и»)	x_1 и x_2 (и x_1 и x_2)
y_2	0 0 1 0	$x_1 \leftarrow x_2$ ($x_1 \supset x_2$; $x_1 \setminus x_2$)	Отрицание импликации (совпадение с запретом, антисовпадение, запрет)	x_1, но не x_2
y_3	0 0 1 1	x_1	Повторение (утверждение, доминация) первого аргумента	Как x_1
y_4	0 1 0 0	$x_2 \leftarrow x_1$ ($x_1 \subset x_2$; $x_2 \setminus x_1$)	Отрицание обратной импликации (обратное антисовпадение)	Не x_1, но x_2
y_5	0 1 0 1	x_2	Повторение (утверждение, доминация) второго аргумента	Как x_2
y_6	0 1 1 0	$x_1 + x_2$ ($x_1 \nabla x_2$; $x_1 \oplus x_2$)	Сумма по модулю 2 (неравнозначность, антиэквивалентность)	x_1 не как x_2 (или x_1 или x_2)
y_7	0 1 1 1	$x_1 \vee x_2$ ($x_1 + x_2$; $x_1 \bigcup x_2$)	Дизъюнкция (разделение, логическая сумма, сборка, логическое «или»)	x_1 или x_2 (x_1 или хотя бы x_2)
y_8	1 0 0 0	$x_1 \downarrow x_2$ ($x_1 \overline{\nabla} x_2$; $x_1 \bigcirc x_2$)	Стрелка Пирса (функция Вебба, отрицание дизъюнкции, логическое «не — или»)	Ни x_1, ни x_2

Продолжение табл. 1

x_1 / x_2	0 0 1 1 / 0 1 0 1	Обозначения	Названия	Чтение
y_9	1 0 0 1	$x_1 \sim x_2$ ($x_1 \equiv x_2$; $x_1 \leftrightarrow x_2$)	Эквиваленция (равнозначность, эквивалентность, взаимозависимость)	x_1 как x_2 (x_1, если и только если x_2)
y_{10}	1 0 1 0	$\bar{x}_2$ (x_2; $\sim x_2$; $\neg x_2$)	Отрицание (инверсия) второго аргумента (дополнение к первой переменной)	Не x_2
y_{11}	1 0 1 1	$x_2 \rightarrow x_1$ ($x_1 \subset x_2$; $x_1 < x_2$)	Обратная импликация (обратное разделение с запретом, обратная селекция)	Если x_2, то x_1 (x_1 или не x_2)
y_{12}	1 1 0 0	$\bar{x}_1$ (x_1; $\sim x_1$; $\neg x_1$)	Отрицание (инверсия) первого аргумента (дополнение ко второй переменной)	Не x_1
y_{13}	1 1 0 1	$x_1 \rightarrow x_2$ ($x_1 \supset x_2$; $x_1 > x_2$)	Импликация (разделение с запретом, следование, селекция)	Если x_1, то x_2 (не x_1 или x_2)
y_{14}	1 1 1 0	x_1/x_2 ($x_1 \overline{\wedge} x_2$; $x_1 \overline{\&} x_2$)	Штрих Шеффера (отрицание конъюнкции, несовместность, логическое «не—и»)	Не x_1 или не x_2
y_{15}	1 1 1 1	1	Константа 1 (тождественная единица, всегда истинно)	Любое 1

7. Зависимость между булевыми функциями. Из табл. 1 видно, что между функциями имеются зависимости $y_i = \bar{y}_{15-i}$ ($i = 0, 1, \ldots, 15$), на основании которых можно записать соотношения для констант $0 = \bar{1}$ и $1 = \bar{0}$, для функции одной переменной $x = \bar{\bar{x}}$ и для функций двух переменных:

$$x_1 x_2 = \overline{x_1/x_2};\ x_1 \leftarrow x_2 = \overline{x_1 \rightarrow x_2};\ x_1 + x_2 = \overline{x_1 \sim x_2};\ x_1 \vee x_2 = \overline{x_1 \downarrow x_2},$$

или

$$x_1/x_2 = \overline{x_1 x_2};\ x_1 \rightarrow x_2 = \overline{x_1 \leftarrow x_2};\ x_1 \sim x_2 = \overline{x_1 + x_2};\ x_1 \downarrow x_2 = \overline{x_1 \vee x_2}.$$

Из этих зависимостей следует, что любая функция двух переменных (включая константы) выражается в аналитической форме через совокупность шести функций, содержащей отрицание $\bar{x}$ и любую из каждой пары функций $\{y_0, y_{15}\}$, $\{y_1, y_{14}\}$, $\{y_2, y_{13}\}$, $\{y_6, y_9\}$, $\{y_7, y_8\}$. Например, такой совокупностью могут служить функции: константа 0, отрицание $\bar{x}$, конъюнкция $x_1 x_2$, дизъюнкция $x_1 \vee x_2$,

эквиваленция $x_1 \sim x_2$ и импликация $x_1 \to x_2$. Они используются в исчислении высказываний.

Выбранная таким способом совокупность шести функций является избыточной. Можно показать, что импликация и эквиваленция выражаются через остальные функции этой совокупности:

$$x_1 \to x_2 = \bar{x}_1 \vee x_2;$$
$$x_1 \sim x_2 = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2).$$

Для этого достаточно построить таблицу соответствия и сравнить ее с табл. 1:

x_1	0	0	1	1	
x_2	0	1	0	1	
$\bar{x}_1$	1	1	0	0	
$\bar{x}_2$	1	0	1	0	
$\bar{x}_1 \vee x_2$	1	1	0	1	$x_1 \to x_2$
$x_1 \vee \bar{x}_2$	1	0	1	1	
$(x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2)$	1	0	0	1	$x_1 \sim x_2$

Таким образом, комплект элементарных функций сокращается до четырех: константа 0, отрицание $\bar{x}$, конъюнкция x_1x_2 и дизъюнкция $x_1 \vee x_2$. Этот комплект обладает существенными удобствами и часто применяется на практике, но и он может быть сокращен. Так, из законов де Моргана и свойства двойного отрицания вытекают тождества:

$$x_1 \vee x_2 = \overline{\bar{x}_1 \bar{x}_2}; \quad x_1 x_2 = \overline{\bar{x}_1 \vee \bar{x}_2}.$$

Отсюда следует, что булевы функции выражаются через отрицание и конъюнкцию или через отрицание к дизъюнкцию.

Более того, для записи любой булевой функции достаточно только одной из двух элементарных функций — стрелки Пирса или штриха Шеффера. Это вытекает из соотношений (их доказательство приводится аналогично с помощью таблиц соответствия):

$$\bar{x} = x \downarrow x = x/x;$$
$$x_1x_2 = (x_1/x_2)/(x_1/x_2); \quad x_1 \vee x_2 = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2).$$

8. Булевы функции многих переменных. С помощью суперпозиции функций, т. е. подстановки в логические формулы вместо переменных некоторых булевых функций, можно получить более сложные функции от любого числа переменных. Например, подставляя в выражение ab формулы $a = x_1 \vee x_2$ и $b = x_2 \to c$, а также $c = \bar{x}_3$, получаем $(x_1 \vee x_2)(x_2 \to \bar{x}_3)$. Таблица соответствия для сложных формул записывается на основании общей таблицы для элементарных функций. Для данного примера она имеет вид:

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$x_1 \vee x_2$	0	0	1	1	1	1	1	1
$\bar{x}_3$	1	0	1	0	1	0	1	0
$x_2 \to \bar{x}_3$	1	1	1	0	1	1	1	0
$(x_1 \vee x_2)(x_2 \to \bar{x}_3)$	0	0	1	0	1	1	1	0

Если на всех наборах значений переменных функция принимает значение 0 или 1, то она вырождается в соответствующую константу и называется *тождественным нулем* или *тождественной единицей.* Например,

$$x \vee \bar{x} = 1; \quad x\bar{x} = 0; \quad x\bar{x} \vee x\bar{x}y = 0; \quad ((xy \vee \bar{y}z) \to \bar{z}) \vee \vee (x \vee \bar{y})\, z = 1; \quad x(x \to y) \to y = 1$$ и т. п.

9. Геометрическое представление. Область определения булевых функций от n переменных $y = f(x_1, x_2, \ldots, x_n)$ можно рассматривать как совокупность n-мерных векторов (слов длины n), компонентами которых являются буквы 0 и 1 двоичного алфавита. При $n = 3$ каждый вектор представляется вершиной единичного куба в трехмерном пространстве (рис. 1).

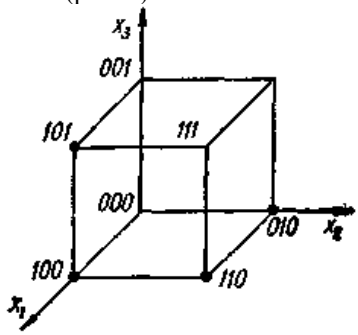


Рис. 1
Отображение булевой функции $y = (x_1 \vee x_2) \times (x_2 \to \bar{x}_3)$
на трехмерном кубе

В общем случае совокупность векторов $(x_1, x_2, \ldots, x_n)$ отображается на множество вершин n-мерного куба. Все такие вершины образуют *логическое пространство.*

Булева функция отображается на n-мерном кубе путем выделения вершин, соответствующих векторам $(x_1, x_2, \ldots, x_n)$, на которых булева функция $y = f(x_1, x_2, \ldots, x_n)$ принимает значения 1. Обычно

такие вершины отмечают жирными точками. Так, на рис. 1 отображена функция $(x_1 \vee x_2)(x_2 \to \bar{x}_3)$ в соответствии с таблицей из (8).

10. Неоднородные функции. Аргументы *неоднородных функций,* в отличие от однородных, могут принимать значения из любых конечных или бесконечных множеств, но область значений самих функций ограничена конечными множествами.

Важным примером неоднородных функций являются двузначные *n-местные предикаты*. Предикат $P(x_1, x_2, \ldots, x_n)$ принимает одно из двух значений — «истинно» (1) или «ложно» (0) в зависимости от конкретных значений, приписываемых переменным $x_1, x_2, \ldots, x_n$. Если значения переменных выбираются из некоторого множества *M* (универсума), то *n*-местный предикат можно рассматривать как *n*-местное отношение, определенное на этом множестве.

Одноместный предикат *P(x)* задает некоторое свойство элементов множества *M* и вполне определяется подмножеством $P \subset M$ тех объектов $x \in M$, на которых он принимает значение «истинно».

Множество объектов, на которых предикат *P(x)* принимает значение «ложно», соответствует дополнению множества *P,* т. е. $\bar{P}$. Очевидно, если *P(x)* истинно, то $\overline{P(x)}$— ложно и наоборот. Например, если на множестве натуральных чисел определен предикат *P(x)* = «*x* — четное число», то $\overline{P(x)}$ = «*x* — нечетное число». Таким образом, одноместный предикат, определенный на множестве *M,* разбивает это множество на два подмножества *P* и $\bar{P}$. Подмножество $P \subset M$, на котором предикат *P(x)* принимает значение «истинно», называется *характеристическим подмножеством.* Пусть на *M* определены два предиката *P(x)* и *Q(x),* характеристическими подмножествами которых являются соответственно *P* и *Q*. Рассматривая предикаты как двузначные функции, можно с помощью операций алгебры логики строить новые одноместные предикаты на множестве *M. Конъюнкция P(x)* и *Q(x)* — это предикат $R(x) = P(x) \wedge Q(x)$, который истинен для тех и только тех объектов из *M,* для которых оба предиката *P(x)* и *Q(x)* истинны. Характеристическим множеством предиката *P(x)* является пересечение $P \cap Q$. Подобным образом вводятся и операции дизъюнкции $P(x) \vee Q(x)$, импликации $P(x) \to Q(x)$, эквиваленции $P(x) \sim Q(x)$ и др. На рис. 2 показаны соответствующие этим операциям характеристические подмножества (область истинных значений заштрихована). Их легко получить из таблиц соответствия для функций двух переменных.

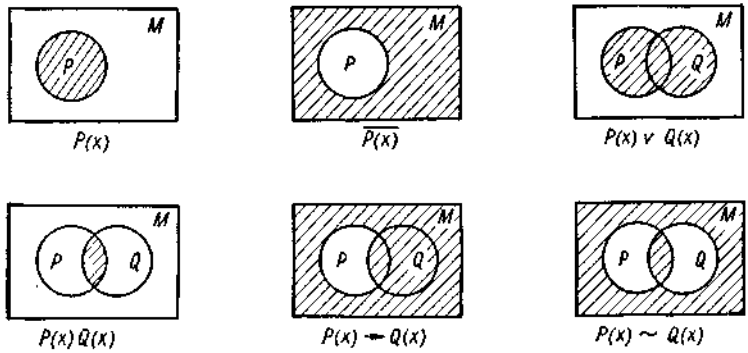


Рис 2 Характеристические подмножества, соответствующие операциям над предикатами (область истинных значений заштрихована)

Имеют место также соответствия между различными операциями, вытекающие из зависимостей между булевыми функциями: $P(x) \to Q(x)$ соответствует $\overline{P}(x) \vee Q(x)$, $P(x) \sim Q(x)$ соответствует $(P(x) \vee \overline{Q(x)})(\overline{P(x)} \wedge Q(x))$ или $P(x)Q(x) \vee \overline{P(x)}\,\overline{Q(x)}$ и т. п.

5.2. Алгебра логики

1. Двойственность формул булевой алгебры. В булевой алгебре, как и в алгебре множеств, имеет место *принцип двойственности.* Взаимно двойственными операциями являются дизъюнкция и конъюнкция. Заменяя в некоторой формуле каждую операцию на двойственную ей, получаем *двойственную формулу.* Например, из формулы $x(y \vee z(u \vee v))$ имеем $x \vee y(z \vee uv)$.

На основе законов де Моргана выводится следующее положение: если $\varphi(x_1, x_2, \ldots, x_n)$ и $\varphi^*(x_1, x_2, \ldots, x_n)$ — двойственные формулы, то $\overline{\varphi}^*(x_1, x_2, \ldots, x_n)$ равносильна $\varphi(\overline{x}_1, x_2, \ldots, x_n)$. Отсюда следует, что $\varphi^*(x_1, x_2, \ldots, x_n) = \overline{\varphi}(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$, т. е. двойственная формула выражается как отрицание формулы, полученной из исходной замещением каждой переменной ее отрицанием. Таблица соответствия двойственной функции получается заменой значений аргументов в исходной функции на противоположные, т. е. 0 заменяется на 1, а 1 — на 0. Формула или функция, равносильная своей двойственной, называется *самодвойственной.*

Если формулы $\varphi_1(x_1, x_2, \ldots, x_n)$ и $\varphi_2(x_1, x_2, \ldots, x_n)$ равносильны,

то и двойственные им формулы $\varphi_1^*(x_1, x_2, \ldots, x_n)$ и $\varphi_2^*(x_1, x_2, \ldots, x_n)$ также равносильны.

2. Нормальные формы. *Дизъюнктивная (конъюнктивная) нормальная форма* — это дизъюнкции (конъюнкция) конечного числа различных членов, каждый из которых представляет собой конъюнкцию (дизъюнкцию) отдельных переменных, или их отрицаний, входящих в данный член не более одного раза.

Данная формула приводится к нормальной форме следующим путем:
1) с помощью законов де Моргана формула преобразуется к такому виду, чтобы знаки отрицания относились только к отдельным переменным;
2) на основе первого (второго) дистрибутивного закона формула сводится к дизъюнкции конъюнкций (конъюнкции дизъюнкций);
3) полученное выражение упрощается в соответствии с тождествами $xx = x$ и $x\bar{x} = 0 (x \vee x = x$ и $x \vee \bar{x} = 1)$.

Пример: $(xy \vee \bar{y}z)\,\bar{x}u = (xy \vee \bar{y}z)(x \vee \bar{u}) = (xy \vee \bar{y}z)\,x \vee (xy \vee \bar{y}z)\,\bar{u} =$
$= xyx \vee \bar{y}zx \vee xy\bar{u} \vee \bar{y}z\bar{u} = xy \vee x\bar{y}z \vee xy\bar{u} \vee \bar{y}z\bar{u}$ (дизъюнктивная нормальная форма);

$$(xy \vee \bar{y}z)\,\bar{x}\bar{u} = (xy \vee \bar{y}z)(x \vee \bar{u}) = (x \vee \bar{y}z)(y \vee \bar{y}z)(x \vee \bar{u}) =$$
$$= (x \vee \bar{y})(x \vee z)(y \vee \bar{y})(y \vee z)(x \vee \bar{u}) = (x \vee \bar{y})(x \vee z)(y \vee z)(x \vee \bar{u})$$

(конъюнктивная нормальная форма).

Члены дизъюнктивной (конъюнктивной) нормальной формы, представляющие собой элементарные конъюнкции (дизъюнкции) k букв, называют *минитермами (макстермами)* k-го ранга. Так, в приведенных выше формах xy — минитерм второго ранга, $x\bar{y}z$ — минитерм третьего ранга, а $x \vee \bar{y}$ — макстерм второго ранга.

Если исходная формула содержит другие операции, то они предварительно выражаются через дизъюнкцию, конъюнкцию и отрицание, например:

$$\overline{x \to (\bar{x} \sim z)}\;(y \to \bar{z}) \vee \overline{x \to z} = \overline{\bar{x} \vee (x \vee z)(\bar{x} \vee \bar{z})}\,(\bar{y} \vee \bar{z}) \vee \overline{\bar{x} \vee z} =$$
$$= x\,\overline{(x \vee z)(\bar{x} \vee \bar{z})}\,(\bar{y} \vee \bar{z}) \vee x\bar{z} = x\,(\bar{x}\bar{z} \vee xz)(\bar{y} \vee \bar{z}) \vee x\bar{z} =$$
$$= (x\bar{x}\bar{z} \vee xxz)(\bar{y} \vee \bar{z}) \vee x\bar{z} = xz(\bar{y} \vee \bar{z}) \vee x\bar{z} = xz\bar{y} \vee xz\bar{z} \vee x\bar{z} = x\bar{y}z \vee x\bar{z}.$$

3. Совершенные нормальные формы. Если в каждом члене нормальной формы представлены все переменные (либо в прямом, либо в инверсном виде), то она называется *совершенной нормальной формой.*

Можно показать, что любая булева функция, не являющаяся тождественным нулем (единицей), имеет одну и только одну совершенную дизъюнктивную (конъюнктивную) нормальную форму. Если какой-либо член φ дизъюнктивной (конъюнктивной) нормальной формы не содержит переменной x_i, то она вводится тождественным

преобразованием $\varphi = \varphi(x_i \vee \bar{x}_i) = \varphi x_i \vee \varphi\bar{x}_i$ (соответственно $\varphi = \varphi \vee x_i\bar{x}_i = (\varphi \vee x_i)(\varphi \vee \bar{x}_i)$). В силу тождеств $\varphi \vee \varphi = \varphi$ и $\varphi\varphi = \varphi$ одинаковые члены, если они появляются, заменяются одним таким членом.

Продолжая второй пример из (2), приведем данную функцию к совершенной дизъюнктивной нормальной форме: $x\bar{y}\bar{z} \vee x\bar{z} = x\bar{y}z \vee x\bar{z}(y \vee \bar{y}) = x\bar{y}z \vee xy\bar{z} \vee x\bar{y}\bar{z}$. Приведение к совершенной конъюнктивной нормальной форме иллюстрируется следующим примером:

$$\bar{x} \vee \overline{y\bar{z}}(x \vee z) = \bar{x}y\bar{z}(x \vee z) = \bar{x}(\bar{y} \vee z)(x \vee z) = (\bar{x} \vee y\bar{y})(\bar{y} \vee z \vee x\bar{x})(x \vee$$
$$\vee z \vee y\bar{y}) = (\bar{x} \vee y)(\bar{x} \vee \bar{y})(\bar{y} \vee z \vee x)(\bar{y} \vee z \vee \bar{x})(x \vee z \vee y)(x \vee z \vee \bar{y}) =$$
$$= (\bar{x} \vee y \vee z\bar{z}) \wedge (\bar{x} \vee \bar{y} \vee z\bar{z})(x \vee \bar{y} \vee z)(\bar{x} \vee \bar{y} \vee z)(x \vee y \vee z)(x \vee \bar{y} \vee z) =$$
$$= (\bar{x} \vee y \vee z)(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee \bar{y} \vee z) \wedge$$
$$\wedge (\bar{x} \vee \bar{y} \vee \bar{z})(x \vee \bar{y} \vee z)(\bar{x} \vee \bar{y} \vee z)(x \vee y \vee z)(x \vee \bar{y} \vee z) =$$
$$= (\bar{x} \vee y \vee z)(\bar{x} \vee y \vee \bar{z}) \wedge$$
$$\wedge (\bar{x} \vee \bar{y} \vee \bar{z})(\bar{x} \vee \bar{y} \vee z)(x \vee \bar{y} \vee z)(x \vee y \vee z).$$

4. Проблема разрешимости. Формула (или соответствующая ей функция) называется *выполнимой,* если она не является тождественным нулем или единицей. Решение с помощью конечного числа действий вопроса, является ли данная формула выполнимой, т. е. не равна ли она тождественно нулю или единице, носит название *проблемы разрешимости.*

Ответ на этот вопрос можно получить, построив для данной формулы таблицу соответствия, что сводится по существу к определению значений формулы при всевозможных наборах значений входящих в нее переменных. Если на всех наборах формула принимает значения только 0 или только 1, то она невыполнима.

При большом количестве переменных такой способ практически неосуществим из-за огромного числа возможных наборов значений переменных. Более удобный путь — приведение формулы к нормальной форме. Если в процессе такого приведения формула не обращается в тождественный 0 или 1, то это свидетельствует о ее выполнимости.

5. Конституенты и представление функций. Для совокупности переменных $x_1, x_2, \ldots, x_n$ выражение $\tilde{x}_1\tilde{x}_2\ldots\tilde{x}_n$ называют *конституентой единицы,* а выражение $\tilde{x}_1 \vee \tilde{x}_2 \vee \ldots \vee \tilde{x}_n$ *—конституентой нуля* ($\tilde{x}_i$ означает либо x_i, либо $\bar{x}_i$). Данная конституента единицы (нуля) обращается в единицу (нуль) только при одном соответствующем ей наборе значений переменных, который получается, если все переменные принять равными единице (нулю), а их отрицания — нулю

(единице). Например, конституенте единицы $x_1\bar{x}_2x_3x_4$ соответствует набор (1011), а конституенте нуля $\bar{x}_1 \vee x_2 \vee x_3 \vee \bar{x}_4$ — набор (1001). Так как совершенная дизъюнктивная (конъюнктивная) нормальная форма является дизъюнкцией (конъюнкцией) конституент единицы (нуля), то можно утверждать, что представляемая ею булева функция $f(x_1, x_2, \ldots, x_n)$ обращается в единицу (нуль) только при наборах значений переменных $x_1, x_2, \ldots, x_n$, соответствующих этим конституентам. На остальных наборах эта функция обращается в нуль (единицу).

Справедливо и обратное утверждение, на котором основан способ представления в виде формулы любой булевой функции, заданной таблицей. Для этого необходимо записать дизъюнкции (конъюнкции) конституент единицы (нуля), соответствующих наборам значений переменных, на которых функция принимает значение, равное единице (нулю). Например функции, заданной таблицей

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
y	0	1	1	0	1	0	0	1

соответствуют совершенные нормальные формы:

$$y = \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3 =$$
$$= (x_1 \vee x_2 \vee x_3)(x_1 \vee \bar{x}_2 \vee \bar{x}_3)(\bar{x}_1 \vee x_2 \vee \bar{x}_3)(\bar{x}_1 \vee \bar{x}_2 \vee x_3).$$

Полученные выражения можно преобразовать к другому виду на основании свойств булевой алгебры.

6. Алгебра Жегалкина. Другая замечательная алгебра булевых функций строится на основе операций сложения по модулю 2 и конъюнкции. Она называется *алгеброй Жегалкина* по имени предложившего ее ученого. Непосредственной проверкой по таблицам соответствия устанавливаются следующие основные свойства этой алгебры:

коммутативность $x + y = y + x$; $xy = yx$;

ассоциативность $x + (y + z) = (x + y) + z$; $x(yz) = (xy)z$;

дистрибутивность умножения относительно сложения $x(y + z) = xy + xz$;

свойства констант $x \cdot 1 = x$; $x \cdot 0 = 0$; $x + 0 = x$.

Все эти свойства подобны обычной алгебре, но в отличие от булевой алгебры закон дистрибутивности сложения относительно умножения не имеет силы $(xy + z \neq xz + yz)$. Справедливы также следующие тождества:

закон приведения подобных членов при сложении $x + x = 0$;

закон идемпотентности для умножения $xx = x$.

Таким образом, в формулах алгебры Жегалкина, как и в булевой алгебре, не могут появляться коэффициенты при переменных и показатели степени. С помощью табл. 1 выводятся также следующие соотношения:

$$\bar{x} = 1 + x;\ x_1 \vee x_2 = x_1 + x_2 + x_1x_2;\ x_1 + x_2 = x_1\bar{x}_2 \vee \bar{x}_1x_2$$

Первые два тождества позволяют перейти от любой формулы булевой алгебры к соответствующей ей формуле алгебры Жегалкина, а с помощью третьего тождества осуществляется обратный переход. Например:

$$x(\bar{x} \vee y) = x[(1 + x) + y + (1 + x)y] = x(1 + x + y + y + xy) =$$
$$= x(1 + x + xy) = x + xx + xxy = x + x + xy = xy;$$
$$1 + x + y + xy = (1 + x)(1 + y) = \bar{x}\bar{y}.$$

Через операции алгебры Жегалкина можно выразить все другие булевы функции:

$$x_1 \to x_2 = \bar{x}_1 \vee x_2 = 1 + x_1 + x_1x_2;$$
$$x_1 \sim x_2 = (\bar{x}_1 \vee x_2)(x_1 \vee \bar{x}_2) = 1 + x_1 + x_2;$$
$$x_1 \leftarrow x_2 = \overline{x_1 \to x_2} = x_1 + x_1x_2;$$
$$x_1/x_2 = \overline{x_1x_2} = 1 + x_1x_2;$$
$$x_1 \downarrow x_2 = \overline{x_1 \vee x_2} = 1 + x_1 + x_2 + x_1x_2.$$

7. Канонические многочлены. Любая булева функция приводится к каноническому многочлену, члены которого не содержат числовых коэффициентов и линейны относительно любой из переменных (переменные входят только в первой степени).

Действительно, если привести данную функцию к совершенной нормальной форме и заменить все знаки дизъюнкции знаками суммы (по модулю 2), а отрицание переменных представить в соответствии с тождеством $\bar{x} = 1 + x$, то после раскрытия скобок получим некоторое алгебраическое выражение. Оно приводится к *каноническому многочлену* на основе соотношений $x+x=0$ и $xx=x$. Такое представление всегда возможно и единственно (с точностью до порядка расположения членов).

Пример: $(1 + x + y)(1 + xy) + (x + xy)y = 1 + x + y + xy +$
$+ xxy + yxy + xy + xyy = 1 + x + y + xy + xy + xy + xy +$
$+ xy = 1 + x + y + xy.$

Проблема разрешимости в алгебре Жегалкина сводится к указанным преобразованиям, в процессе которых делается вывод о выполнимоститой или иной формулы. Например, $x(x \to y) \to y =$
$= x(1 + x + xy) \to y = xy \to y = 1 + xy + xyy = 1 + x\bar{y} +$
$+ xy = 1$; так как эта формула является тождественной единицей, то она невыполнима.

Преимущество алгебры Жегалкина состоит в арифметизации логики, что позволяет выполнять преобразования булевых функций, используя опыт преобразования обычных алгебраических выражений. Ее недостаток по сравнению с булевой алгеброй — сложность формул, что особенно сказывается при значительном числе переменных, например: $x \vee y \vee z = x + y + z + xy + xz + yz + xyz$.

Однако при использовании вычислительных машин различия в сложности выполнения операций булевой алгебры и арифметических операций значительно ослабляются.

8.Типы булевых функций. В алгебре логики из множества $v = 2^{2^n}$ различных булевых функций n переменных $y = f(x_1, x_2, \ldots, x_n)$ выделяются следующие пять типов булевых функций.

1) *Функции, сохраняющие константу* 0, т. е. такие $f(x_1, x_2, \ldots, x_n)$, что $f(0, 0, \ldots, 0) = 0$. Так как на одном из 2^n наборов $(x_1, x_2 \ldots x_n)$ значения таких функций фиксированы, то их число равно $2^{2^n-1} = \frac{1}{2} 2^{2^n} = \frac{1}{2} v$, т. е. половина всех функций n переменных сохраняет константу 0.

2) *Функции, сохраняющие константу* 1, т. е. такие $f(x_1, x_2, \ldots, x_n)$, что $f(1, 1, \ldots, 1) = 1$. Их число, как и в предыдущем случае, равно половине общего числа всех функций n переменных.

3) *Самодвойственные функции,* т. е. такие, которые принимают противоположные значения на любых двух противоположных наборах. Если в общей таблице соответствия наборы, как обычно следуют в порядке их номеров, то противоположные друг другу наборы располагаются симметрично относительно середины их расположения. Это значит,что строка значений самодвойственной функции должна быть антисимметричной относительно своей середины. Самодвойственная функция полностью определяется заданием ее значений на половине всех наборов (остальные значения определяются по условию антисимметричности), поэтому число независимых наборов равно $\frac{1}{2} 2^n$ и число всех таких функций: $2^{\frac{1}{2}2^n} = \sqrt{2^{2^n}} = \sqrt{v}$.

4) *Линейные функции,* т. е. такие, которые представляются в алгебре Жегалкина каноническим многочленом, не содержащем произведений переменных: $a_0 + a_1 x_1 + \ldots + a_n x_n$, где коэффициенты $a_0, a_1 \ldots, a_n$ принимают значения 0 или 1. Так как всего коэффициентов $n+1$, то число различных линейных многочленов будет 2^{n+1}. В силу однозначности представления функции каноническим многочленом это число выражает и количество линейных функций.

5) *Монотонные функции,* т. е. такие, которые для любых двух наборов из множества значений переменных, частично упорядоченного соотношением $(\alpha_1, \alpha_2, \ldots, \alpha_n) \leqslant (\beta_1, \beta_2, \ldots, \beta_n)$ при $\alpha_i \leq \beta_i$ *(i*= 1, 2, ..., *n),* удовлетворяют неравенству

$$f(\alpha_1, \alpha_2, \ldots, \alpha_n) \leqslant f(\beta_1, \beta_2, \ldots, \beta_n).$$

Рассмотренные типы функций замкнуты относительно операции суперпозиции, т. е. суперпозиция любого числа булевых функций данного типа является функцией того же типа.

9. Функциональная полнота. Система функций, суперпозицией которых может быть представлена любая функция из некоторого множества булевых функций, называется *функционально полной.* Если в такой системе допускаются константы 0 и 1, то ее называют *ослабленно функционально полной.* Говорят, что функционально полная система функций образует *базис в логическом пространстве.* Система функций называется минимально полным базисом, если удаление из нее любой функции превращает эту систему в неполную.

Функционально полные системы комплектуются путем сопоставления различных выражений для булевых функций. Общее решение вопроса основано на *теореме о функциональной полноте:* для того чтобы система булевых функций быта полной, необходимо и достаточно, чтобы она включала хотя бы одну функцию: несохраняющую константы 0, несохраняющую константы 1, несамодвойственную, нелинейную и немонотонную. Эту теорему следует понимать так, что одна и та же функция может представлять в функционально полной системе одно или несколько требуемых свойств, если она обладает этими свойствами. С помощью табл. 1 можно следующим образом охарактеризовать свойства булевых функций с позиций функциональной полноты (звездочкой отмечены свойства, которыми обладает данная функция):

Булева функция	Формулы	Свойства				
		Несохранение 0	Несохранение 1	Несамодвойственность	Нелинейность	Немонотонность
Константа 0	0			*		
Константа 1	1	*		*		
Отрицание	$\bar{x}$	*	*			*
Конъюнкция	$x_1 x_2$			*	*	
Дизъюнкция	$x_1 \vee x_2$			*	*	
Импликация	$x_1 \to x_2$	*		*	*	*
Эквиваленция	$x_1 \sim x_2$	*		*		*
Отрицание импликации	$x_1 \leftarrow x_2$		*	*	*	*
Сумма по модулю 2	$x_1 + x_2$		*	*		*
Штрих Шеффера	x_1 / x_2	*	*	*	*	*
Стрелка Пирса	$x_1 \downarrow x_2$	*	*	*	*	*

Отсюда видно, что системы операций (дизъюнкция и отрицание, конъюнкция и отрицание, штрих Шеффера, стрелка Пирса) удовлетворяют теореме о функциональной полноте. Система операций алгебры Жегалкина (сумма по модулю 2 и конъюнкция) вместе с константой 1 образует ослабленно функционально полную систему.

Выбрав любую элементарную функцию и дополнив ее одной или несколькими другими функциями так, чтобы все они вместе удовлетворяли теореме о функциональной полноте, можно выразить через них все другие булевы функции. Например, в основу одного из таких комплектов можно положить импликацию и константу 0. Тогда $x_1 \vee x_2 = (x_1 \to x_2) \to x_2$ и $\bar{x} = x \to 0$, а через дизъюнкцию и отрицание выразятся и все остальные функции. В качестве другого функционально полного комплекта можно взять конъюнкцию, эквиваленцию и константу 0. При этом $\bar{x} = 0 \sim x$ и формулы алгебры логики, построенной на этих операциях, будут двойственны формулам алгебры Жегалкина, если в качестве двойственных символов принять + и ~, а также 1 и 0.

По-видимому, все лучшее, что можно извлечь из различных вариантов функционально полных систем, уже заложено в булевой алгебре и алгебре Жегалкина. Но при решении специальных задач не исключается построение и применение других алгебр логики.

10. Булевы алгебры. Обычно при определении булевой алгебры одну из операций (дизъюнкцию) называют сложением, а другую (конъюнкцию) — умножением и наделяют их свойствами, аналогичными уже рассмотренным свойствам.

Сравнив свойства булевой алгебры и алгебры множеств, легко убедиться, что алгебра множеств также является булевой алгеброй относительно операции объединения $\cup$ и пересечения $\cap$. Роль единицы и нуля играют соответственно исходное множество (универсум) U и пустое множество $\varnothing$, а операции отрицания соответствует дополнение до исходного множества. В то же время алгебра Жегалкина (6) не относится к классу булевых алгебр, так как одна из ее операций (сложение по модулю 2) не является дистрибутивной относительно другой операции (конъюнкции).

Приведем еще один пример булевой алгебры на ограниченном множестве M действительных чисел, содержащем верхнюю p и нижнюю q грани. Операции сложения и умножения (дизъюнкции и конъюнкции) можно определить как $x \vee y = \max(x, y)$ и $xy = \min(x, y)$. Роль 1 и 0 играют соответственно p и q. Отрицание $\bar{x}$ определяется числом, симметричным числу x относительно центра множества $\frac{1}{2}(p+q)$, т. е. предполагается, что множество M симметрично относительно своего центра (сам центр может и не входить в состав множества). Эта алгебра включает и двоичную алгебру как частный случай, когда множество M состоит только из двух чисел 0 и 1, причем $p = 1$ и $q = 0$ (центр $^1/_2$ не входит в M).

5.3. Контактные схемы

1. Контакты. Любую булезу функцию можно реализовать схемой, состоящей из последовательно и параллельно соединенных ключей. Каждый такой ключ может находиться в двух состояниях — разомкнут (0) и замкнут (1), а переход из одного состояния в другое осуществляется каким-либо управляющим органом.

В электрических цепях роль ключей играют многочисленные устройства, предназначенные для коммутации (замыкания и размыкания): выключатели, электромагнитные реле, электронные ключевые схемы и т. п. Обычные выключатели и подобные им устройства управляются человеком. Состояние электромагнитного реле изменяемся под воздействием электрического тока, протекающего по обмотке катушки (рис. 1, *а)*.

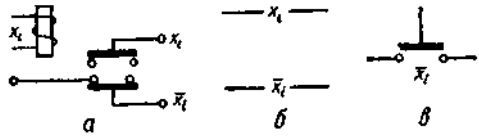


Рис. 1. Контакты:
а —электромагнитное реле; *б* — условное изображение размыкающих (x_i) и замыкающих ($\bar{x}_i$) контактов (*в*)

Ключом в широком смысле является всякое устройство, способное принимать только одно из двух возможных состояний: механические защелки, дверные замки, рычаги управления, железнодорожные светофоры и т. п. Более того, двузначную переменную, независимо от ее конкретного смысла, можно рассматривать, как ключ, состояние которого соответствует значению этой переменной. В рамках общей теории распознавания целесообразно отвлечься от конструктивных и специфических особенностей ключевых объектов и интерпретировать ключ как отрезок проводника с контактом, который можем быть разомкнут или замкнут. Разомкнутое состояние контакта отождествляется с нулем, а замкнутое — с единицей.

Замыкающие (нормально разомкнутые) контакты обозначаются x_i, *размыкающие* (нормально замкнутые) контакты — через $\bar{x}_i$ (рис. 1, б) При управляющем воздействии контакт меняет свое состояние: нормально разомкнутый контакт замыкается, а нормально замкнутый — размыкается. В зависимости от своего состояния контакты пропускают электрический ток или препятствуют его прохождению.

Процесеы переключения в реальных устройствах занимают некоторое, иногда довольно большое время. Однако во многих задачах время переключения можно не учитывать, считая, что контакты переходят из одного состояния в другое мгновенно.

2. Однотактные схемы. Схемы, образованные соединением контактов, которые переключаются одновременно (за один такт), а время переключения не учитывается, называются *однотактными.*

Каждая из них, будучи включена в цепь с источником, в результате совместного действия контактов замыкает или размыкает эту цепь и, следовательно, сама является некоторым контактом по отношению к цепи с источником (рис. 2, *а).* Подобные контактные схемы называют *двухполюсными.*

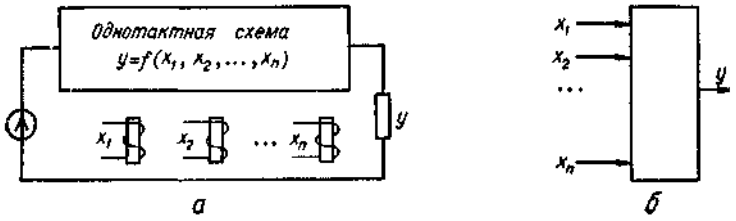


Рис. 2. Контактная схема с n входами ($а$) и ее условное представление *(б).*

Соответствие между двухполюсной контактной схемой и булевой функцией $y = f(x_1, x_2, \ldots, x_n)$ выражается следующим образом: значения переменных $x_1, x_2, \ldots, x_n$ определяются наличием (1) или отсутствием (0) тока в обмотке реле, а значения функции y — состоянием двухполюсной цепи (как и для контактов, 0 соответствует разомкнутой, а 1 — замкнутой цепи).

Независимо от характера ключей двухполюсная контактная схема представляется как схема с n входами $x_1, x_2, \ldots, x_n$ и одним выходом y (2, б). Состояния входов определяют воздействия на контакты схемы, причем вход x_i управляет всеми контактами, обозначенными этой буквой (x_i или $\bar{x}_i$).

3. Анализ контактных схем. Задача *анализа* контактной схемы состоит в построении соответствующей ей булевой функции. Для параллельно-последовательных схем эта задача решается на основе того, что параллельное соединение контакгов соответствует дизъюнкции, а последовательное соединение — конъюнкции переменных, которыми эти контакты обозначены в схеме. Например, для двухполюсной контактной схемы (рис. 3) $y = (x_1 \vee x_2)\bar{x}_3 \vee \bar{x}_2 x_3 \vee (\bar{x}_1 x_3 \vee x_2 \bar{x}_3) x_4$.

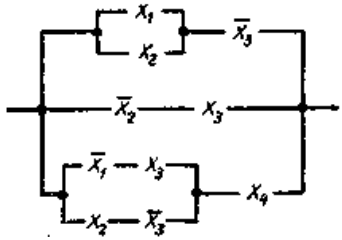


Рис. 3. Контактная схема, соответствующая булевой функции $y = (x_1 \vee x_2)\bar{x}_3 \vee \bar{x}_2 x_3 \vee (\bar{x}_1 x_3 \vee x_2 \bar{x}_3) x_4$

Если схема (или ее часть) имеет произвольную структуру, то ее анализ проводится путем выделения всех путей между входным и выходным полюсами схемы. Каждый такой путь представляется коньюнкцией переменных входящих в нее контактов, а вся схема — дизъюнкцией этих конъюнкций. Например, для мостиковой схемы (рис. 4) $y = x_1\bar{x}_2x_3 \vee x_1x_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3$.

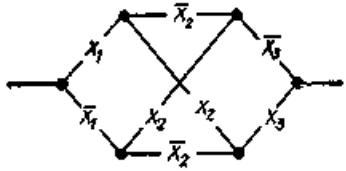
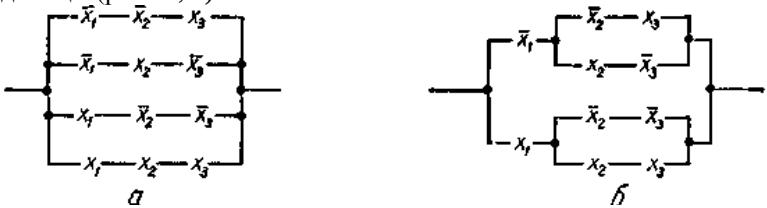


Рис. 4. Мостиковая схема, соотвестиующая булевой функции

$$y = x_1\bar{x}_2x_3 \vee x_1x_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 = x_1 \dotplus x_2 \dotplus x_3$$

Интересно отметить, что эта функция реализует операцию сложения по модулю 2 трех двоичных переменных, т. е. $y = x_1 \dotplus x_2 \dotplus x_3$, в чем можно убедиться по таблицам соответствующих функций.

4. Синтез контактных схем. При построении контактной схемы по заданной булевой функции *(задача синтеза)* исходная функция может быть задана как логической формулой, так и таблицей. В обоих случаях прежде всего необходимо выразить функции через операции конъюнкции, дизъюнкции и отрицания. Каждая операция конъюнкции соответствует последовательному соединению контактов, а операция дизъюнкции — параллельному соединению. В результате получаем последовательно-параллельную контактную схему. Пусть, например, функция задана таблицей соответствия, приведенной в (2.5). На основе ее в совершенной дизъюнктивной нормальной форме строится схема в виде параллельного соединения ветвей, каждая из которых представляет собой последовательное соединение контактов, соответствующих переменным конституент единицы (рис. 5, *а*).



а *б*

Рис. 5. Контактные схемы, соответствующие совершенной дизъюнктивной нормальной форме *(а)* и упрощенному выражению (б) булевой функции

Преобразуя исходное выражение, можно получить другие контактные схемы, соответствующие данной функции. Так, для рассматриваемого примера: $y = \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3 = \bar{x}_1(\bar{x}_2x_3 \vee x_2\bar{x}_3) \vee x_1(\bar{x}_2\bar{x}_3 \vee x_2x_3)$.

Этому выражению соответствует схема рис. 5, *б,* которая содержит на два контакта меньше. Еще проще мостиковая схема (рис. 4), которая реализует ту же функцию.

Центральной проблемой синтеза является построение наиболее простой или в каком-то смысле оптимальной схемы. Часто эта проблема сводится к *минимизации булевых функций,* т. е. к такому их представлению, в котором соответствующие формулы содержат минимальное количество вхождений переменных. Проблема оптимального синтеза еще далека от полного решения, но разработанные методы позволяют существенно упрощать формулу и схемы, а в сравнительно простых случаях получать и оптимальные схемы.

5. Схемы со многими выходами. Если необходимо реализовать несколько булевых функций, то каждая из них может быть представлена соответствующей контактной схемой. Однако такой путь неэкономичен. Более целесообразно построить единую схему с несколькими выходами (рис. 6), соответствующими данной системе функций:

$$y_1 = f_1(x_1, \ldots, x_n);\ y_2 = f_2(x_1, \ldots, x_n);$$
$$\ldots;\ y_m = f_m(x_1, \ldots, x_n).$$

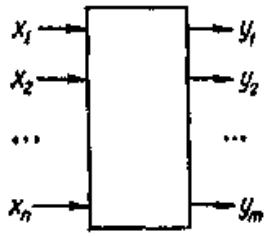


Рис. 6.
Контактная схема с n входами и m выходами

Примером *многовыходной схемы* может служить *полное релейное дерево,* в котором каждая конституента единицы представлена одним выходным полюсом, а всего имеется 2^n выходов (на рис. 7, *а* изображено полное релейное дерево для $n = 3$).

Любую функцию от n переменных можно реализовать объединением выходов полного релейного дерева, которые соответствуют тем наборам переменных, на которых функция принимает значения 1.

Контакты, которые не подсоединены к требуемым выходам, удаляются из схемы.

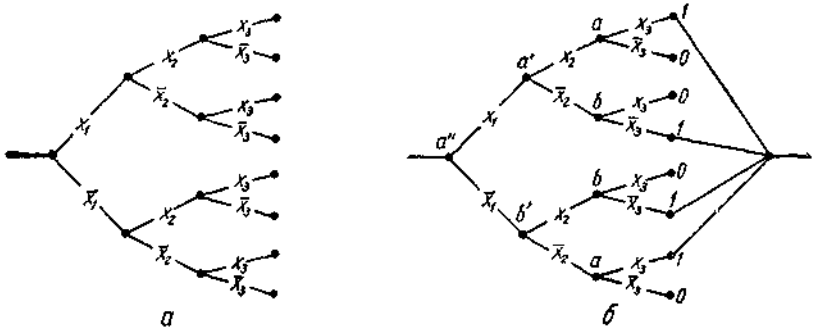


Рис. 7. Полное релейное дерево для трех переменных *(а)* и его преобразование для конкретной функции *(б)*

Например, для функции, заданной таблицей в (2.5), построение приведено на рис. 7, *б*. После упрощения эта схема приводится к виду рис. 5, *б*.

Более простые схемы можно получить объединением участков релейного дерева, общих для путей, которые соответствуют различным конституентам. Для этого обозначаем одинаковыми буквами или цифрами те узлы, из которых выходят пары x_n и $\bar{x}_n$ с совпадающими значениями функции. Дальше аналогично обозначаем одинаковыми буквами узлы, из которых выходят пары x_{n-1} и $\bar{x}_{n-1}$ с совпадающими предыдущими обозначениями (порядок букв также учитывается) и т. д. до последней пары x_1 и $\bar{x}_1$.После этого одинаково обозначенные узлы объединяются и проводятся упрощения в соответствии с рис. 8.

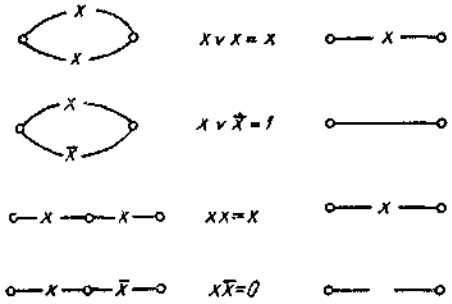


Рис. 8. Упрощение контактных схем для одной переменной

Так, в схеме рис. 7, б для пар $(x_3, \bar{x}_3)$ имеется две комбинации значений (1, 0) и (0, 1). Узлы, из которых выходят пары с комбинациями (1, 0), обозначаем буквой *a,* а узлы, из которых выходят пары с комбинациями (0, 1) — буквой *b.* Для пар $(x_2, \bar{x}_2)$ также встречаются две комбинации в предыдущих обозначениях: *(a, b)* и *(b, a).* Узлы, из которых выходят эти пары, обозначаем соответственно через a' и b'. Наконец, для пары $(x_1, \bar{x}_1)$ имеется единственная комбинация (a', b'), и узел, из которого выходит эта пара, обозначаем через a''. Объединяя узлы с одинаковыми обозначениями *(a* и *b),* приходим к схеме, показанной на рис. 9, которая после замены параллельных контактов x_3 и $x_я$ на x_3, а также $\bar{x}_3$ и $\bar{x}_3$ на x_3, совпадает с мостиковой схемой (рис. 4).

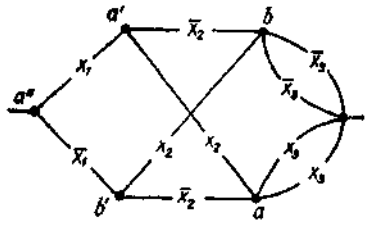


Рис. 9. Преобразование контактной схемы (рис. 7, *б)* к мостиковой (рис. 4)

Объединяя выходы полного релейного дерева, можно построить контактные схемы и для нескольких функций при условии, что множества наборов значений переменных, на которых эти функции принимают значения 1, не пересекаются. Пусть, например, требуется построить контактную схему с двумя выходами, реализующую функции $y_1 = x_1x_2 \vee \bar{x}_1\bar{x}_2x_3$ и $y_2 = x_1\bar{x}_2 \vee \bar{x}_1\bar{x}_3$. Из таблицы соответствия для этих функций

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
y_1	0	1	0	0	0	0	1	1
y_2	1	0	1	0	1	1	0	0

видим, что ни на одном наборе значений переменных функции не принимают одновременно значений, равных 1. Следовательно, для построения требуемой контактной схемы можно воспользоваться полным релейным деревом (рис. 10, *а*), в результате преобразования которого получаем схему с двумя выходами (рис. 10, *б).*

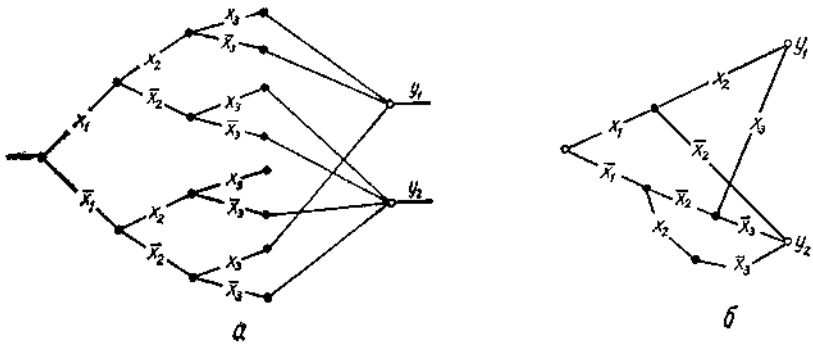


Рис. 10 Построение схемы с двумя выходами:
а— преобразование полного релейного дерева; б — контактная схема

6. Булевы матрицы. Для описания контактных схем произвольной структуры с любым числом выходов используются различные типы *булевых матриц,* элементами которых являются константы 0 и 1, переменные $x_1, x_2, ..., x_n$ и функции этих переменных.

Пусть контактная схема имеет k узлов. *Матрица непосредственных связей (примитивная матрица соединений)* P — это квадратная таблица $k \times k$, элементы главной диагонали которой равны 1, а элементы $p_{ij} = p_{ji}$ представляют собой булеву функцию прямого соединения между узлами i и j. *Матрица полных связей (полная матрица соединений)* Q отличается тем, что ее элементы $q_{ij} = q_{ji}$ представляют собой булеву функцию с учетом всевозможных путей без циклов между узлами i и j. Так, для схемы рис. 11 имеем:

$$P = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & x_4 & x_2 \\ 0 & x_4 & 1 & x_3 \\ x_1 & x_2 & x_3 & 1 \end{bmatrix};$$

$$Q = \begin{bmatrix} 1 & x_1(x_2 \vee x_3x_4) & x_1(x_3 \vee x_2x_4) & x_1 \\ x_1(x_2 \vee x_3x_4) & 1 & x_4 \vee x_2x_3 & x_2 \vee x_3x_4 \\ x_1(x_3 \vee x_2x_4) & x_4 \vee x_2x_3 & 1 & x_3 \vee x_2x_4 \\ x_1 & x_2 \vee x_3x_4 & x_3 \vee x_2x_4 & 1 \end{bmatrix}.$$

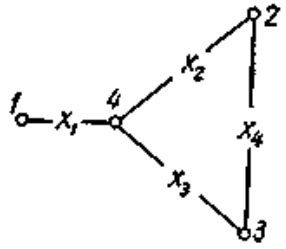


Рис. 11. К определению булевых матриц контактной схемы.

Произведение булевых матриц определяется, как и для обычных матриц, правилом «строка на столбец», но операциям сложения и умножения действительных чисел соответствуют дизъюнкция и конъюнкция логических переменных и функций. Элементы матрицы $C = AB$, где A и B — булевы матрицы, выражаются соотношением $c_{ij} = a_{i1}b_{1j} \vee a_{i2}b_{2j} \vee \ldots \vee a_{in}b_{nj}$. Произведения матрицы самой на себя выражаются как ее степени $AA = A^2$, $A^2A = A^3, \ldots, A^{n-1}A = A^n$.

Можно показать, что для любой контактной схемы с k узлами существует такое $r \leqslant k - 1$, что $P^r = P^{r+s} = Q$, где s — произвольное целое положительное число. Это значит, что матрицу полных связей можно получить умножением матрицы непосредственных связей P на саму себя до тех пор, пока результат не начнет повторяться, причем число таких умножений не превышает k — 1. Так, для рассматриваемого примера имеем:

$$P^2 = \begin{bmatrix} 1 & x_1x_2 & x_1x_3 & x_1 \\ x_1x_2 & 1 & x_4 \vee x_2x_3 & x_2 \vee x_3x_4 \\ x_1x_3 & x_4 \vee x_2x_3 & 1 & x_3 \vee x_2x_4 \\ x_1 & x_2 \vee x_3x_4 & x_3 \vee x_2x_4 & 1 \end{bmatrix}; \quad P^3 = Q.$$

Следует отметить, что элементы матрицы P^i представляют собой функции всех связей между узлами посредством не более чем i — 1 узлов. В частности, каждый элемент матрицы P^2 учитывает непосредственные связи между парой узлов и связи между ними посредством еще одного узла. Например, $p_{23} = p_{32} = x_4 \vee x_2x_3$ соответствует непосредственной связи между узлами 2 и 3 через контакт x_4, а также связи посредством узла 4 (член x_2x_3).

7. Исключение узлов (анализ). При анализе контактной схемы с помощью булевых матриц сначала записывается матрица непосредственных связей P, а затем путем возведения ее в соответствующую степень получается матрица полных связей Q. Элементы q_{ij}

матрицы Q и представляют собой булевы функции данной контактной схемы между парами узлов с номерами i и j.

Однако такой способ в большинстве случаев не является рациональным, так как обычно представляют интерес только некоторые из функций q_{ij} между внешними узлами (полюсами) схемы. Поэтому имеет смысл предварительно исключить внутренние узлы и таким образом уменьшить порядок матрицы P, прежде чем возводить ее в требуемую степень. При исключении s-го узла в матрице непосредственных связей вычеркиваются s-я строка и s-й столбец и каждый ее элемент p_{ij} заменяется элементом $p_{ij} \vee p_{is}p_{sj}$. Член $p_{is}p_{sj}$ учитывает путь между узлами i и j через узел s, который действует параллельно с непосредственной связью p_{ij}. В результате исключения узла матрица P преобразуется к матрице P_s на единицу меньшего порядка, которая представляет собой матрицу непосредственных связей относительно неисключенной совокупности узлов. Пусть, например, в схеме рис. 12 требуется определить булевы функции между узлами 1, *2* и *3*.

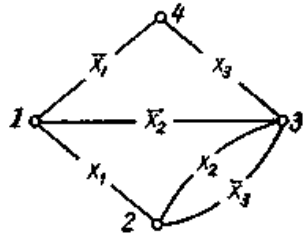


Рис. 12. Контактная схема к примеру

Матрицы P и P_4 имеют вид:

$$P = \begin{bmatrix} 1 & x_1 & \bar{x}_2 & \bar{x}_1 \\ x_1 & 1 & x_2 \vee x_3 & 0 \\ \bar{x}_2 & x_2 \vee \bar{x}_3 & 1 & x_3 \\ \bar{x}_1 & 0 & x_3 & 1 \end{bmatrix}; \quad P_4 = \begin{bmatrix} 1 & x_1 & \bar{x}_2 \vee \bar{x}_1 x_3 \\ x_1 & 1 & x_2 \vee \bar{x}_3 \\ \bar{x}_2 \vee x_1 x_3 & x_2 \vee \bar{x}_3 & 1 \end{bmatrix}.$$

Определив P^2_4 после преобразований, получим матрицу полных связей относительно узлок 1, *2* и *3*, напиваемую *матрицей выходов:*

$$F = \begin{bmatrix} 1 & x_1 \vee (x_2 \vee \bar{x}_3)(\bar{x}_2 \vee \bar{x}_1 x_3) & \bar{x}_2 \vee \bar{x}_1 x_3 \vee x_1(x_2 \vee \bar{x}_3) \\ x_1 \vee (x_2 \vee \bar{x}_3)(\bar{x}_2 \vee \bar{x}_1 x_2) & 1 & x_1\bar{x}_2 \vee x_2 \vee x_3 \\ \bar{x}_2 \vee \bar{x}_1 x_3 \vee x_1(x_2 \vee \bar{x}_3) & x_1\bar{x}_2 \vee x_2 \vee x_3 & 1 \end{bmatrix}.$$

Элементы этой матрицы являются функциями выходов: $f_{12} = x_1 \vee (x_2 \vee \bar{x}_3)(\bar{x}_2 \vee \bar{x}_1 x_2)$; $f_{13} = \bar{x}_2 \vee \bar{x}_1 x_3 \vee x_1(x_2 \vee \bar{x}_3)$; $f_{23} = x_1\bar{x}_2 \vee x_2 \vee x_3$.

8. Введение узлов (синтез). При синтезе контактных схем задаются функции для внешних узлов (полюсов), которые определяют матрицу выходов. Необходимое и достаточное условие непротиворечивости этих функций состоит в том, что матрица выходов должна быть *устойчивой,* т. е. удовлетворять равенству $F = F^2$.

Структуру контактной схемы, реализующей заданную непротиворечивую совокупность функций, можно получить из матрицы F путем ее последовательного расширения, соответствующего *операции введения узла.* Эта операция обратная исключению узла и приводит к матрице F_s, порядок которой на единицу выше, а элементы таковы, что при исключении узла s снова получим матрицу F. Последовательным применением операции введения узла исходная матрица расширяется и преобразуется к виду, при котором элементы представляют собой константы 0 или 1, переменные, их отрицания или элементарные конъюнкции переменных. Тогда полученную матрицу можно рассматривать как матрицу непосредственных связей, на основе которой легко построить соответствующую контактную схему. При этом элементарные конъюнкции реализуются последовательными соединениями соответствующих контактов.

Операция введения неоднозначна, поэтому можно получать различные схемы, удовлетворяющие заданным функциям. Выбор наилучшего пути преобразования матрицы F к матрице непосредственных связей P, определяющей вид контактной схемы, в значительной степени зависит от искусства ЛРО.

Пусть требуется построить контактную схему со следующими функциями; $f_{12} = \bar{x}_1\bar{x}_2 \vee x_1x_3$; $f_{13} = \bar{x}_3(x_2 \vee x_1x_4)$; $f_{23} = 0$. Матрица выходов имеет вид:

$$F = \begin{bmatrix} 1 & \bar{x}_1\bar{x}_2 \vee x_1x_3 & \bar{x}_3(x_2 \vee x_1x_4) \\ \bar{x}_1\bar{x}_2 \vee x_1x_3 & 1 & 0 \\ \bar{x}_3(x_2 \vee x_1x_4) & 0 & 1 \end{bmatrix}.$$

Элементы этой матрицы можно рассматривать как результат исключения узла 4, который мы должны ввести, т. е. $\bar{x}_1\bar{x}_2 \vee x_1x_3 = f'_{12} \vee f'_{14}f'_{42}$, $\bar{x}_3(x_2 \vee x_1x_4) = f'_{13} \vee f'_{14}f'_{43}$ и $0 = f'_{23} \vee f'_{24}f'_{43}$.

Полагая $f'_{14} = x_2 \vee x_1x_4$ и $f'_{43} = \bar{x}_3$ (возможны и другие варианты), имеем $f'_{13} = f'_{42} = f'_{23} = f'_{24} = 0$ и $f'_{12} = \bar{x}_1\bar{x}_2 \vee x_1x_3$. Таким образом, в результате введения узла 4 имеем матрицу

$$F_{(4)} = \begin{bmatrix} 1 & \bar{x}_1\bar{x}_2 \vee x_1x_3 & 0 & x_2 \vee x_1x_4 \\ \bar{x}_1\bar{x}_2 \vee x_1x_3 & 1 & 0 & 0 \\ 0 & 0 & 1 & \bar{x}_3 \\ x_2 \vee x_1x_4 & 0 & \bar{x}_3 & 1 \end{bmatrix}.$$

Продолжая аналогично, можно записать соотношения для элементов матрицы $F_{(4,5)}$, соответствующей введению узла 5: $\bar{x}_1\bar{x}_2 \vee x_1x_3 = f''_{12} \vee f''_{15}f''_{52}$; $0 = f''_{13} \vee f''_{15}f''_{53}$; $x_2 \vee x_1x_4 = f''_{14} \vee f''_{15}f''_{54}$; $0 = f''_{23} \vee f''_{25}f''_{53}$; $0 = f''_{24} \vee f''_{25}f''_{54}$; $\bar{x}_3 = f''_{31} \vee f''_{35}f''_{51}$. Если принять $f''_{15} = x_1$, то необходимо положить $f''_{12} = \bar{x}_1\bar{x}_2$; $f''_{52} = x_3$; $f''_{13} = f''_{53} = 0$; $f''_{14} = x_2$; $f''_{51} = x_4$; $f''_{23} = f''_{25} = f''_{24} = 0$. В результате приходим к матрице, Которую можно рассматривать как матрицу непосредственных связей P синтезируемой схемы:

$$P = F_{(4,5)} = \begin{bmatrix} 1 & \bar{x}_1\bar{x}_2 & 0 & x_2 & x_1 \\ \bar{x}_1\bar{x}_2 & 1 & 0 & 0 & x_3 \\ 0 & 0 & 1 & \bar{x}_3 & 0 \\ x_2 & 0 & \bar{x}_3 & 1 & x_4 \\ x_1 & x_3 & 0 & x_4 & 1 \end{bmatrix}.$$

Схема, соответствущая этой матрице, показана на рис. 13.

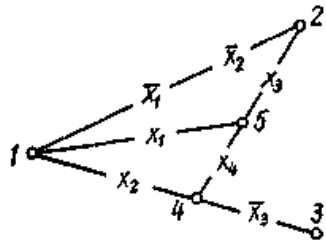


Рис. 13. Схема, построенная по матрице непосредственпых связей.

9. Вентильные схемы. До сих пор предполагалось, что контакты обладают двусторонней проводимостью, т. е. в открытом состоянии они пропускают сигналы как в прямом, так и в обратном направлениях. Таковы, например, контакты электромагнитных реле. Однако при использовании электронных ключей, например управляемых диодов, проводимость в прямом направлении настолько превышает проводимость в обратном направлении, что практически можно считать контакты односторонними, т. е. пропускающими сигналы

только в прямом направлении. Схемы с односторонними контактами называют *вентильными схемами.*

На вентильных схемах, как и ранее, изображаются только соединения контактов, а управляющие цепи обычно опускаются. При этом предполагается, что управление осуществляется как сигналами, соответствующим переменными $x_1, x_2, \ldots, x_n$, так и их отрицаниям $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$, что отмечается на схеме одним из символов x_i или $\bar{x}_i$ для каждого контакта. Кроме того, в вентильных схемах обычно имеет место естественное разделение сигналов: если к узлу схемы одновременно поступают несколько сигналов, то результирующий сигнал в этом узле действует как их дизъюнкция. Направления прохождений сигналов обозначаются на схемах стрелками, относящимися к соответствующим контактам. Пример вентильной схемы показан на рис. 14.

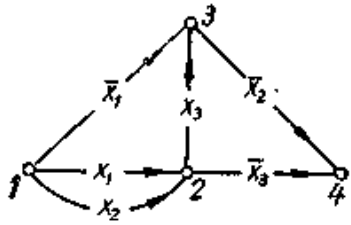


Рис. 14. Вентильная схема.

Булевы матрицы вентильных схем в общем случае несимметричны. Так, для приведенной схемы имеем:

$$P = \begin{bmatrix} 1 & x_1 \vee x_2 & \bar{x}_1 & 0 \\ 0 & 1 & x_3 & \bar{x}_3 \\ 0 & 0 & 1 & \bar{x}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Q = \begin{bmatrix} 1 & x_1 \vee x_2 & \bar{x}_1 \vee x_3 & \bar{x}_2 \vee \bar{x}_3 \\ 0 & 1 & x_3 & \bar{x}_2 \vee x_3 \\ 0 & 0 & 1 & \bar{x}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

При этом в соответствии с (6) $Q = P^3$. Матрицу Q можно также записать непосредственно из вентильной схемы, учитывая для ее элементов q_{ij} все пути от i-го узла к j-му узлу по направлению стрелок. Так, $q_{12} = x_1 \vee x_2$; $q_{13} = \bar{x}_1 \vee (x_1 \vee x_2)x_3 = \bar{x}_1 \vee x_1x_3 \vee x_2x_3 = \bar{x}_1 \vee x_3 \vee x_2x_3 = \bar{x}_1 \vee x_3$; $q_{14} = \bar{x}_1\bar{x}_2 \vee (x_1 \vee x_2)(\bar{x}_3 \vee x_3\bar{x}_2) = \bar{x}_2 \vee \bar{x}_3$ и т.д. Булева функция для любого выхода может быть определена также последовательным исключением узлов, кроме входного и выходного.

Синтез вентильных схем осуществляется аналогично изложенному в (8), причем в исходной матрице выходов все функции, кроме заданных, обычно полагаются тождественно равными нулю. Пусть, например,

$$f_{12} = x_1x_2 \vee \bar{x}_1\bar{x}_3 \text{ и } f_{13} = x_1\bar{x}_3 \vee \bar{x}_1x_2.$$

Матрица выходов и ее расширенияимеют вид:

$$F = \begin{bmatrix} 1 & x_1x_2 \vee \bar{x}_1\bar{x}_3 & x_1\bar{x}_3 \vee \bar{x}_1x_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad F_{(4)} = \begin{bmatrix} 1 & \bar{x}_1\bar{x}_3 & \bar{x}_1x_2 & x_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & x_2 & \bar{x}_3 & 1 \end{bmatrix};$$

$$F_{(4,5)} = \begin{bmatrix} 1 & 0 & 0 & x_1 & \bar{x}_1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & x_2 & \bar{x}_3 & 1 & 0 \\ 0 & \bar{x}_3 & x_2 & 0 & 1 \end{bmatrix}.$$

Схемы, соответствующие $F_{(4)}$ и $F_{(4,5)}$ показаны на рис. 15.

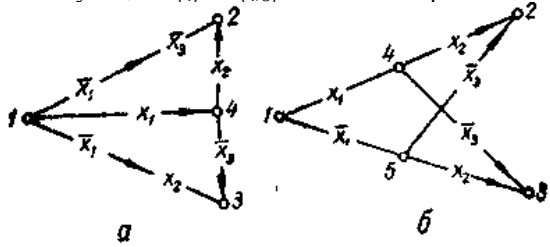


Рис. 15. Схемы, реализующие функции $f_{12} = x_1x_2 \vee \bar{x}_1\bar{x}_3$ и $f_{13} = x_1\bar{x}_3 \vee \bar{x}_1x_2$.

а — с четырьмя узлами; *б* — с пятью узлами

Как видно, вторая схема (рис. 15, б) содержит на один контакт меньше, чем первая (рис. 15, *а).*

10. Криотронные схемы. Существенным ключевым элементом является пленочный криотрон, действие которого основано на явлении сверхпроводимости при низких температурах. Условное изображение криотрона показано на рис. 16.

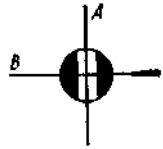


Рис. 16. Условное изображение криотрона

При отсутствии тока в управляющей шине материал (например, олово) обладает сверхпроводимостью, а при прохождении по шине *A* тока достаточной величины этот материал имеет конечное сопротивление. В результате цепь *B* действует как двусторонний управляемый контакт, причем для управления используются сигналы, соответствующие переменным *x,* и их отрицаниям $\bar{x}_i$.

Для анализа и синтеза криотронных схем применяют все рассмотренные методы с учетом специфических особенностей криотронов. Например, на рис. 17, *а* показана *криотронная схема с инверсными выходами,* реализующая функции $y = x_1x_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2$ и $\bar{y} = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_2x_3$, а на рис. 17, *б* — соответствующая ей последовательно-параллельная контактная схема.

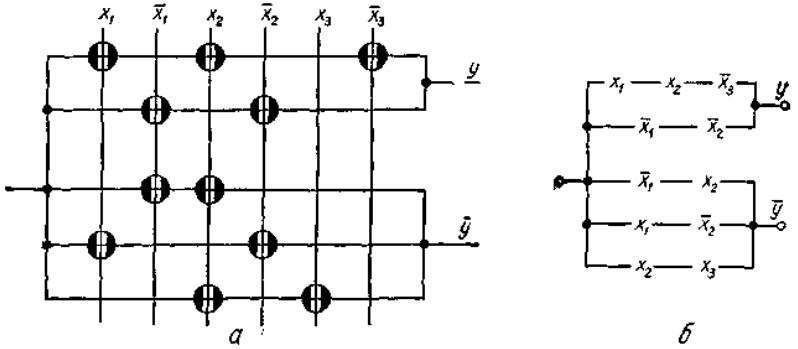


Рис. 17. Криотронная схема с инверсными выходами *(а)* и соответствующая ей контактная схема (б).

Аналогично используются полное криотрониое дерево, булевы матрицы и т. п.

5.4. Логические схемы

1. Логические элементы. Контактные схемы исторически были первыми техническими средствами реализации булевых функций и первыми объектами применения алгебры логики для решения технических задач. Впоследствии появилось много различных устройств, реализующих элементарные булевы функции одной и нескольких переменных. Они основаны на использовании электронных и магнитных цепей, параметронов, струйной техники (пневмоники) и т. д.

Устройства, реализующие элементарные булевы функции, называют *логическими элементами.* Их входы соответствуют булевым переменным, а выход — реализуемой функции.

В теории распознавания для обозначения логических элементов используют различные графические символы и названия, которые учитывают свойства и специфические особенности конкретных элементов. В теории принимаются упрощенные изображения в виде прямоугольников или других фигур, внутри которых помещаются условные названия или символы соответствующей функции (табл. 1). Обычно рассматривают элементы с одним (для отрицания) и двумя входами (для функций двух переменных).

2. Логические схемы. Подобно суперпозиции функций логические схемы образуются *суперпозицией элементов* посредством объединения их внешних узлов (полюсов). При этом множество всех узлов схемы разбивается на *входные, выходные* и *внутренние узлы.* Например, на рис. 1, *а* показана схема, реализующая функцию $y = (x_1/x_2) + (\bar{x}_3 \to x_1)$, которая имеет три входных, один выходной и три внутренних узла. Обычно для упрощения узлы на схемах не изображаются и во избежание излишних пересечений входы рассредоточиваются с указанием связанных с ними переменных (рис. 1, *б).*

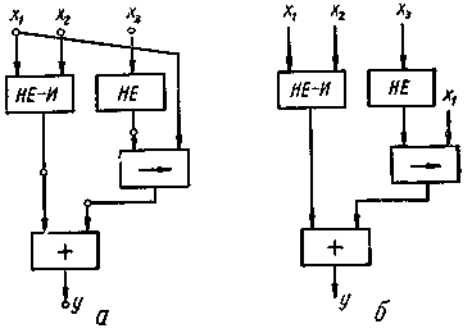
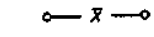
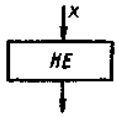
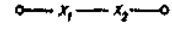
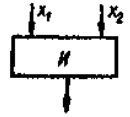
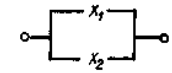
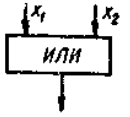
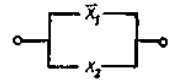
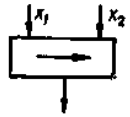
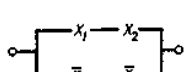
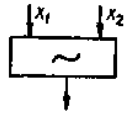
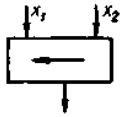
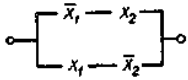
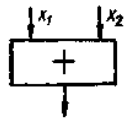


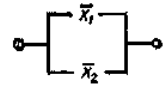
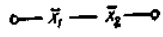
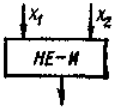
Рис. 1. Логическая схема (*а*) и ее упрощенное изображение (*б*).

Таблица 1

Логические элементы, реализующие элементарные булевы функции

Функция	Нормальная форма	Контактная схема	Графическое изображение элемента	Название элемента
Отрицание $\bar{x}$	$\bar{x}$			Инвертор
Конъюнкция x_1x_2	x_1x_2			Совпадение
Дизъюнкция $x_1 \vee x_2$	$x_1 \vee x_2$			Разделение
Импликация $x_1 \to x_2$	$\bar{x}_1 \vee x_2$			Разделение с запретом
Эквиваленция $x_1 \sim x_2$	$x_1x_2 \vee \bar{x}_1\bar{x}_2$			Равнозначность
Отрицание импликации $x_1 \leftarrow x_2$	$x_1\bar{x}_2$			Совпадение с запретом
Сумма по модулю 2 $x_1 + x_2$	$\bar{x}_1x_2 \vee x_1\bar{x}_2$			Неравнозначность

Продолжение табл. 1

Функция	Нормальная форма	Контактная схема	Графическое изображение элемента	Название элемента
Штрих Шеффера x_1/x_2	$\bar{x}_1 \vee \bar{x}_2$			Разделение с двумя запретами
Стрелка Пирса $x_1 \downarrow x_2$	$\bar{x}_1\bar{x}_2$			Совпадение с двумя запретами

Корректно построенные схемы должны удовлетворять следующим условиям:

1) не допускать замкнутых контуров, которые могут привести к неоднозначности сигналов на входах элементов;

2) любой вход элемента должен быть связан только с одним входом схемы или выходом другого элемента;

3) выходы элементов, не являющиеся выходами схемы и не связанные со входами других элементов, считаются лишними и исключаются из схемы.

Не составляет большого труда записать булеву функцию для данной логической схемы. Так же просто строится логическая схема для данного аналитического выражения булевой функции. Однако задача проектирования логических схем состоит в том, чтобы обеспечить наиболее экономичную реализацию булевой функции в некотором базисе, который обусловлен имеющимся в распоряжении инженера набором логических элементов или выбирается по соображениям наибольшей простоты реализации данного класса функций.

3. Реализация п различных базисах. Прежде всего исходная функция преобразуется к такому виду, чтобы она представляла собой суперпозицию только тех функций, которые входят в данный базис. Например, в базисе, состоящем из отрицания, конъюнкции и дизъюнкции, функция из (2) преобразуется к виду $y = \overline{(x_1/x_2)} + (\bar{x}_3 \to x_1) = \overline{x_1x_2} + (x_3 \vee x_1) = \overline{(x_1x_2 \vee x_3 \vee x_1)\,(\overline{x_1x_2} \vee \overline{x_3 \vee x_1})}$. Ее реализация в системе базисных элементов $\{НЕ, И, ИЛИ\}$ показана на рис. 2, *а*.

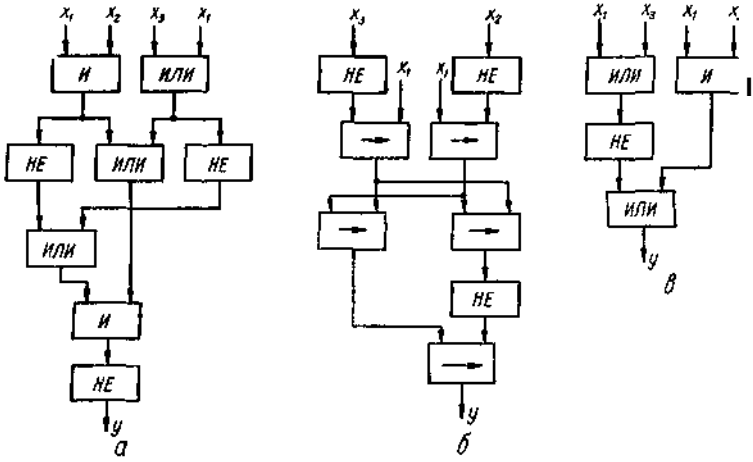


Рис. 2. Логические схемы, реализующие функцию $y = (x_1/x_2) + (\bar{x}_3 \to x_1)$:
а — в базисе {НЕ, И, ИЛИ}; *б* — в базисе {НЕ, →}; *в* — упрощенная схема в базисе {НЕ, И, ИЛИ}.

Если в качестве базиса приняты отрицание и импликация, то функцияпреобразуется по формулам: $x_1 \vee x_2 = \bar{x}_1 \to x_2$; $x_1 x_2 = \overline{x_1 \to \bar{x}_2}$; $x_1/x_2 = x_1 \to \bar{x}_2$; $x_1 \downarrow x_2 = \overline{\bar{x}_1 \to x_2}$; $x_1 \sim x_2 = (x_1 \to \bar{x}_2) \to \bar{x}_1 \to x_2$; $x_1 + x_2 = (\bar{x}_1 \to \bar{x}_2) \to \overline{x_1 \to x_2} = (x_2 \to x_1) \to \overline{x_1 \to x_2}$.
Так, для рассматриваемого примера имеем:

$$y = (x_1/x_2) + (\bar{x}_3 \to x_1) =$$
$$= (x_1 \to \bar{x}_2) + (\bar{x}_3 \to x_1) = ((\bar{x}_3 \to x_1) \to (x_1 \to \bar{x}_2)) \to \overline{(x_1 \to \bar{x}_2) \to (\bar{x}_3 \to x_1)}.$$

соответствующаялогическая схема в базисе $\{HE, \to\}$ изображена на рис. 2, *б*.

Аналогично реализуются схемы и в других базисах. Как правило, в практике используются неминимальные базисы, так как минимальные базисы не всегда обеспечивают наиболее экономичную реализацию булевых функций.

4. Упрощение формул. Между формулой, выражающей булеву функцию, и функциональной схемой, реализующей эту функцию, имеется функциональное соответствие. Однако, поскольку одна и та же функция может быть выражена различными формулами, ее реализация неоднозначна. Всегда можно построить много различных

логических схем, соответствующих данной логической функции. Такие схемы называют *эквивалентными.*

Из множества эквивалентных схем можно выделить наиболее экономичную или хотя бы достаточно простую схему путем упрощения формулы, соответствующей данной функции. Обычно принято считать более простыми те формулы, которые содержат меньшее количество вхождений переменных и символов логических операций.

Задача упрощения аналитических выражении решается в конкретном базисе с помощью тождественных преобразований. Чаще всего эту задачу связывают с базисом, состоящим из отрицания, дизъюнкции н конъюнкции, который будем называть *булевым базисом.* После того как формула выражена через основные операции, она упрощается на основании тождеств булевой алгебры, приведенных в (5.2. 1).

Например, функция из (5.3) упрощается следующим образом:

$$y = (x_1/x_2) \dotplus (\bar{x}_3 \to x_1) = \overline{x_1 x_2} \dotplus (x_3 \vee x_1) = x_1 x_2 (x_3 \vee x_1) \vee \overline{x_1 x_2} x_3 \overline{\vee x_1} =$$
$$= x_1 x_2 \vee x_1 x_2 (x_3 \vee x_1) = x_1 x_2 \vee x_1 \vee x_3.$$

Соответствующая логическая схема показана на рис. 2, *в.*

5. Минимальные формы. Как было показано в (5.2.3), любая булева функция представима в совершенной нормальной форме (дизъюнктивной или конъюнктивной). Более того, такое представление является первым шагом перехода от табличного задания функции к ее аналитическому выражению. В дальнейшем будем исходить из дизъюнктивной формы, а соответствующие результаты для конъюнктивной формы получаются на основе принципа двойственности (5.2. 1).

Каноническая задача синтеза логических схем в булевом базисе сводится к минимизации булевых функций, т. е. к представлению их в дизъюнктивой нормальной форме, которая содержит наименьшее число букв (переменных и их отрицаний). Такие формы называют *минимальными.* При каноническом синтезе предполагается, что ни входы схемы подаются как сигналы x_i, так и их инверсии $\bar{x}_i$.

Формула, представленная в дизъюнктивной нормальной форме, упрощается многократным применением *операции склеивания* $ab \vee a\bar{b} = a$ и *операций поглощения* $a \vee ab = a$ и $a \vee \bar{a}b = a \vee b$ (дуальные тождества для конъюнктивной нормальной формы имеют вид: $(a \vee b)(a \vee \bar{b}) = a$; $a(a \vee b) = a$ и $a(\bar{a} \vee b) = ab$). ... Здесь под a и b можно понимать любую формулу булевой алгебры. В результате приходим к такому аналитическому выражению, когда дальнейшие преобразования оказываются уже невозможными, т. е. получаем *тупиковую форму.*

Среди тупиковых форм находится и минимальная дизъюнктивная форма, причем она может быть неединственной. Чтобы убедиться в том, что данная тупиковая форма является минимальной, необходимо найти все тупиковые формы и сравнить их по числу входящих в них букв.

Пусть, например, функция задана в совершенной нормальной дизъюнктивной форме: $y = \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3$. Группируя члены и применяя операцию склеивания, имеем

$$y = (\bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3) \vee (x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3) \vee x_1x_2x_3 = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_1x_2x_3.$$

При другом способе группировки получим

$$y = \bar{x}_1x_2\bar{x}_3 \vee (\bar{x}_1x_2x_3 \vee x_1x_2x_3) \vee (x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3) = \bar{x}_1x_2\bar{x}_3 \vee x_2x_3 \vee x_1\bar{x}_3.$$

Обе тупиковые формы не являются минимальными. Чтобы получить минимальную форму, нужно догадаться повторить в исходной формуле один член (это всегда можно сделать, так как $x \vee x = x$). В первом случае таким членом может быть $\bar{x}_1x_2x_3$. Тогда

$$y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee (x_1x_2x_3 \vee \bar{x}_1x_2x_3) = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_2x_3.$$

Добавив член $x_1\bar{x}_2x_3$, получим:

$$y = x_1\bar{x}_2 \vee x_1\bar{x}_2 \vee (x_1x_2x_3 \vee x_1\bar{x}_2x_3) = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_1x_3.$$

Перебрав все возможные варианты, можно убедиться, что две последние формы являются минимальными.

Работа с формулами на таком уровне подобна блужданию в потемках. Процесс поиска минимальных форм становится более наглядным и целеустремленным, если использовать некоторые графические и аналитические представления и специально разработанную для этой цели символику.

6. Многомерный куб. Каждой вершине n-мерного куба, можно поставить в соответствие конституенту единицы (5.2.5). Следовательно, подмножество отмеченных вершин является отображением на n-мерном кубе булевой функции от n переменных в совершенной дизъюнктивной нормальной форме. На рис. 3 показано такое отображение для функции из (5).

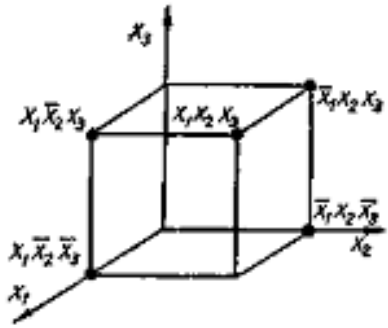


Рис 3. Отображение на трехмерном кубе функции, представленной в совершенной дизъюнктивной нормальной форме

Для отображения функции от n переменных, представленной в любой дизъюнктивной нормальной форме, необходимо установить соответствие между ее минитермами (5.2. 2) и элементами n-мерного куба.

Минитерм $(n$ — 1)-го ранга φ_{n-1} можно рассматривать как результат склеивания двух минитермов n-го ранга (конституент единицы), т. е. $\varphi_{n-1} = \varphi_{n-1}x_i \vee \varphi_{n-1}\bar{x}_i$. На n-мерном кубе это соответствует замене двух вершин, которые отличаются только значениями координаты x_i, соединяющим эти вершины ребром (говорят, что ребро *покрывает* инцидентные ему вершины). Таким образом, минитермам $(n$ — 1)-го порядка соответствуют ребра n-мерного куба. Аналогично устанавливается соответствие минитермов (n- 2)-го порядка граням n-мерного куба, каждая из которых покрывает четыре вершины (и четыре ребра).

Элементы n-мерного куба, характеризующиеся s измерениями, называют *s-кубами.* Так, вершины являются 0-кубами, ребра — 1-кубами, грани — 2-кубами и т.д. Обобщая приведенные рассуждения, можно считать, что минитерм $(n$ - $s)$-го ранга в дизъюнктивной нормальной форме для функции n переменных отображается s-кубом, причем каждый s-куб покрывает все те s-кубы низшей размерности, которые связаны только с его вершинами. В качестве примера на рис. 4 дано отображение функции трех переменных $y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_3$.

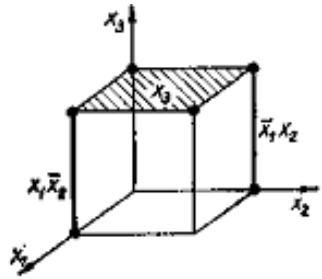


Рис. 4. Покрытие функции $y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_3$ совокупностью s-кубов

Здесь минитермы $\bar{x}_1x_2$ и $x_1\bar{x}_2$ соответствуют 1-кубам (s= 3 — 2 = 1), а минитерм x_3 отображается 2-кубом (s=3-l=2).

Итак, любая дизъюнктивная нормальная форма отображается на n-мерном кубе совокупностью s-кубов, которые покрывают все вершины, соответствующие конституентам единицы (0-кубы). Справедливо и обратное утверждение: если некоторая совокупность s-кубов покрывает множество всех вершин, соответствующих единичным значениям функции, то дизъюнкция соответствующих этим s-кубам минитермов является выражением данной функции в дизъюнктивной нормальной форме. Говорят, что такая совокупность s-кубов (или соответствующих им минитермов) образует *покрытие функции.* Стремление к минимальной форме интуитивно понимается как поиск такого покрытия, число s-кубов которого было бы поменьше, а их размерность s — побольше. Покрытие, соответствующее минимальной форме, называют *минимальным покрытием.* Например, для функции из (5) покрытие на рис. 5, *а* соответствует неминимальной форме $y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_1x_3 \vee x_2x_3$, а покрытия на рис. 5, *б* и *в* — минимальным формам и $y = \bar{x}_1x_2 \vee x_1\bar{x}_2 \vee x_1x_3$.

$y = \bar{x}_1x_2 \vee x_1x_2 \vee x_2x_3$

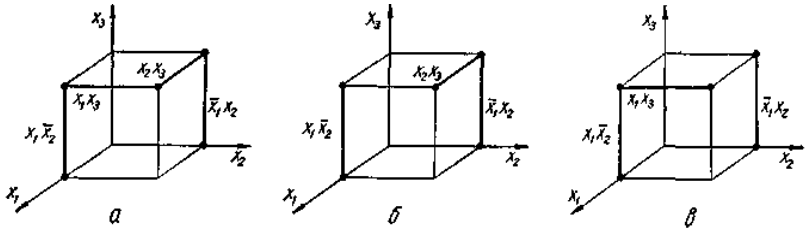


Рис. 5. Покрытия функции
$y = \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3$;

а — неминимальное; *б, в* — минимальные

Отображение функции на *n*-мерном кубе наглядно и просто при $n < 3$. Четырехмерный куб можно изобразить, как показано на рис. 6, где отображены функция четырех переменных и ее минимальное покрытие, соответствующие выражению $y = x_1\bar{x}_3 \vee x_2x_4 \vee \bar{x}_1x_3\bar{x}_4$. Использование этого метода при $n > 4$ требует настолько сложных построений, что теряются все его преимущества.

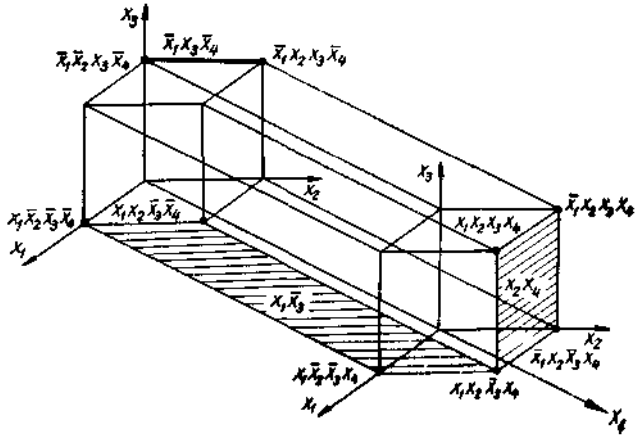


Рис. 6. Отображение функции $y = x_1\bar{x}_3 \vee x_2x_1 \vee \bar{x}_1x_3\bar{x}_4$ на четырехмерном кубе.

7. Карты Карно. В другом методе графического отображения булевых функций используются *карты Карно,* которые представляют собой специально организованные таблицы соответствия. Столбцы и строки таблицы соответствуют всевозможным наборам значений не более двух переменных, причем эти наборы расположены в таком порядке, что каждый последующий отличается от предыдущего значением только одной из переменных. Благодаря этому и соседние клетки таблицы по горизонтали и вертикали отличаются значением только одной переменной. Клетки, расположенные по краям таблицы, также считаются соседними п обладают этим свойством. На рис. 7 показаны карты Карно для двух, трех и четырех переменных.

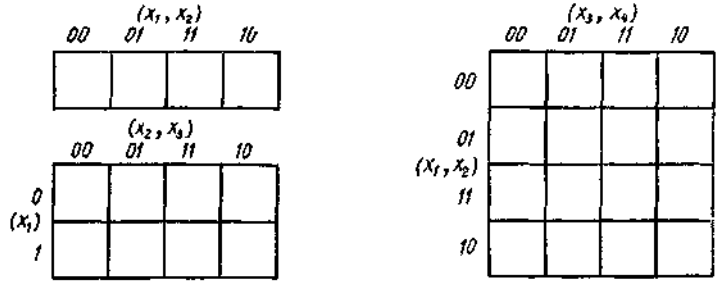


Рис. 7. Карты Карно для двух, трех и четырех переменных

Как и в обычных таблицах соответствия (5.1.3), клетки наборов, на которых функция принимает значение 1, заполняются единицами (нули обычно не вписывают, им соответствуют пустые клетки).

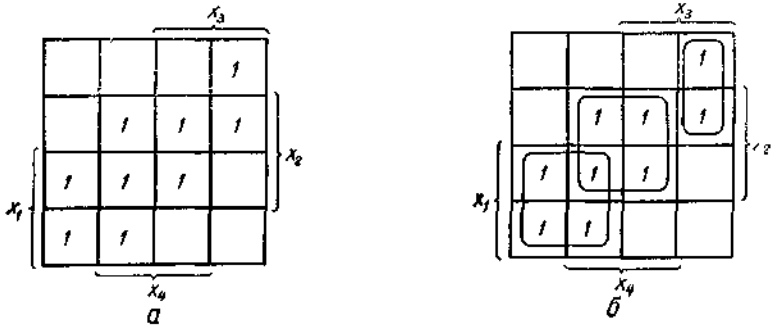


Рис. 8. Отображение на карте Карно функции четырех переменных (*а*) и ее минимального покрытия (б)

Например, на рис. 8, *а* показана карта Карно для функции, отображение которой на четырехмерном кубе дано на рис. 6. Для упрощения строки и столбцы, соответствующие значениям 1 для некоторой переменной, выделяются фигурной скобкой с обозначением этой переменной.

Между отображениями функции на *n*-мерном кубе и на карте Карно имеет место взаимно-однозначное соответствие. На карте Карно *s*-кубу соответствует совокупность 2 соседних клеток, размещенных в строке, столбце, квадрате или прямоугольнике (с учетом соседства противоположных краев карты). Поэтому все положения, изложенные в (6), справедливы и для карт Карно. Так, на рис. 8, *б* показано

покрытие единиц карты, соответствующее минимальной дизъюнктивной форме

$$y = x_1\bar{x}_3 \vee x_2x_4 \vee \bar{x}_1x_3\bar{x}_4$$

рассматриваемой функции.

Считывание минитермов с карты Карно осуществляется по простому правилу. Клетки, образующие *s*-куб, дают минитерм *(n* — s)-го ранга, в который входят те *(n — s)* переменные, которые сохраняют одинаковые значения на этом *s*-кубе, причем значениям 1 соответствуют сами переменные, а значениям 0 — их отрицания. Переменные, которые не сохраняют свои значения на *s*-кубе, в минитерме отсутствуют. Различные способы считывания приводят к различным представлениям функции в дизъюнктивной нормальной форме (рис. 9).

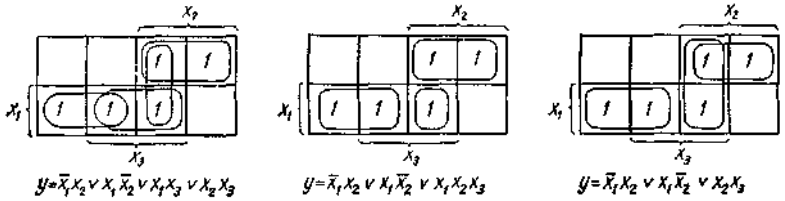


Рис. 8. Способы считывания с карты Карно дизъюнктивной нормальной формы булевой функции.

Использование карт Карно требует более простых построений по сравнению с отображением на *n*-мерном кубе, особенно в случае четырех переменных. Для отображения функций пяти переменных используются две карты Карно на четыре переменные, а для функций шести переменных — четыре таких карты. При дальнейшем увеличении числа переменных карты Карно становятся практически непригодными. Описанные в литературе *карты Вейча* отличаются только другим порядком следования наборов значений переменных и обладают теми же свойствами, что и карты Карно.

8. Комплекс кубов. Несостоятельность графических методов при большом числе переменных компенсируется различными аналитическими методами представления булевых функций. Одним из таких представлений *является комплекс кубов,* использующий терминологию многомерного логического пространства в сочетании со специально разработанной символикой.

Комплекс кубов *К (y)* функции $y = f(x_1, x_2, \ldots, x_n)$ определяется как объединение множеств $K^s(y)$ всех ее *s*-кубов $(s = 0, 1, \ldots, n)$, т. е. $K(y) = \bigcup K^s(y)$, причем некоторые из $K^s(y)$ могут быть пустыми. Для записи *s*-кубов и минитермов функции от *n* переменных

используются слова длины *n,* буквы которых соответствуют всем *n* переменным. Входящие в минитерм переменные называются *связанными* и представляются значениями, при которых минитерм равен единице (1 для x_i и 0 для $\bar{x}_i$). Не входящие в минитерм переменные являются *свободными* и обозначаются через *x.* Например, 2-куб функции пяти переменных, соответствующий минитерму $x_2\bar{x}_3x_5$, запишется как $(x10x1)$, 0-кубы, соответствующие конституентам единицы, представляются наборами значений переменных, на которых функция равна единице. Очевидно, в записи *s*-куба всегда имеется *s* свободных переменных. Если все *n* переменных свободны, что соответствует *n*-кубу, то это означает тождественность единице рассматриваемой функции. Таким образом, для функций, не равных тождественно единице, $K^n(y) = \varnothing$.

Множество всех *s*-кубов *K(y)* записывается как совокупность слов, соответствующих каждому *s*-кубу. Для удобства будем располагать слова *s*-кубов в столбцы, а их совокупность заключать в фигурные скобки. Например, комплекс кубов, соответствующий представлению функции на трехмерном кубе (рис. 10, а), выражается как $K(y) = K^0 \cup K^1 \cup K^2$, где

$$K^0 = \begin{Bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{Bmatrix}; \quad K^1 = \begin{Bmatrix} 0 & 0 & 0 & 0 & x & x \\ 0 & x & x & 1 & 0 & 1 \\ x & 0 & 1 & x & 1 & 0 \end{Bmatrix}; \quad K^2 = \begin{Bmatrix} 0 \\ x \\ x \end{Bmatrix}.$$

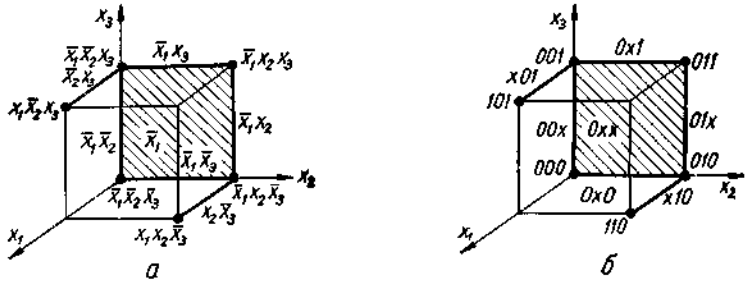


Рис. 10. Комплекс кубов функции трех переменных *(а)* и его символическое представление *(б).*

Для сравнения на рис. 10, *б* изображен комплекс кубов в принятых обозначениях.

Комплекс кубов образует *максимальное покрытие функции.* Исключая из него все те *s*-кубы, которые покрываются кубами высшей размерности, получаем покрытия, соответствующие тупиковым формам. Так, для рассматриваемого примера (рис. 10) имеем тупиковое покрытие

$$C = \begin{Bmatrix} x & x & 0 \\ 0 & 1 & x \\ 1 & 0 & x \end{Bmatrix},$$

которое соответствует функции $y = \bar{x}_2 x_3 \vee x_2 \bar{x}_3 \vee \bar{x}_1$. В данном случае это покрытие является и минимальным.

Для двух булевых функций операция дизъюнкции соответствует объединению их комплексов кубов $K(y_1 \vee y_2) = K(y_1) \cup K(y_2)$, а операция конъюнкции — пересечению комплексов кубов $K(y_1 y_2) = K(y_1) \cap K(y_2)$. Отрицанию функции соответствует дополнение комплекса кубов, т. е. $K(\bar{y}) = \overline{K(y)}$, причем $\overline{K(y)}$ определяется всеми вершинами, на которых функция принимает значение 0.

Таким образом, имеет место взаимно-однозначное соответствие (изоморфизм) между алгеброй булевых функций и алгеброй множеств, представляющих комплексы кубов.

Представление функций в виде комплексов кубов менее наглядно, однако его важнейшие достоинства состоят в том, что снимаются ограничения по числу переменных и облегчается кодирование информации при использовании вычислительных машин.

9. Реализация в различ ных формах. Реализация функции в дизъюнктивной нормальной форме представляет собой логическую схему *И—ИЛИ*. Например, функция $y = \bar{x}_1 x_2 \vee x_1 \bar{x}_2 \vee x_2 x_3$ реализуется логической схемой (рис. 11, *а*).

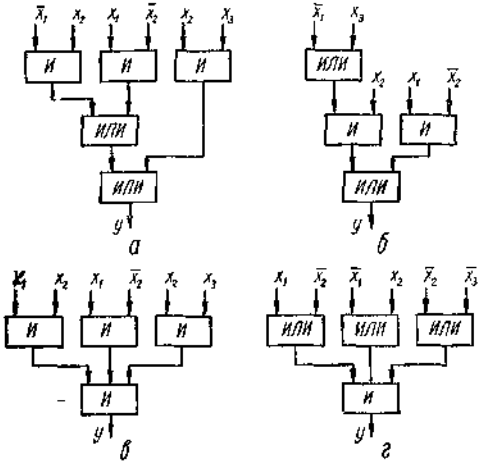


Рис. 11. Реализация функции $y = \bar{x}_1 x_2 \vee x_1 \bar{x}_2 \vee x_2 x_3$:
а — схемой *И-ИЛИ; б* — упрощенной схемой; *в* — двухуровневой схемой *И—ИЛИ; в* — двухуровневой схемой *ИЛИ-И*.

Более экономичная реализация получается, если общий множитель вынести за скобки: $y = x_2(\bar{x}_1 \vee x_3) \vee x_1\bar{x}_2$ (рис. 11, *б*). При использовании элементов со многими входами получаем двухуровневую логическую схему *И—ИЛИ* (рис. 11, *в*).

В соответствии с принципом двойственности (5.2.1), заменяя в дизъюнктивной нормальной форме операции конъюнкции на дизъюнкции, операции дизъюнкции на конъюнкции и беря отрицание каждой переменной, получаем конъюнктиимую нормалышо форму, которая выражает функцию $\bar{y}$, обратную к y. Ее реализация с помощью многовходовых элементов представляет собой двухуровневую логическую схему *ИЛИ—И*. Для рассматриваемой функции $\bar{y} = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2)(\bar{x}_2 \vee \bar{x}_3)$ соответствующая реализация показана на рис. 11, *г*. Если требуется получить схему для данной функции y, то используется инвертор или элемент, реализующий операцию *НЕ—И*.

Конъюнктивную нормальною форму можно получить и другим путем. Для этого используются рассуждения и методы, дуальные рассмотренным по отношению к дизъюнктивным нормальным формам. На многомерном кубе ищется покрытие множества вершин для нулевых значений функции, а на карте Карно — покрытие нулевых клеток. Рассматриваемый пример иллюстрируется на рис. 12, *а* и *б*.

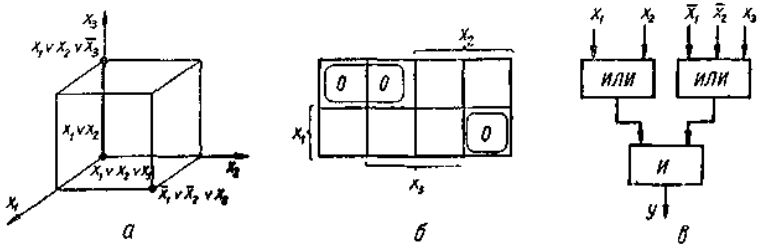


Рис. 12. Считывание конъюнктивной нормальной формы булевой функции с куба (*а*), с карты Карно *(б)* и ее реализация логической схемой *(в)*

Соответствующая конъюнктивная нормальная форма $y = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2 \vee x_3)$ реализуется схемой (рис. 12, *в*). Комплекс кубов этой функции и его дополнение имеют вид:

$$K(y) = \left\{\begin{array}{ccccc|ccccc} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & x \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & x & 1 \\ 0 & 1 & 0 & 1 & 1 & x & x & 1 & 1 \end{array}\right\}; \quad \overline{K(y)} = \left\{\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & x \end{array}\right\},$$

а их покрытия

$$C = \begin{Bmatrix} 0 & 1 & x \\ 1 & 0 & 1 \\ x & x & 1 \end{Bmatrix}; \quad \bar{C} = \begin{Bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & x \end{Bmatrix}.$$

Покрытию $\bar{C}$ соответствует дизъюнктивная нормальная форма для отрицания функций $\bar{y} = x_1x_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2$, откуда можно получить приведенное выше выражение функции в конъюнктивной нормальной форме.

10. Многовыходные схемы. Схемы, реализующие несколько функций, можно представить как простое объединение схем, реализующих каждую функцию отдельно. Но такой путь, как правило, является неэкономичным. Часто бывает целесообразно преобразовать совокупность данных функций к такому виду, чтобы реализующие их схемы содержали общие части, а схема с многими выходами представляла собой единое целое.

Задача сводится к выбору для каждой функции такого покрытия, которое включало бы возможно большее число s-кубов, содержащихся в покрытиях других функций. Этому требованию удовлетворяют, например, покрытия для трех функций (рис. 13).

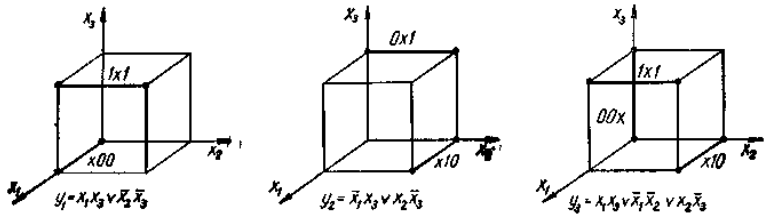


Рис. 13. Покрытия для трех выходных функций.

Соответствующая трехвыходная схема показана на рис.14. Если бы для функции y_3 было выбрано другое покрытие, то схема получилась бы менее экономичной.

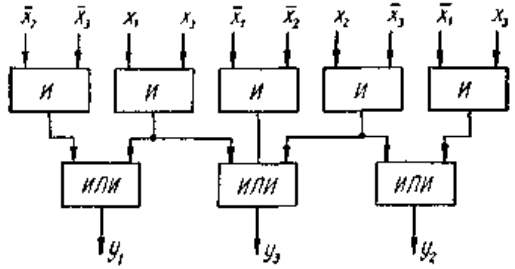


Рис. 14. Логическая схема с тремя выходами

Мы описали различные методы представления булевых функций применительно к задаче минимизации. При небольшом числе переменных эта задача обозрима, и ее можно решить простым перебором различных вариантов. Для функции многих переменных разработаны формальные методы минимизации, которые рассматриваются ниже.

5.5. Минимизация булевых функций

1. Постановка задачи. Минимизация схемы в булевом базисе сводится к поиску минимальной дизъюнктивной формы, которой соответствует минимальное покрытие. Общее число букв, входящих в нормальную форму, выражается *ценой покрытия*

$$c = \sum_s q_s (n - s),$$

где q_s — число s-кубов, образующих покрытие данной функции от n переменных. Используются и другие определения цены покрытия, например, $c' = c + q$, где q — общее число всех кубов покрытия. Во всех случаях минимальное покрытие характеризуется наименьшим значением его цены.

Обычно задача минимизации решается в два шага. Сначала ищут сокращенное покрытие, которое включает все s-кубы максимальной размерности, но не содержит ни одного куба, покрывающегося каким-либо кубом этого покрытия. Соответствующую дизъюнктивную нормальную форму называют *сокращенной,* а ее минитермы— *простыми импликантами.* Для данной функции сокращенное покрытие является единственным, но оно может быть избыточным вследствие того, что некоторые из кубов покрываются совокупностями других кубов.

На втором шаге осуществляется переход от сокращенной к тупиковым дизъюнктивным нормальным формам, из которых выбираются минимальные формы. Тупиковые формы образуются путем исключения из сокращенного покрытия всех избыточных кубов, без которых оставшаяся совокупность кубов еще образует покрытие данной функции, но при дальнейшем исключении любого из кубов она уже не покрывает множества всех вершин, соответствующих единичным значениям функции, т. е. перестает быть покрытием.

Куб сокращенного покрытия, который покрывает вершины данной функции, не покрываемые никакими другими кубами, не может оказаться избыточным и всегда войдет в минимальное покрытие. Такой куб, как и соответствующая ему импликанта, называют

экстремалью (существенной импликантой), а покрываемые им вершины—*отмененными вершинами.* Множество экстремалей образует *ядро покрытия.* Ясно, что при переходе от сокращенного покрытия к минимальному прежде всего следует выделить все экстремали. Если множество экстремалей не образует покрытия, то оно дополняется до покрытия кубами из сокращенного покрытия. Приведенные определения иллюстрируются на рис. 5. п.5.4, где сокращенное покрытие Z (см. рис. 5, *а)* и минимальные покрытия $C'_{\min}$ (рис. 5, *б)* и $C''_{\min}$ (см. рис. 5, *в)* выражаются следующим образом:

$$Z = \begin{Vmatrix} 1 & 1 & x & 0 \\ 0 & x & 1 & 1 \\ x & 1 & 1 & x \end{Vmatrix}; \quad C'_{\min} = \begin{Vmatrix} 1 & x & 0 \\ 0 & 1 & 1 \\ x & 1 & x \end{Vmatrix}; \quad C''_{\min} = \begin{Vmatrix} 1 & 1 & 0 \\ 0 & x & 1 \\ x & 1 & x \end{Vmatrix}.$$

Сокращенная форма представляет собой дизъюнкцию четырех простых кмпликпнт, т. е. $y = x_1\bar{x}_2 \vee x_1x_3 \vee x_2x_3 \vee \bar{x}_1x_2$. Экстремалями являются простые импликанты $x_1\bar{x}_2$ и $\bar{x}_1x_2$, которым соответствуют 1-кубы (10*x*) и (01*x*), а отмеченные вершины — $x_1\bar{x}_2\bar{x}_3$ и $\bar{x}_1x_2\bar{x}_3$ или соответственно (100) и (010).

2. Метод Квайна — Мак-Класки. Этот метод используется в случаях, когда функция задана в дизъюнктивной совершенной нормальной форме (или таблицей соответствия). Приведение к сокращенной форме осуществляется последовательным применением операции склеивания $ax_i \vee a\bar{x}_i = a$, где a — конъюнкции переменных, отличных от x_i. Множеству конституент единицы, представленных в исходной форме, соответствует совокупность 0-кубов K^0, а операции склеивания — объединение двух 0-кубов, которые отличаются только одной координатой. Результатом такого объединения является 1-куб, в котором различные координаты исходных 0-кубов замещены символом *x*. Сравнивая попарно все 0-кубы, получаем множество 1-кубов K^1. Применяя к K^1 операцию склеивания, находим множество 2-кубов K^2 и т. д. Этот процесс продолжается до тех пор, пока получаемое из K^s очередное K^{s+1} не окажется пустым множеством. В результате множество K^0 преобразуется в комплекс кубов $K = \{K^0, K^1, K^2, \ldots, K^s\}$, причем $s \leqslant n$.

Для выделения из K множества простых импликант $Z \subset K$ при каждой операции склеивания необходимо отмечать каким-либо знаком (например, меткой $\vee$) те кубы, которые объединяются в кубы высшей размерности. Очевидно, неотмеченные кубы и образуют множество простых импликант Z. Чтобы уменьшить число сравниваемых пар при операции объединения целесообразно разбить

множество K^s на классы, в каждом из которых содержатся s-кубы с одинаковым числом единиц (или нулей), и упорядочить эти классы по возрастающему числу единиц. Так как объединяться могут только такие два s-куба, число единиц в которых точно на одну больше или меньше, то достаточно ограничиться попарным сравнением s-кубов одного класса с s-кубами соседнего класса.

На втором шаге при извлечении экстремалей и образовании минимального покрытия используйся таблица покрытий. Ее строки соответствуют простым импликантам, а столбцы — конституентам единицы дизъюнктивной совершенной нормальной формы данной функции, которые представляются 0-кубами (вершинами) комплекса кубов. В клетку таблицы записывается метка, если простая импликанта в данной строке покрывает вершину в данном столбце. Экстремалям соответствуют те строки таблицы, которые содержат единственную метку в каком-либо столбце. Удаляя строки экстремалей и все столбцы, в которых эти строки имеют метки, получаем более простую таблицу. На основе этой таблицы выбираем простые импликанты, которые дополняют выделенное множество экстремалей до минимального покрытия функции.

3. Пример минимизации функции. Рассмотрим в качестве примера функцию четырех переменных $y = f(x_1, x_2, x_3, x_4)$, заданную таблицей соответствия

x_1	0000	0000	1111	1111
x_2	0000	1111	0000	1111
x_3	0011	0011	0011	0011
x_4	0101	0101	0101	0101
y	0001	1101	0101	1100

Ей соответствует дизъюнктивная совершенная нормальная форма

$$y = \bar{x}_1\bar{x}_2x_3x_4 \vee \bar{x}_1x_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee \bar{x}_1x_2x_3x_4 \vee x_1\bar{x}_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3x_4 \vee$$
$$\vee x_1x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3x_4.$$

Множество 0-кубов после разбиения и упорядочения записывается следующим образом:

$$K^0 = \left\{\begin{array}{c|cc|ccc} & \vee & \vee & \vee & \vee & \vee \\ \hline 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right\}.$$

Объединяя кубы и отмечая те из них, которые покрываются кубами большей размерности, имеем:

$$K^1 = \begin{Bmatrix} \overset{\vee}{0} & \overset{\vee}{x} & 0 & 0 & x & 1 & \overset{\vee}{x} & 1 & \overset{\vee}{1} \\ 1 & 1 & x & 1 & 0 & 0 & 1 & x & 1 \\ 0 & 0 & 1 & x & 1 & x & 0 & 0 & 0 \\ x & 0 & 1 & 1 & 1 & 1 & 1 & 1 & x \end{Bmatrix}; \quad K^2 = \begin{Bmatrix} x \\ 1 \\ 0 \\ x \end{Bmatrix}.$$

Простым импликантам соответствуют неотмеченные кубы. Составляем таблицу покрытия Z, которому соответствует сокращенная форма $y = \bar{x}_1 x_3 x_4 \vee \bar{x}_1 x_2 x_4 \vee \bar{x}_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_3 x_4 \vee x_2 x_3$:

K^0 / Z	0100	0011	0101	1001	1100	0111	1011	1101	Обозначения импликант
0 x 1 1		∨				∨			*A*
0 1 x 1			∨			∨			*B*
x 0 1 1		∨					∨		*C*
1 0 x 1				∨			∨		*D*
1 x 0 1				∨				∨	*E*
x 1 0 x	∨		∨		∨			∨	*F*

Извлекаем единственную экстремаль *(x*10*x),* которой соответствует минитерм $x_2\bar{x}_3$, и упрощаем таблицу к виду:

K^0_1 / Z_t	0011	1001	0111	1011
0 x 1 1	∨		∨	
0 1 x 1			∨	
x 0 1 1	∨			∨
1 0 x 1		∨		∨
1 x 0 1		∨		

В качестве дополнительных целесообразно выбрать кубы (0*x*11) и (10*x*1), так как они совместно с экстремалью *(x*10*x)* образуют покрытие функции, минимальная форма которой имеет вид:

$$y = \bar{x}_1 x_3 x_4 \vee x_1 \bar{x}_2 x_4 \vee x_2 \bar{x}_3.$$

Соответствующее этой функции мини-мальное покрытие иллюстрируется на четырехмерном кубе и на карте Карно (рис. 1).

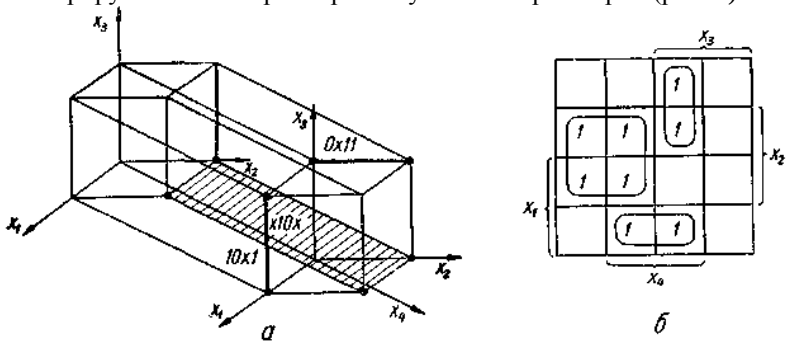


Рис. 1. Минимальное покрытие функции на четырехмерном кубе *(а)* и карте Карно (*б*).

4. Алгебраический метод. Выбор минимального покрытия на заключительном этапе формализуется с помощью *алгебраического метода,* предложенного С. Петриком. Простые импликанты обозначаются какими-либо символами (обычно для этой цели используются прописные буквы латинского алфавита), и по столбцам таблицы покрытий записываются дизъюнкции тех имплнкант, которые отмечены в данном столбце. Смысл этой записи вытекает из того, что любая из отмеченных импликант покрывает данную вершину. Покрытию функции соответствует конъюнкция всех записанных дизъюнкций.

Раскрывая скобки и упрощая выражения на основе тождеств булевой алгебры (упрощать можно и до раскрытия скобок), переходим к дизъюнктивной форме, каждый член которой представляет собой конъюнкцию простых импликант и соответствует некоторому тупиковому покрытию рассматриваемой функции. Сравнивая все тупиковые покрытия и отбирая те из них, которые характеризуются минимальной ценой, приходим к одному или нескольким минимальным покрытиям.

Так, для примера из (3) имеем: $F(A \vee C)(B \vee F)(D \vee E)F(A \vee B) \wedge$

$\wedge (C \vee D)(E \vee F) = F(A \vee C)(A \vee B)(D \vee E)(C \vee D) = F(A \vee AB \vee$

$\vee AC \vee BC)(CD \vee CE \vee D \vee DE) = F(A \vee BC)(D \vee CE) = ADF \vee$

$\vee ACEF \vee BCDF \vee BCEF$. Итак, получаем четыре тупиковых покрытия

$$C_1 = \begin{Vmatrix} 0 & 1 & x \\ x & 0 & 1 \\ 1 & x & 0 \\ 1 & 1 & x \end{Vmatrix}; \quad C_2 = \begin{Vmatrix} 0 & x & 1 & x \\ x & 0 & x & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & x \end{Vmatrix};$$

$$C_3 = \begin{Vmatrix} 0 & x & 1 & x \\ 1 & 0 & 0 & 1 \\ x & 1 & x & 0 \\ 1 & 1 & 1 & x \end{Vmatrix}; \quad C_4 = \begin{Vmatrix} 0 & x & 1 & x \\ 1 & 0 & x & 1 \\ x & 1 & 0 & 0 \\ 1 & 1 & 1 & x \end{Vmatrix},$$

цены которых $c_1 = 2(4 — 1) + 1\ (4 — 2) = 8$ и $c_2 = c_3 = c_4 = 3(4—1) + 1(4—2) = 11$, т. е. $C_{\min} = C_1$.

Алгебраические преобразования упрощаются, если исходить из таблицы покрытий, получаемой после извлечения экстремалей. Тогда результатом таких преобразований являются множества простых импликант, дополняющих совокупность экстремалей до тупиковых покрытий. Сравнивая эти множества по их цене, выбираем минимальные дополнения, которые совместно с множеством экстремалей образуют минимальные покрытия.

5. Метод Блейка—Порецкого. При минимизации функции методом Квайна—Мак-Класкп требуется предварительно представить ее в совершенной дизъюнктивной нормальной форме, что часто связано с дополнительными преобразованиями.

Если исходить из произвольной дизъюнктивном нормальной формы, то для получения промежуточной сокращенной формы можно воспользоваться прямым *методом Блейка—Порецкого*. Он основан на тождестве

$$ac \vee b\bar{c} = ac \vee b\bar{c} \vee ab,$$

называемом *операцией обобщенного склеивания.* Действительно, $ac \vee b\bar{c} = ac \vee abc \vee b\bar{c} \vee ab\bar{c} = ac \vee b\bar{c} \vee ab(c \vee \bar{c}) = ac \vee b\bar{c} \vee ab$. Разумеется, входящие в это тождество буквы могут представлять любые булевы формулы и, в частности, конъюнкции переменных.

Можно показать, что произвольная дизъюнктивная нормальная форма приводится к сокращенной применением всех возможных обобщенных склеиваний с последующим устранением минитермов на основе операции поглощения $a \vee ab = a$. При этом возможны следующие случаи.

1) Конъюнкция a содержит переменную x_i, а конъюнкция b — отрицание той же переменной $\bar{x}_i$ (или наоборот). Тогда $ab = 0$ и в

результате операции обобщенного склеивания не получаются новые минитермы. Таким образом, следует подвергать этой операции только те пары минитермов, в которых единственная переменная представлена как x_j и $\bar{x}_j$.

2) Конъюнкция a содержит только те переменные, которые входят в конъюнкцию b (или наоборот), т. е. $b = ac$. Тогда $ax_i \vee \vee b\bar{x}_i = ax_i \vee ac\bar{x}_i = ax_i \vee ac\bar{x}_i \vee ac = ax_i \vee ac = ax_i \vee b$, т. е. минитерм исходной дизъюнктивной нормальной формы поглощается минитермом, образованным в результате обобщенного склеивания. Пусть, например, функция из (3) задана некоторым покрытием, которое соответствует дизъюнктивной нормальной форме: $y = x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_4 \vee \bar{x}_1x_2x_4 \vee \bar{x}_1\bar{x}_2x_3x_4$. Применяя операцию обобщенного склеивания к парам $(x_2\bar{x}_3\bar{x}_4,\ \bar{x}_1x_2x_4)$; $(x_1x_2\bar{x}_3,\ x_1\bar{x}_2x_4)$; $(x_1x_2\bar{x}_3,\ \bar{x}_1x_2x_4)$; $(x_1\bar{x}_2x_4,\ \bar{x}_1x_2x_4)$; $(\bar{x}_1x_2x_4,\ \bar{x}_1\bar{x}_2x_3x_4)$ и учитывая, что в двух последних парах происходит поглощение минитермов, получаем: $y = (x_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2x_4 \vee \bar{x}_1x_2\bar{x}_3) \vee (x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_4 \vee \vee (\bar{x}_1x_2x_4 \vee \bar{x}_1x_3x_4)$. Удаляя одинаковые члены $(a \vee a = a)$ и группируя старые и новые минитермы, имеем: $y = (x_2\bar{x}_3\bar{x}_4 \vee \vee \bar{x}_1x_2x_4 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_4) \vee (\bar{x}_1x_2\bar{x}_3 \vee x_1\bar{x}_3x_4 \vee x_2\bar{x}_3x_4 \vee \bar{x}_2x_3x_4 \vee \vee \bar{x}_1x_3x_4)$. Очевидно, при дальнейшем обобщенном склеивании имеет смысл рассматривать только пары, образованные новыми минитермами со всеми минитермами полученной дизъюнктивной нормальной формы. Такими парами являются: $(x_2\bar{x}_3\bar{x}_4,\ x_1\bar{x}_3x_4)$; $(x_2\bar{x}_3\bar{x}_4,\ x_2\bar{x}_3x_4)$; $(\bar{x}_1x_2x_4,\ x_1\bar{x}_3x_4)$; $(\bar{x}_1x_2x_4,\ \bar{x}_2x_3x_4)$; $(x_1x_2\bar{x}_3,\ \bar{x}_1x_2\bar{x}_3)$; $(x_1\bar{x}_2x_4,\ x_2\bar{x}_1x_4)$; $(x_1\bar{x}_2x_4,\ \bar{x}_1x_3x_4)$; $(\bar{x}_1x_2\bar{x}_3,\ x_1\bar{x}_3x_4)$; $(\bar{x}_1x_2\bar{x}_3,\ \bar{x}_1x_3x_4)$; $(x_1\bar{x}_3x_4,\ \bar{x}_2x_3x_4)$; $(x_2\bar{x}_3x_4,\ \bar{x}_1x_3x_4)$. Применяя к каждой паре операции обобщенного склеивания и поглощения в соответствии с приведенными выше правилами, находим: $y = \bar{x}_1x_2x_4 \vee \vee x_1\bar{x}_2x_4 \vee x_1\bar{x}_3x_4 \vee \bar{x}_2x_3x_4 \vee \bar{x}_1x_3x_4 \vee x_2\bar{x}_3$. Единственный новый минитерм $x_2\bar{x}_3$ в паре с любым из остальных минитермов не приводит к появлению новых минитермсв. Поэтому полученная форма является сокращенной. Она, как и должно быть, совпадает с найденной в (3).

6. Склеивание и поглощение кубов. Геометрически операции сообщенного склеивания и поглощения соответствуют некоторым операциям над кубами, имеющими противоположные грани. В результате получается новый куб, который либо располагается между исходными кубами, либо поглощает один из кубов или оба куба.

Преобразования, выполненные в (5) иллюстрируются на рис. 2. Исходной дизъюнктивной нормальной форме соответствует некоторое

покрытие (рис. 2, *а),* которое преобразуется к промежуточному покрытию (рис. 2, *б).*

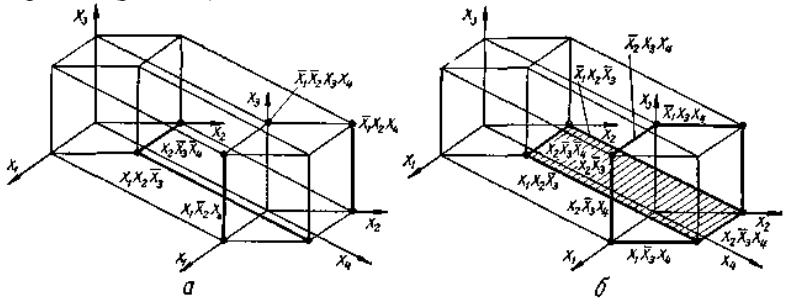


Рис. 2. Покрытие функции $y = x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_1 \vee x_1\bar{x}_2x_4 \vee \bar{x}_1x_2x_1 \vee \bar{x}_1\bar{x}_2x_3x_4$;
а — исходное; *б* — промежуточное

Сокращенной нормальной форме соответствует покрытье, получаемое из рис 2, *б* поглощением кубов $x_2\bar{x}_3\bar{x}_4$, $\bar{x}_1x_2\bar{x}_3$, $x_2\bar{x}_3x_4$ и $x_1x_2\bar{x}_4$ кубом $x_2\bar{x}_3$.

Операции над кубами удобно выполнять в символической форме. Сравнивая в исходном покрытии C_0 попарно кубы, имеющие противоположные значения 0 и 1 только для одной координаты, образуем множество новых кубов C^*_0. Координаты этих кубов можно определить с помощью *операции покоординатного произведения (*),* задаваемой таблицей:

*	0	1	x
0	0	x	0
1	x	1	1
x	0	1	x

Так, для рассматриваемою примера имеем:

$$C_0 = \begin{Vmatrix} x & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & x & x & 1 \\ 0 & x & 1 & 1 & 1 \end{Vmatrix}; \quad C_0^* = \begin{Vmatrix} 0 & 1 & x & x & 0 \\ 1 & x & 1 & 0 & x \\ 0 & 0 & 0 & 1 & 1 \\ x & 1 & 1 & 1 & 1 \end{Vmatrix},$$

где кубы множества C^*_0 получены в результате операции покоординатного произведения над следующими парами кубов из C_0:

$$\begin{Vmatrix} x & 0 \\ 1 & 1 \\ 0 & x \\ 0 & 1 \end{Vmatrix}; \quad \begin{Vmatrix} 1 & 1 \\ 1 & 0 \\ 0 & x \\ x & 1 \end{Vmatrix}; \quad \begin{Vmatrix} 1 & 0 \\ 1 & 1 \\ 0 & x \\ x & 1 \end{Vmatrix}; \quad \begin{Vmatrix} 1 & 0 \\ 0 & 0 \\ x & 1 \\ 1 & 1 \end{Vmatrix}; \quad \begin{Vmatrix} 0 & 0 \\ 1 & 0 \\ x & 1 \\ 1 & 1 \end{Vmatrix}.$$

Объединяя множества C_0 и C^*_0, выполняем операции поглощения в соответствии с тождествами $a \vee a = a$ и $a \vee ab = a$. Это

соответствует удалению из множества $C_0 \cup C_0^*$ повторяющихся кубов, а также тех кубов, которые покрываются другими кубами (куб покрывает все кубы низшей размерности, если отличные от x координаты покрывающего куба совпадают с соответствующими координатами покрываемых кубов). В нашем примере повторяющихся кубов нет, а куб (0011) поглощается кубом (x011) или (0x11). В результате получаем промежуточное покрытие

$$C_1 = \left(\begin{array}{cccc|ccccc} x & 1 & 1 & 0 & 0 & 1 & x & x & 0 \\ 1 & 1 & 0 & 1 & 1 & x & 1 & 0 & x \\ 0 & 0 & x & x & 0 & 0 & 0 & 1 & 1 \\ 0 & x & 1 & 1 & x & 1 & 1 & 1 & 1 \end{array}\right),$$

где исходные и новые кубы разделены пунктирной линией. Дальше операция обобщенного склеивания выполняется над покрытием C_1 покоординатным произведением кубов, расположенных справа от разделяющей линии, с каждым кубом из C_1, который подлежит склеиванию. Получаем множество новых кубов

$$C_1^* = \left(\begin{array}{ccccccccccc} 1 & x & x & 0 & x & 1 & x & x & 0 & 1 & 0 \\ 1 & 1 & 1 & x & 1 & x & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & x & x & x \\ x & x & 1 & 1 & x & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

После операции поглощения в множестве $C_1 \cup C_1^*$ имеем следующее преобразованное покрытие:

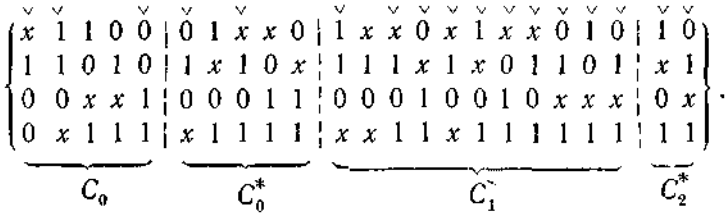
$$C_2 = \left(\begin{array}{cc|ccccc} 1 & 0 & 1 & x & 0 & x \\ 0 & 1 & x & 0 & x & 1 \\ x & x & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & x \end{array}\right).$$

Продолжая склеивание кубов последней группы (она содержит единственный 2-куб) со всеми кубами из C_2, получаем множество

$$C_2^* = \left(\begin{array}{cc} 1 & 0 \\ x & 1 \\ 0 & x \\ 1 & 1 \end{array}\right),$$

объединяя которое с C_2 операциями поглощения, приходим снова к C_2, так как C^*_2 не содержит новых кубов. Отсюда следует, что покрытие C_2 соответствует сокращенной дизъюнктивной нормальной форме данной функции.

Ниже приведена более рациональная запись преобразования произвольной дизъюнктивной нормальной формы к сокращенной форме:



На каждом этапе над поглощаемыми кубами ставятся метки $\vee$ (или соответствующие столбцы вычеркиваются). По окончании преобразования сокращенное покрытие определяется совокупностью неотмеченных кубов.

7. Частично определенные функции. В практике нередко приходится иметь дело с такими функциями, которые определены не на всех наборах значений переменных. Подобные случаи встречаются, когда по условиям функционирования некоторые из наборов не используются и поэтому безразлично, какие значения принимает функция на этих наборах Это обстоятельство можно использовать при минимизации функции, доопределив ее на *безразличных наборах* так, чтобы обеспечить наиболее экономичнаю реализацию.

Пусть дана частично определенная функция $y = f(x_1, x_2, \ldots, x_n)$. Обозначим через $y^1 = f^1(x_1, x_2, \ldots, x_n)$ функцию, которая доопределена на всех безразличных наборах единицами, а через $y^0 = f^0(x_1, x_2, \ldots, x_n)$ — нулями. Задача оптимального доопределения данной функции сводится к выбору из сокращенного покрытия для функции y^1 минимального количества кубов максимальной размерности, совокупность которых покрывала бы все вершины функции y^0. Такая совокупность кубов и образует минимальное покрытие частично определенной функции y. При этом оно может покрывать и некоторые вершины, соответствующие безразличным наборам, что означает доопределение функции на этих наборах единичными значениями.

8. Преобразователь кодов. Примером частично определенных функций может служить таблица соответствия преобразования кода прямого замещения в двоично-десятичный код 2421:

Десятичное число		0 1 2 3 4 5 6 7 8 9	Избыточные наборы
Код прямого замещения	x_1	0 0 0 0 0 0 0 0 1 1	1 1 1 1 1 1
	x_2	0 0 0 0 1 1 1 1 0 0	0 0 1 1 1 1
	x_3	0 0 1 1 0 0 1 1 0 0	1 1 0 0 1 1
	x_4	0 1 0 1 0 1 0 1 0 1	0 1 0 1 0 1
Двоично десятичный код 2421	y_1	0 0 0 0 0 1 1 1 1 1	Функции не определены
	y_2	0 0 0 0 1 0 1 1 1 1	
	y_3	0 0 1 1 0 1 0 0 1 1	
	y_4	0 1 0 1 0 1 0 1 0 1	

Код прямого замещения предаваляет собой обычное представление одноразрядного десятичного числа в двоичной системе счисления, т. е. $x_1 \cdot 2^3 + x_2 \cdot 2^2 + x_3 \cdot 2^1 + x_4 \cdot 2^0 = 8x_1 + 4x_2 + 2x_3 + x_4$. Код 2421 соответствует представлению числа в виде $y_1 \cdot 2^1 + y_2 \cdot 2^2 + y_3 \cdot 2^1 + y_4 \cdot 2^0 = 2y_1 + 4y_2 + 2y_3 + y_4$. Таким образом, преобразователь кодов представляет собой схему с четырьмя входами и четырьмя выходами.

Проиллюстрируем минимизацию схемы на картах Карно (рис. 3) с учетом положений о многовыходных схемах, изложенных в (5.4.10).

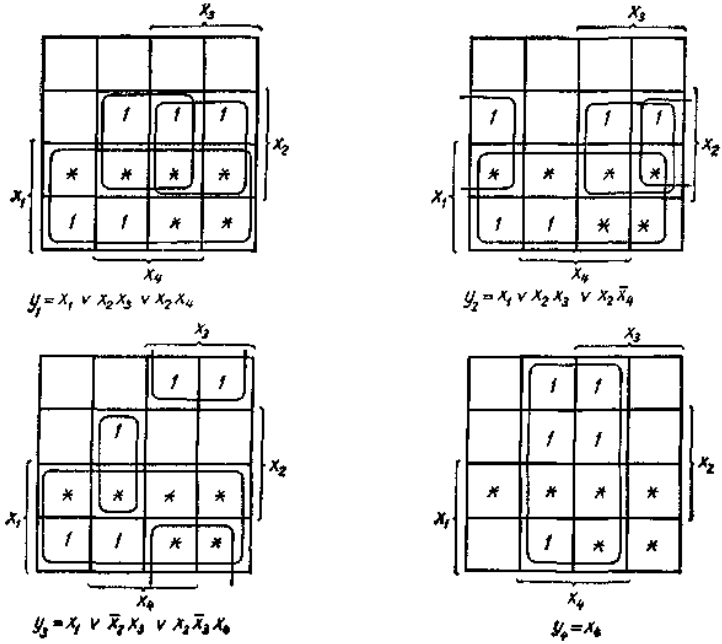


Рис. 3. Минимальные покрытия выходных функций преобразователя кодов.

Используя избыточные наборы, которые отмечены на карте звездочками, образуем минимальные покрытия для каждой из четырех функций которые включали бы возможно больше однотипных кубов.

Соответствующая логическая схема показана на рис. 4.

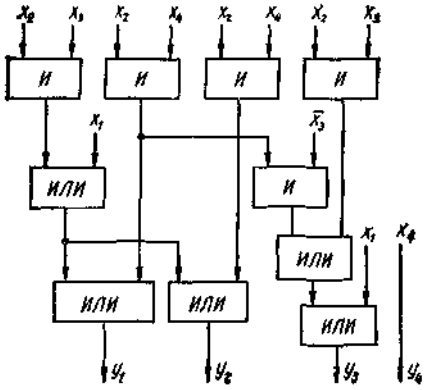


Рис. 4. Логическая схема преобразователя кодов.

9. Сумматор. Другим примером логической схемы, который дает повод использовать частично определенные функции, является одноразрядный сумматор, выполняющий арифметическое сложение двоичных чисел x_k и y_k k-го разряда и переноса из младшего разряда p_{k-1}. В результате должна получаться сумма s_k и перенос в старший разряд p_k. Таблица соответствия такого сумматора имеет вид:

x_k	0	0	0	0	1	1	1	1
y_k	0	0	1	1	0	0	1	1
p_{k-1}	0	1	0	1	0	1	0	1
s_k	0	1	1	0	1	0	0	1
p_k	0	0	0	1	0	1	1	1

Отображение функций s_k и p_k на трехмерных кубах показана на рис. 5.

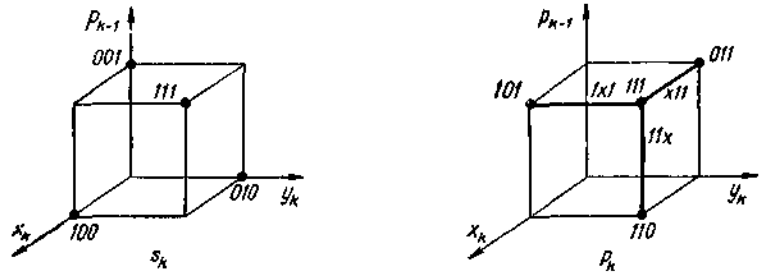


Рис. 5. Отображение выходных функций сумматора на трехмерных кубах

Их дизъюнктивные нормальные формы имеют вид:

$s_k = \bar{x}_k\bar{y}_k p_{k-1} \vee \bar{x}_k y_k \bar{p}_{k-1} \vee x_k\bar{y}_k\bar{p}_{k-1} \vee x_k y_k p_{k-1}$ и $p_k = x_k p_{k-1} \vee$ $\vee x_k y_k \vee y_k p_{k-1}$, соответствующие минимальным покрытиям. Как видно, выражение для s_k не поддается минимизации изложенными ранее методами. Единственная возможность — это использовать вынесение за скобки: $s_k = (x_k\bar{y}_k \vee \bar{x}_k y_k)\,\bar{p}_{k-1} \vee (x_k y_k \vee \bar{x}_k\bar{y}_k)\,p_{k-1}$.

В подобных случаях для минимизации применяется прием, основанный на использовании более простой реализации функции $p_k = f(x_k,\ y_k,\ p_{k-1})$ в качестве составной части другой функции s_k. При этом p_k рассматривается как переменная, т. е. $s_k = \varphi(x_k,$ $y_k,\ p_{k-1},\ p_k)$. Но таблица соответствия для s_k теперь содержит избыточные наборы переменных, которые отмечены звездочками:

x_k	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
y_k	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
p_{k-1}	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
p_k	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
s_k	0	1	*	1	*	*	0	1	*	*	0	*	0	*	1	

Используем для минимизации полученной частично определенной функции s_k карту Карно (рис. 6).

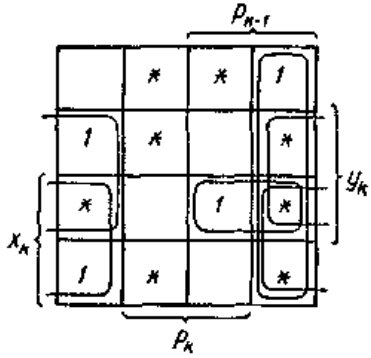


Рис. 6. Минимизация функции s_k сумматора на карте Карно.

Минимальному покрытию соответствует выражение $s_k = x_k\bar{p}_k \vee y_k\bar{p}_k \vee p_{k-1}\bar{p}_k \vee x_k y_k p_{k-1}$. После вынесения за скобки получаем подготовленные к реализации выражения: $s_k = (x_k \vee y_k \vee p_{k-1})\,\bar{p}_k \vee x_k y_k p_{k-1}$; $p_k = x_k y_k \vee (x_k \vee y_k)\,p_{k-1}$. Соответствующая схема показана на рис. 7.

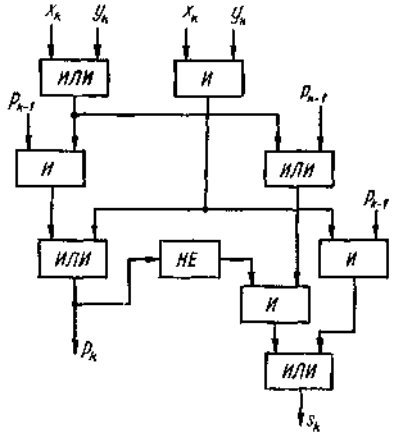


Рис. 7. Логическая схема сумматора.

10. Минимизация в других системах. В реальных условиях проектирование логических схем основывается на использовании некоторого конкретного набора элементов. Обычно стремятся стандартизировать такие элементы с тем, чтобы при одинаковой конструкции они позволяли в зависимости от способа включения реализовать различные логические функции.

Например, комплект интегральных схем может включать многовходовые транзисторные вентили *НЕ—ИЛИ* и *НЕ—И,* а также полусумматоры, реализующие сумму по модулю 2 (неравнозначность). С помощью схем на ферритах обычно реализуются отрицание, дизъюнкция, конъюнкция, штрих Шеффера и стрелка Пирса. Один из пневмисторных модулей, наряду с этими функциями, позволяет реализовать также импликацию и отрицание импликации. В связи с этим перед разработчиком возникаем задача представления и минимизации функции в различных функционально полных системах элементов Известны методы получения канонических форм для логических функций в любом базисе на основе табличного задания или преобразования другого базиса. Что же касается проблемы минимизации в общем виде, то она остается пока решенной не полностью. Обычно применяются частные методы минимизации, аналогичные разработанным для булевого базиса. Часто минимальное представление в булевом базисе используется как исходное и при реализации в других базисах, соответствующее выражение функции в которых получается на основе тождественных преобразований.

5.6. Многозначная логика

1. Функции многозначной логики. Естественным обобщением двузначной логики являйся *k-значная логика* при $k > 2$. Она рассматривает однородные логические функции, определяемые на множестве $\{0, 1, \ldots, k-1\}$, состоящем из k элементов. В силу однородности сама функция k-значной логики от n переменных $f(x_1, x_2, \ldots, x_n)$ принимает значения из того же конечного множества. Подобно функциям двузначной логики, k-значные логические функции можно задать в виде конечной таблицы. Как уже указывалось в (5.1. 3), количество столбцов в такой таблице равно k^n, а количество всевозможных функций выражается числом $k^{(k)^n}$. Это число сильно возрастает с ростом k даже для небольших значений n. Так, при $k = 10$ будем иметь 10^{10} функций одной переменной и 10^{100} функций двух переменных. Поэтому нечего и думать о том, чтобы изучить свойства таких функций путем их перебора, как это делается в двузначной логике.

Обобщение двузначной логики на k-значный случай еще не полностью завершено. Обычно ограничиваются рассмотрением наиболее важных многозначных функций одной и двух переменных. По аналогии с двузначными функциями вводятся понятия равенства и суперпозиции функций k-значной логики. Проблема полноты для многозначной логики также еще далека от полного решения. Полученные результаты в основном сводятся либо к общим условиям существования полных систем функций, либо к рассмотрению конкретных базисов и полных систем, которые по тем или иным соображениям считаются удобными для представления функций многозначной логики и практического применения для синтеза логических схем.

2. Константы и функции одной переменной. В k-значной логике имеется k *констант* $f_0 = 0, f_1 = 1, \ldots, f_{k-1} = k - 1$. Среди функций одной переменной $f(x)$ наиболее употребительны следующие:

1) *характеристические функции* i-го порядка, число которых равно k $(i = 0, 1, \ldots, k-1)$

$$\varphi_i(x) = \begin{cases} k-1 & \text{при } x = i, \\ 0 & \text{при } x \neq i; \end{cases}$$

2) *функция инверсии* $\bar{x} = k - 1 - x$;

3) *функция циклического отрицания (цикл)* $\hat{x} = x + 1 \pmod k$. Ниже приведена таблица этих функций при $k = 5$:

$i(x)$ \ x	0	1	2	3	4
$\varphi_0(x)$	4	0	0	0	0
$\varphi_1(x)$	0	4	0	0	0
$\varphi_2(x)$	0	0	4	0	0
$\varphi_3(x)$	0	0	0	4	0
$\varphi_4(x)$	0	0	0	0	4
$\bar{x}$	4	3	2	1	0
$\hat{x}$	1	2	3	4	0

Очевидно, инверсия характеристических функций выражается соотношением:

$$\overline{\varphi_i(x)} = \begin{cases} 0 & \text{при } x = i; \\ k-1 & \text{при } x \neq i. \end{cases}$$

В многозначной логике используются также функции одной переменной более общего вида:

$$e_{ij}(x) = \begin{cases} j & \text{при } x = i; \\ 0 & \text{при } x \neq i, \end{cases}$$

частным случаем которых при $j = k-1$ являются характеристические функции, т. е. $\varphi_i(x) = e_{i,\,k-1}(x)$. При j=1 получаем другой тип характеристических функций

$$\psi_i(x) = \begin{cases} 1 & \text{при } x = i; \\ 0 & \text{при } x \neq i. \end{cases}$$

В двузначном случае $\varphi_0(x)$, $\bar{x}$ и $\hat{x}$ совпадают с отрицанием, а

$\varphi_1(x) = x$.

3. Функции двух переменных. Наиболее важное значение имеют следующие функции двух переменных:

1) *k-значная дизъюнкция* $x_1 \vee x_2 = \max(x_1, x_2)$;
2) *k-значная конъюнкция* $x_1 \wedge x_2 = \min(x_1, x_2)$;
3) *k-значная функция Шеффера—Вебба* $x_1/x_2 = x_1 \vee x_2 + 1 \pmod k$;
4) *функция сложения по модулю* k $x_1 + x_2 \pmod k$;
5) *функция умножения по модулю* k $x_1 x_2 \pmod k$.

Значения этих функций при $k = 4$ приведены ниже

x_1	0 0 0 0	1 1 1 1	2 2 2 2	3 3 3 3
x_2	0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
$x_1 \vee x_2$	0 1 2 3	1 1 2 3	2 2 2 3	3 3 3 3
$x_1 \wedge x_2$	0 0 0 0	0 1 1 1	0 1 2 2	0 1 2 3
x_1/x_2	1 2 3 0	2 2 3 0	3 3 3 0	0 0 0 0
$x_1 + x_2 \pmod k$	0 1 2 3	1 2 3 0	2 3 0 1	3 0 1 2
$x_1 x_2 \pmod k$	0 0 0 0	0 1 2 3	0 2 0 2	0 3 2 1

При k=2 функции $x_1 \vee x_2$ и $x_1 \wedge x_2$ совпадают с соответствующими функциями двузначной логики и между ними имеют место аналогичные зависимости: $x_1 \vee x_2 = \overline{\bar{x}_1 \wedge \bar{x}_2}$; $x_1 \wedge x_2 = \overline{\bar{x}_1 \vee \bar{x}_1}$.

Как и в двузначной логике, k -значные дизъюнкция и конъюнкция подчиняются ассоциативному, коммутативному и обоим дистрибутивным законам. Поэтому вместе с инверсией эти операции превращают множество $\{0, 1, \ldots, k-1\}$ в булеву алгебру (5. 2. 10).

4. Нормальные формы. Воспользовавшись понятием характеристических функций для двузначного случая (k=2), можно представить совершенные (дизъюнктивную и конъюнктивную) нормальные формы булевой функции выражениями

$$f(x_1, \ldots, x_n) = \bigvee f(\alpha_1, \ldots, \alpha_n) \wedge \varphi_{\alpha_1}(x_1) \wedge \cdots \wedge \varphi_{\alpha_n}(x_n) =$$
$$= \bigwedge f(\alpha_1, \ldots, \alpha_n) \vee \overline{\varphi_{\alpha_1}(x_1)} \vee \cdots \vee \overline{\varphi_{\alpha_n}(x_n)}.$$

Здесь дизъюнкция в первом выражении и конъюнкция во втором берутся по всем двоичным наборам $(\alpha_1, \ldots, \alpha_n)$ значений аргументов $x_1, \ldots, x_n$. Ясно, что оба выражения равны единице только на тех наборах $(\alpha_1, \ldots, \alpha_n)$, на которых функция принимает единичные значения, так как при этом $f(\alpha_1, \ldots, \alpha_n) = 1$, а также по определению характеристических функций (2) $\varphi_{\alpha_i}(x_i) = 1$ и $\overline{\varphi_{\alpha_i}(x_i)} = 0$. На тех наборах $(\alpha_1, \ldots, \alpha_n)$, на которых функция принимает нулевые значения, $f(\alpha_1, \ldots, \alpha_n) = 0$ и, следовательно, оба выражения обращаются в нуль.

Таким образом, первое выражение представляет собой совершенную дизъюнктивную нормальную форму. Ее члены $\varphi_{\alpha_1}(x) \wedge \ldots \wedge \varphi_{\alpha_n}(x)$, называемые *характеристическими конъюнкциями,* играют роль конституент единицы (5. 2. 5). При этом в дизъюнкцию входят только те из них, которые соответствуют $f(\alpha_1, \ldots, \alpha_n) = 1$. Второе выражение представляет собой совершенную конъюнктивную нормальную форму. Его члены $\varphi_{\alpha_1}(x) \vee \ldots \vee \varphi_{\alpha_n}(x)$, называемые *характеристическими дизъюнкциями,* играют роль конституент нуля. При этом в конъюнкцию входят только те из них, которые соответствуют $f(\alpha_1, \ldots, \alpha_n) = 0$.

Приведенные выражения распространяются на случай $k>2$ и рассматриваются как *k-значные совершенные нормальные формы.* Возможность такого обобщения следует из того, что по определению характеристические конъюнкции равны k — 1 и характеристические дизъюнкции равны нулю только при условии $x_i = \alpha_i$ для всех $i = 1, 2, 3, \ldots, n$. Из определений k -значных дизъюнкций и конъюнкций (3) ясно, что совершенная дизъюнктивная нормальная форма содержит только члены, соответствующие $f(\alpha_1, \ldots, \alpha_n) \neq 0$, а конъюнктивная — только члены, соответствующие $f(\alpha_1, \ldots,) \neq k - 1$. Кроме того, для упрощения можно не вписывать значение $f(\alpha_1, \ldots, \alpha_n) = k - 1$ в нервом случае и значение $f(\alpha_1, \ldots, \alpha_n) = 0$ — во втором случае.

В качестве примера рассмотрим четырехзначную функцию переменных, заданную таблицей-

x_1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
x_2	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$f(x_1, x_2)$	2	0	3	0	0	3	1	2	3	1	0	0	3	2	0	3

В соответствии с изложенными правилами имеем: $f(x_1, x_2) =$

$$= (2 \wedge \varphi_0(x_1) \wedge \varphi_0(x_2)) \vee (\varphi_0(x_1) \wedge \varphi_2(x_2)) \vee (\varphi_1(x_1) \wedge \varphi_1(x_2)) \vee (1 \wedge \varphi_1(x_1) \wedge \varphi_2(x_2)) \vee (2 \wedge \varphi_1(x_1) \wedge \varphi_3(x_2)) \vee (\varphi_2(x_1) \wedge \varphi_0(x_2)) \vee (1 \wedge \varphi_2(x_1) \wedge \varphi_1(x_2)) \vee (\varphi_3(x_1) \wedge \varphi_0(x_2)) \vee (2 \wedge \varphi_3(x_1) \wedge \varphi_1(x_2)) \vee (\varphi_3(x_1) \wedge \varphi_3(x_2)) = (2 \vee \varphi_0(x_1)) \vee (\varphi_0(x_2)) \wedge (\varphi_0(x_1) \vee \varphi_1(x_2)) \wedge (\varphi_0(x_1) \vee \varphi_3(x_2)) \wedge (\varphi_1(x_1) \vee \varphi_0(x_2)) \wedge (1 \vee \varphi_1(x_1) \vee \varphi_2(x_2)) \wedge (2 \vee \varphi_1(x_1) \vee \varphi_3(x_2)) \wedge (1 \vee \varphi_2(x_1) \vee \varphi_1(x_2)) \wedge (\varphi_2(x_1) \vee \varphi_2(x_2)) \wedge (\varphi_2(x_1) \vee \varphi_3(x_2)) \wedge (2 \vee \varphi_3(x_1) \vee \varphi_1(x_2)) \wedge (\varphi_3(x_1) \vee \varphi_2(x_2)).$$

5. Функционально полные системы. Возможность представления любой многозначной функции в совершенной дизъюнктивной нормальной форме служит доказательством полноты системы, включающей дизъюнкцию, конъюнкцию, характеристические функции и константы *(система Россера и Тьюкетта).* Предложено также много других функционально полных систем в k -значной логике. Например, полную систему образуют дизъюнкции и циклическое отрицание *(система Поста).* Система, состоящая из единственной функции x_1/x_2 *(система Вебба),* также является полной. Для доказательства полноты системы Поста достаточно выразить константы, характеристические функции и конъюнкцию через дизъюнкцию и циклическое отрицание. Так как $\bigvee_{i=0}^{k-1}(x + i) = k - 1$ (здесь и далее сложение по модулю k), то все константы можно получить с помощью функции циклического отрицания и

дизъюнкции. Можно также по казать, что $\varphi_i(x) = \bigvee_s (x + s) + 1$, где $0 \leqslant s \leqslant k-1$ и $s \neq k-1-i$.

Действительно, если $x = i$, то $i + s \neq k-1$ и $\max(x+s) = k-2$, т. е. $\varphi_i(x) = k-1$. При $x \neq i$ имеем $s = k-1-x$ и $\max(x+s) = k-1$, т. е. $\varphi_i(x) = 0$. Конъюнкция выражается через дизъюнкцию и инверсию $x_1 \wedge x_2 = \overline{\bar{x}_1 \vee \bar{x}_2}$. В свою очередь, $\bar{x} = \bigvee_i^{k-1} e_{i,\,k-1-i}(x)$ и $e_{ij}(x) = |\varphi_i(x) \vee (k-1-j)| + j + 1$, в чем можно убедиться, полагая $x = i$ и $x \neq i$. Приведенная цепочка зависимостей и служит доказательством полноты системы Поста.

Циклическое отрицание и дизъюнкция выражаются через функцию Шеффера—Вебба следующим образом: $x + 1 = x \vee x + 1 = x/x$ и $x_1 \vee x_2 = (x_1 \vee x_2 + 1) + (k-1) = x_1/x_2 + (k-1)$.

Поэтому из полноты системы Поста следует и полнота системы Вебба.

6. Полиномиальные представления. Подобно каноническим многочленам в алгебре Жегалкина (5. 2. 6), рассматривается вопрос о полиномиальных представлениях и в k -значной логике. При решении этого вопроса будом исходить из возможности выражения любой функции в так называемой Σ—П *(сигма-пи) форме:*

$$f(x_1, \ldots, x_n) = \sum f(\alpha_1, \ldots, \alpha_n)\, e_{\alpha_1}(x_1) \ldots e_{\alpha_n}(x_n),$$

где суммирование ведется по всем наборам значений переменных и используются операции сложения и умножения по модулю k. Действительно, поскольку $e_{\alpha_i}(x_i) = 1$ при $x_i = \alpha_i$ и $e_{\alpha_i}(x_i) = 0$ при $x_i \neq \alpha_i$, то произведение $e_{\alpha_1}(x) \cdots e_{\alpha_n}(x)$ отлично от нуля только на наборе $(\alpha_1, \ldots, \alpha_n)$ и равно 1. Поэтому на каждом наборе единственный ненулевой член суммы всегда равен $f(\alpha_1, \ldots, \alpha_n)$, т. е. значению функции на данном наборе. Теперь необходимо найти полиноминальные представления функций $e_{\alpha_i}(x_i)$.

В соответствии с известной *теоремой Ферма* $x^k = x \pmod k$. Если k — простое число, то множество классов вычетов по модулю k образует поле и при $x \neq 0$ обе части этого выражения можно разделить на x. Тогда $x^{k-1} = 1 \pmod k$, что после умножения на k — 1 и прибавления 1 дает $1 + (k-1)\,x^{k-1} = 0 \pmod k$. Полученное выражение справедливо при $x \neq 0$, а при $x = 0$ его левая часть равна 1, т. е. для всех значений $x = 0, 1, \ldots, k-1$ совпадает с функцией $e_{01}(x)$. Обобщая этот результат, можно записать

$$e_{\alpha_i}(x_i) = 1 + (k-1)(x_i - \alpha_i)^{k-1}, \quad \alpha_i = 0, 1, \ldots, k-1.$$

Подставляя эти значения в Σ—П форму и произведя умножения по модулю k, получаем искомый многочлен, представляющий данную

функцию. Все коэффициенты этого многочлена выражаются целыми числами, заключенными между 1 и k— 1, а степени входящих в него переменных не превышают k — 1. Рассмотрим, например, трехзначную функцию двух переменных:

x_1	0 0 0	1 1 1	2 2 2
x_2	0 1 2	0 1 2	0 1 2
$f(x_1, x_2)$	2 0 0	1 2 0	0 1 0

Представляем функцию в Σ—Π форме: $f(x_1, x_2) = 2(1 + 2x_1^2) \times$ $\times (1 + 2x_2^2) + [1 + 2(x_1 - 1)^2](1 + 2x_2^2) + 2[1 + 2(x_1 - 1)^2][1 +$ $+ 2(x_2 - 1)^2] + [1 + 2(x_1 - 2)^2][1 + 2(x_2 - 1)^2]$, откуда после выполнения арифметических операций по модулю 3 получаем $f(x_1, x_2) = 2 - x_1 + x_2^2 + x_1x_2 - x_1x_2^2$.

Из изложенного ясно, что система функций k -значной логики, состоящая из операций суммы и произведения по модулю k и констант, является полной, если k — простое число. В случае составного k эта система не полна, что ограничивает область ее применения.

7. Минимизация многозначных функций. Процесс минимизации многозначных функций зависит от выбранной полной системы функций и значительно сложнее, чем в двузначной логике. Для этой цели можно использовать тождественные преобразования логических формул, стремясь получить выражение, содержащее по возможности наименьшее число входящих в него характеристических функций и аргументов. Развиты также общие методы приведения канонических форм представления функций к минимальным формам.

Рассмотрим, например, систему Россера—Тьюкетта. Тождественные преобразования в этой системе основаны на соотношениях (для упрощения знак конъюнкции заменим точкой): $x \vee x = x \cdot x = x$ *(идемпотентность);* $x \vee 0 = x; x \vee (k - 1) = k - 1; x \cdot 0 = 0;$ $x(k - 1) = x$ *(свойства констант);* $\varphi_i(x) \cdot \varphi_j(x) = 0$ при $i \neq j$; $\varphi_0(x) \vee \varphi_1(x) \vee \cdots \vee \varphi_{k-1}(x) = k - 1;$ $1 \cdot \varphi_1(x) \vee 2 \cdot \varphi_2(x) \vee \cdots \vee (k - 1)\varphi_{k-1}(x) = x$ *(свойства характеристических функций).*

С помощью этих тождеств можно упрощать выражения многозначных функций, а также приводить их к совершенной нормальной форме. Пусть, папримср, трехзначная функция двух переменных задана в виде:

$$f(x_1, x_2) = 1 \cdot \varphi_0(x_1) \vee 1 \cdot \varphi_1(x_2) \vee 1 \cdot \varphi_2(x_2) \vee \varphi_2(x_1) \cdot \varphi_2(x_2).$$

Беря конъюнкцию членов этого выражения с $\varphi_0(x) \vee \varphi_1(x) \vee \varphi_2(x)$ и замещая x той переменной, которая отсутствует в данном члене, получаем

$$f(x_1, x_2) = 1 \cdot \varphi_0(x_1) \cdot [\varphi_0(x_2) \vee \varphi_1(x_2) \vee \varphi_2(x_2)] \vee 1 \cdot \varphi_1(x_2) \wedge$$
$$\wedge [\varphi_0(x_1) \vee \varphi_1(x_1) \vee \varphi_2(x_1)] \vee 1 \cdot \varphi_2(x_2) \cdot [\varphi_0(x_1) \vee \varphi_1(x_1) \vee \varphi_2(x_1)] \vee \varphi_2(x_1) \wedge$$

$$\wedge \varphi_2(x_2),$$

что после упрощения преобразуется к совершенной дизъюнктивной нормальной форме: $f(x_1, x_2) = 1 \cdot \varphi_0(x_1) \cdot \varphi_0(x_2) \vee 1 \cdot \varphi_0(x_1) \wedge$
$$\wedge [\varphi_0(x_1) \vee \varphi_1(x_1) \vee \varphi_2(x_1)] \vee 1 \cdot \varphi_2(x_2) \cdot [\varphi_0(x_1) \vee \varphi_1(x_1) \vee \varphi_2(x_1)] \vee \varphi_2(x_1) \wedge$$

$$\wedge \varphi_1(x_2) \vee 1 \cdot \varphi_0(x_1) \cdot \varphi_2(x_2) \vee 1 \cdot \varphi_1(x_1) \cdot \varphi_1(x_2) \vee 1 \cdot \varphi_1(x_1) \cdot \varphi_2(x_2) \vee 1 \wedge$$

$$\wedge \varphi_2(x_1) \cdot \varphi_1(x_2) \vee \varphi_2(x_1) \cdot \varphi_2(x_2).$$

Упростим полученное выражение. Воспользовавшись свойствами характеристических функций, имеем: $f(x_1, x_2) = 1 \cdot \varphi_0(x_1) \cdot [\varphi_0(x_2) \vee$
$$\vee \varphi_1(x_2) \vee \varphi_2(x_2)] \vee 1 \cdot \varphi_1(x_2) \cdot [\varphi_0(x_1) \vee \varphi_1(x_1) \vee \varphi_2(x_1)] \vee \varphi_2(x_2) \cdot [1 \wedge$$

$$\wedge \varphi_1(x_1) \vee \varphi_2(x_1)] = 1 \cdot \varphi_0(x_1) \vee 1 \cdot \varphi_1(x_2) \vee x_1 \cdot \varphi_2(x_2).$$

Это выражение проще исходного, но нет уверенности, что оно представляет минимальную форму (на самом деле оно не является минимальным).

Получение минимальной формы основано на выделении простых импликант, образующих сокращенную форму и минимизации последней с помощью методов, аналогичных развитым в двузначной логике. Однако сложность этих методов сильно возрастает с увеличением как величины k, так и числа аргументов функции.

8. Сведение к двузначным функциям. Сложность минимизации в многозначной логике заставляет искать такие представления k-значных функций, которые обслуживались бы хорошо разработанным аппаратом двузначной логики. Для этого элементы множества значений k -значной логики объединяются попарно в пересекающиеся подмножества. В соответствии с одним из способов общим элементом всех подмножеств принимается 0, а остальные элементы $1, 2, \ldots, k-1$ области значений k -значной логики входят по одному в каждое подмножество. В результате получаем k — 1 двухэлементных множеств $\{0, 1\}, \{0, 2\}, \ldots, \{0, k-1\}$.

Для представления k -значных функций можно использовать характеристические функции $e_{ij}(x)$ при фиксированных $j = 1, 2, \ldots, k-1$, т. е. k — 1 функций вида:

$$e_{i1}(x) = \begin{cases} 1, & x = i; \\ 0, & x \neq i; \end{cases} \quad e_{i2}(x) = \begin{cases} 2, & x = i; \\ 0, & x \neq i; \end{cases} \quad \cdots \quad e_{i,k-1} = \begin{cases} k-1, & x = i; \\ 0, & x \neq i. \end{cases}$$

По существу эти функции являются неоднородными двузначными функциями, так как они принимают значения из двухэлементных множеств, а их областью определения служит множество $\{0, 1, \ldots, k-1\}$.

Характеристическая конъюнкция, соответствующая некоторому набору $(\alpha_1, \ldots, \alpha_n)$, на котором функция $f(x_1, x_2, \ldots, x_n)$ принимает значение j, имеет вид $e_{\alpha_1 j}(x_1) \cdot e_{\alpha_2 j}(x_2) \ldots e_{\alpha_n j}(x_n)$ и играет роль конституенты j. Очевидно, любая k-значная функция может быть представлена в дизъюнктивной нормальной форме

$$f(x_1, \ldots, x_n) = \bigvee_{j=1}^{k-1} F_j,$$

где F_j — дизъюнкция всех конституент j данной функции. Каждая дизъюнкция F_j представляет функцию на тех наборах, на которых она принимает значение j и отличается от совершенной дизъюнктивной нормальной формы двузначной логики только тем, что вместо аргументов и их отрицаний в элементарные конъюнкции входят характеристические функции.

Минимальная форма для k-значной функции совпадает с дизъюнкцией минимальных форм двузначных логических функций F_j, принимающих значения на двухэлементных множествах.

В качестве примера запишем в рассмотренной форме функцию, таблица которой приведена в (4): $f(x_1, x_2) = [e_{11}(x_1)\, e_{21}(x_2) \vee \vee e_{21}(x_1) \cdot e_{12}(x_2)] \vee [e_{02}(x_1) \cdot e_{02}(x_2) \vee e_{12}(x_1) \cdot e_{32}(x_2) \vee e_{12}(x_1) \cdot e_{12}(x_2)] \vee \vee [e_{03}(x_1) \cdot e_{23}(x_2) \vee e_{13}(x_1) \cdot e_{13}(x_2) \vee e_{23}(x_1) \cdot e_{03}(x_2) \vee e_{33}(x_1) \cdot e_{03}(x_2) \vee \vee e_{33}(x_1) \cdot e_{33}(x_2)]$.

9.Многозначные элементы. Многозначные функции можно реализовать логическими схемами с двузначными элементами путем кодирования в двоичном структурном алфавите. Для непосредственной реализации многозначной функции требуются элементы с многими устойчивыми состояниями.

По аналогии с потенциальными двоичными элементами естественно представить многозначные элементы в виде некоторых схем, состояния которых различаются уровнями электрического напряжения или тока. Однако такие схемы были реализованы только для трех состояний, так как при увеличении числа уровней снижается их надежность. Более перспективными являются динамические многозначные элементы, признаками состояний которых служат параметры (амплитуда, частота, фаза) периодической последовательности импульсов *(импульсные элементы)* или гармонических колебаний *(гармонические элементы)*.

Наибольшее распространение получили *фазоимпульсные многозначные элементы* (ФИМЭ). Общая блок-схема таких элементов показана на рис. 1, *а,* а иллюстрирующие его работу временные диаграммы для $k = 5$ — на рис. 1, *б.*

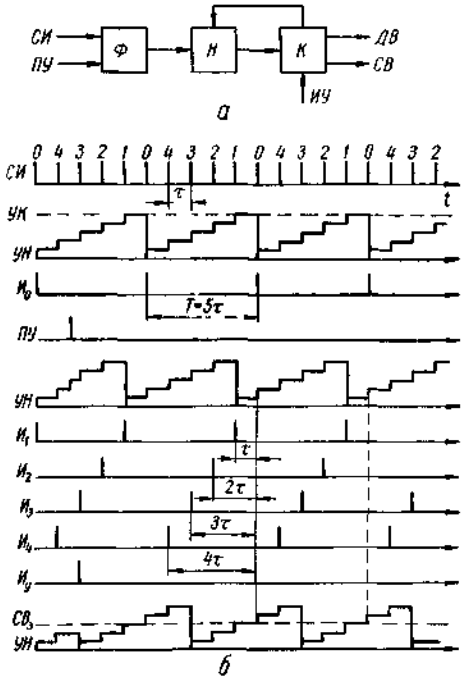


Рис. 1. Фазоимпульсный многозначный элемент.
а — общая блок-схема; *б* — временные диаграммы (при $k = 5$)

Синхронизирующие импульсы *(СИ)* с периодом τ поступают на вход формирователя (*Ф*) и дискретно изменяют уровень накопителя *(УН)*. Накопитель (*Н*) можно выполнить на электрических конденсаторах, магнитных элементах, криотронах и, вообще, любых элементах, способных накапливать энергию. При достижении уровня компарации *(УК)* компаратор *(К)* по цепи сброса приводит накопитель в первоначальное состояние и одновременно выдает импульс. В результате на динамическом выходе *(ДВ)* появляется периодическая последовательность импульсов с периодом $T = k\tau$, где k — число устойчивых состояний.

Один из таких элементов вырабатывает опорную последовательность импульсов $И_0$, которая отождествляется с состоянием 0. На все другие элементы данной системы также подаются синхронизирующие импульсы. Перевод элемента в следующее состояние осуществляется подачей импульса на вход пересчетного управления *(ПУ)*, причем формирователь обеспечивает сдвиг этого импульса так, чтобы он занимал промежуточное положение между соседними синхрони-

зирующими импульсами. Тогда накопитель достигает уровня компарации быстрее на время τ, в результате чего выходная последовательность $И_1$ сдвигается относительно опорной $И_0$ на один такт *СИ.* Воздействие каждого из последующих импульсов, подаваемых на вход *ПУ,* аналогично. В результате на *ДВ* элемента появляются последовательности $И_1$, $И_3$, $И_4$. Пятый импульс переводит элемент в исходное состояние, и процесс периодически повторяется.

Последовательность импульсов $И_i$, сдвинутая по фазе относительно опорной последовательности на время *vx,* является динамическим признаком *i*-го состояния $(i = 0, 1, \ldots, k - 1)$. Статическими признаками состояний служат уровни на выходе *СВ* в моменты времени, определяемые опорной последовательностью $И_0$ (состоянию 0 соответствует высокий уровень, а состоянию k — 1 — наиболее низкий).

Для перевода элемента в любое состояние к информационному входу *(НУ)* кратковременно прилагается последовательность импульсов, соответствующих данному состоянию, или одиночный импульс этой последовательности. Управляющий импульс $И_v$ вызывает срабатывание цепи сброса и переводит накопитель в начальное состояние, благодаря чему происходит соответствующий сдвиг фазы выходной последовательности импульсов.

Фазоимпульсиые многозначные элементы нашли практическое применение в цифровой измерительной технике и автоматике. На их основе разработан и серийно выпускается ряд приборов (счетчики, частотомеры и т. п.).

Разработаны также методы использования таких элементов в вычислительной технике.

Другой перспективный способ реализации многозначных элементов основан на использовании нелинейного звена с характеристикой $U_{вых} = \varphi(U_{вх})$ гребенчатого типа, охваченного обратной связью $U_{вх} = \beta U_{вых}$ (рис. 2, *а).*

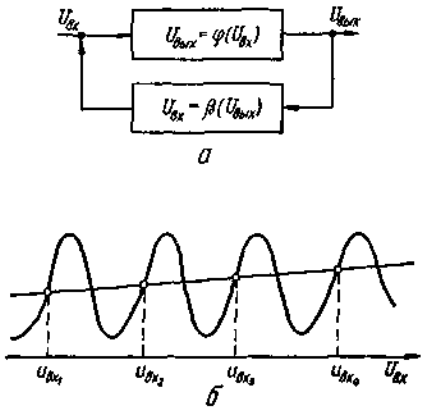


Рис. 2. Многозначный элемент на основе нелинейного звена с характеристикой гребенчатого типа:
а — общая блок-схема; *б* — амплитудная характеристика

Устойчивым состояниям соответствуют отмеченные на рис. 2, *б* пересечения характеристик, каждое из которых характеризуется соответствующим ему напряжением $U_{вх}$. Нелинейный четырехполюсник образуется цепочкой преобразований $U_{вых} = \varphi_1(X_1);\ X_1 = \varphi_2(X_2);\ \ldots;\ X_{n-1} = \varphi_n(U_{вх})$, где $X_1,\ X_2, \ldots, X_{n-1}$ — величины различной физической природы. Для получения гребенчатой характеристики $U_{вых} = \varphi(U_{вх})$ достаточно, чтобы хотя бы одно из преобразований обладало такой характеристикой. Например, при использовании преобразований $U_{вых} = \varphi_1(\omega)$ (гребенчатый фильтр) и $\omega = \varphi_2(U_{вх})$ (электрически управляемый генератор) получим требуемую гребенчатую характеристику. При этом состояния построенного на ее основе многозначного элемента характеризуются частотой ω гармонических колебаний *(частотно-гармонический элемент).* По аналогичному принципу разработаны также *широтно-импульсные элементы,* признаками состояний которых служат дискретные длительности периодической последовательности импульсов.

Важная особенность изложенных принципов реализации многозначных элементов состоит в том, что число их состояний слабо влияет на сложность схемы. Кроме того, наличие динамического и статического признаков состояний открывает дополнительные возможности при проектировании конкретных устройств.

10. Другие логики. Новые идеи реализации логических функций, моделирование нервной деятельности живых организмов, исследование реальных явлений и ситуаций привели к разработке специальных разделов математической логики.

Пороговая логика. С помощью различных технических средств (магнитные элементы, транзисторпо-резистивные схемы, туннельные диоды, параметроны и т. д.) можно построить устройства с n двоичными входами $x_1, \ldots, x_n$ и одним выходом y, функционирование которых описывается соотношениями:

$$y = 1 \text{ при } \sum_{i=1}^{n} \xi_i x_i \geqslant \eta; \quad y = 0 \text{ при } \sum_{i=1}^{n} \xi_i x_i < \eta,$$

где *вес i-го входа* ξ_i и *порог* η выражаются конечными вещественными числами. Такие устройства, условное обозначение которых показано на рис. 3, называют *пороговыми элементами.*

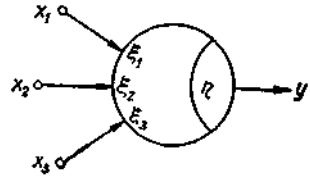


Рис. 3. Пороговый элемент.

Произвольному набору весов ξ_i и порога η, как и любому пороговому элементу, всегда можно сопоставить некоторую логическую функцию, называемую *пороговой функцией.* Однако не всякая логическая функция может быть реализована одним пороговым элементом. Поэтому первой задачей пороговой логики является выделение множества пороговых функций и определение структуры порогового элемента, реализующего пороговую функцию *(синтез порогового элемента).* Если такая реализация невозможна или нецелесообразна, то возникает вторая задача — *синтез схемы из пороговых элементов.*

Мажоритарная логика. В частном случае, когда пороговый элемент имеет нечетное число n входов с единичными весами $(\xi_i = 1)$ и порогом $\eta = (n + 1)/2$, он работает по принципу большинства и называется *мажоритарным элементом.* Действительно, $y = 1$, если взвешенная сумма больше $(n + 1)/2$, т. е. когда больше половины общего числа входных переменных принимает значение 1, и $y = 0$ при условии, что большинство переменных принимает значение 0 (аналогичная ситуация имеет место при голосовании простым большинством). Показано, что любая логическая функция может быть

реализована схемой, состоящей из мажоритарных элементов. Синтез таких схем и является предметом мажоритарной логики.

Нейронная логика. В качестве модели, отражающей функционирование нервных клеток живых организмов, предложен *формальный нейрон* (рис. 4).

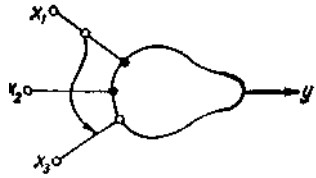


Рис. 4. Формальный нейрон.

Входы нейрона воздействуют на его *тело* посредством *волокон* двух типов: *возбуждающих* с весом 1 и *тормозящих* с весом —1. Место контакта волокна с телом нейрона называют *синапсом* (синапсы возбуждающих волокон обозначаются жирными точками). Выход располагается непосредственно на теле нейрона. Кроме того, допускаются *запрещающие волокна,* оканчивающиеся на запрещаемом волокне, по которому предотвращается поступление сигнала при возбужденном запрещающем входе (на рис. 4 запрет воздействует на волокно входа x_3 при $x_1 = 1$). Как и пороговый элемент, нейрон характеризуется порогом η, причем он возбуждается ($y = 1$), если весовая функция, соответствующая данному набору значений входных переменных, не меньше порога η. Так, изображенный на рис. 4 нейрон будет возбужден на наборах (0, 1, 0), (1, 0, 0), (1, 0, 1), (1, 1, 0) и (1, 1, 1).

В отличие от пороговых элементов, на одном формальном нейроне можно реализовать *любую логическую функцию.* Для синтеза нейронов и нейронных схем успешно используются диаграммы Венна и различные их модификации. При этом стремятся получать схемы, в которых общее число волокон минимально (в этом смысле реализация нейронной схемой может оказаться предпочтительнее одного нейрона). Формальный нейрон приобрел черты универсальной модели в кибернетике и других отраслях науки и техники. В более полной модели абстрактного нейрона вводятся временные соотношения: прохождение сигналов через синапсы происходит с задержкой на временной такт, а значение порога является функцией дискретного времени.

Логика потенциально-импульсных схем. В потенциально-импульсных схемах двоичные переменные представляются сигналами двух типов: уровнями электрических напряжений (потенциалов) и импульсами.

Например, для входных переменных положительный потенциал соответствует 1, а отрицательным — 0, а для выходных переменных наличие импульса соответствует 1, а его отсутствие — 0, причем выходные импульсы могут появляться только при изменении значений входных переменных (рис. 5).

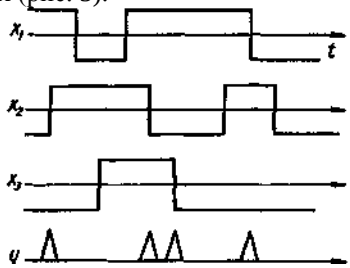


Рис. 5. Временные диаграммы входов x_1, x_2, x_3 и выхода y потенциально-импульсной схемы.

Для представления потенциально-импульсных функций вводится оператор dx_i, при изменении значения x_i с 1 на 0 и $d\bar{x}_i$ при изменении значения x_i с 0 на 1. Роль конституент единицы в дизъюнктивной нормальной форме играют конъюнкции $\tilde{x}_1 \ldots \tilde{x}_{i-1}\tilde{x}_{i+1}\ldots\tilde{x}_n d\tilde{x}_i$, где x_i — входная переменная, при изменении которой выходная переменная принимает значение 1.Так, для приведенной на рис. 5 выходной функции $y = x_1\bar{x}_3 d\bar{x}_2 \vee x_1 x_3 dx_2 \vee x_1\bar{x}_2 dx_3 \vee x_2\bar{x}_3 dx_1$. Такое представление положено в основу методов анализа и синтеза потенциально-импульсных схем.

Фазоимпульсная логика. При фазоимпульсном кодировании двоичных переменных их значения различаются сдвинутыми по времени импульсами. Синхронизация осуществляется двумя последовательностями импульсов t_0 и t_1, играющими роль констант 0 и 1(рис.6,*а*).

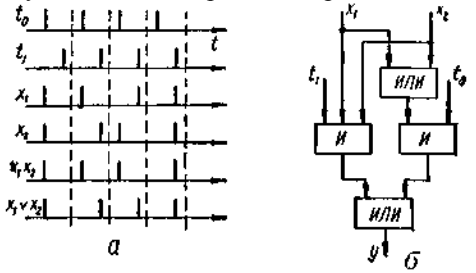


Рис. 6. Фазоимпульсное кодирование:
а — временные диаграммы; *б* — логическая схема для конъюнкции x_1x_2.

Фазоимпульсное представление логической функции можно получить на основании таблицы соответствия как дизъюнкцию двух выражений. Первое из них является дизъюнкцией всех конституент нуля, умноженной на t_0, а второе — дизъюнкцией всех конституент единицы, умноженной на t_1. Минимизируя каждое из этих выражений и используя обычные методы синтеза логических схем, получаем схему, реализующую данную функцию. Например, для конъюнкции двух переменных имеем:

$$x_1x_2 = (x_1x_2 \vee x_1\bar{x}_2 \vee \bar{x}_1x_2)t_0 \vee x_1x_2t_1 = (x_1 \vee x_2)t_0 \vee x_1x_2t_1.$$

Соот ветствующая схема показана на рис. 6, *б*. Аналогичная методика может быть использована и для многозначных функций. Усложнение схем при фазоимпульсном кодировании компенсируется их большей надежностью и помехозащищенностью. Разумеется, схемы можно существенно упростить при использовании специальных элементов, приспособленных для фазоимпульсной логики (например, многозначных фазоимпульсных элементов).

5.7. Логика высказываний

1. Закон исключения третьего. Рассматривая высказывания как двоичные переменные, обычно считают, что они удовлетворяют *закону исключения третьего:* каждое высказывание может быть истинным или ложным (третьего не дано). При этом высказывание не может быть одновременно и истинным и ложным *(закон противоречия).* Значения «истина» и «ложь», соответствующие 1 и 0 в двузначной логике, в логике высказываний он обозначается через «И» и «Л».

Истинность данного высказывания в повседневной жизни устанавливается на основе анализа его смысла. Например, высказывание «Киев — столица Украины» — истинно, а «100 < 10» — ложно. Однако даже в таких категоричных случаях их истинность относительна. Первое предложение перестает быть истинным, если речь идет о периоде, когда столицей Украины был Харьков. Второе предложение становится истинным, если считать, что число 100 записано в двоичной системе счисления, а 10 — в десятичной («4 < 10»).

Таким образом, высказывание может быть либо истинным, либо ложным в зависимости от обстоятельств, которыми руководствуются при его истолковании. Обычно эти обстоятельства не фигурируют явно в простом высказывании. Например, истинность таких высказываний, как «Хорошая погода», «Сегодня — 16 января», «Результат измерений

диаметра цилиндра равен 52 мм» зависит соответственно от вкусов или критерия оценки погоды, сегодняшней даты, требуемой точности измерения. Логика высказываний отвлекается от конкретного смысла предложений, и ответственность за их истолкование возлагает на лиц, компетентных в соответствующей области. Она дает лишь общие методы анализа сложных высказываний и принципы логических рассуждений и доказательств.

Принятие закона исключения третьего позволяет полностью использовать в логике высказываний аппарат двузначной логики. Дальнейшее развитие логики высказываний основано на допущении нескольких значений истинности (например, кроме значений «истина» и «ложь» допускается третье значение — «неопределенность»). В подобных случаях используется аппарат многозначной логики. Если истинность предложений определяется с некоторой вероятностью, то логика высказываний превращается в вероятностную логику. В этом разделе рассматривается только двузначная логика высказываний, причем для обозначения значения «истина» будем применять 1, а значения «ложь» — 0.

2. Сентенциональные связки. Так называют слова «не», «и», «или», «если..., то» и «если и только если», с помощью которых в обычном языке из простых предложений образуются сложные предложения. В каждой из этих связок соответствует своя логическая операция: отрицание, конъюнкция, дизъюнкция, импликация и эквиваленция. Обычно высказывания обозначают прописными буквами, а для операций используются те же символы, что и в алгебре логики. Таблицы соответствия в логике высказываний называют *истинностными таблицами.* Для указанных пяти связок они имеют вид:

P	0	1
$\bar{P}$	1	0

P	0	0	1	1
Q	0	1	0	1
PQ	0	0	0	1
$P \vee Q$	0	1	1	1
$P \to Q$	1	1	0	1
$P \sim Q$	1	0	0	1

Сентенциональные связки в разговорном языке допускают различные варианты. Поэтому при записи сложного предложения в виде формулы алгебры логики важно выяснить характер логической связи между предложениями, не вдаваясь в смысл самих предложений.

Истолкование *отрицания* $\bar{P}$, *конъюнкции* PQ и *дизъюнкции* $P \vee Q$ обычно не вызывает трудностей. *Импликации* $P \to Q$ в обычной речи соответствует условное предложение «если P, то Q», причем P называется *посылкой (антецедентом),* а Q — *следствием*

(консеквентом). Могут встретиться и другие выражения, имеющие тот же тип логической связи, например: «*P* влечет *Q*», «*P* только тогда, когда *Q*», «*P* есть достаточное условие для *Q*», «*Q* при условии, что *P*», «*Q* есть необходимое условие для *P*» и т. п. *Эквиваленция* $P \sim Q$ определяет логическую связь в так называемых *биусловных предложениях* типа «*P*, если и только если *Q*» или в других грамматических формах: «*P* тогда и только тогда, когда *Q*», «если *P*, то *Q* и обратно, если *Q*, то *P*», «*Q* есть необходимое и достаточное условие для *P*».

3. Формулы и подстановки. Всякое сложное предложение, которое состоит из простых предложений, связанных сентенциональными связками, можно представить в символической форме. В результате получаем *высказывательную формулу*. На каждом наборе значений истинности букв (переменных) формула принимает некоторое значение. Следовательно, всякую формулу логики высказываний можно рассматривать как *истинностную функцию*.

Рассмотрим, например, сложное высказывание: *«Если* применить стальные конструкции (*P*), *то* масса снижается (*Q*) *и* стоимость увеличивается (*R*). Стальные конструкции *не* применяются $(\bar{P})$, *а* масса снижается *(Q)».* Соответствующая формула $(P \to QR)\bar{P}Q$ представляется следующей таблицей истинности:

P	0	0	0	0	1	1	1	1
Q	0	0	1	1	0	0	1	1
R	0	1	0	1	0	1	0	1
QR	0	0	0	1	0	0	0	1
$P \to QR$	1	1	1	1	0	0	0	1
$\bar{P}Q$	0	0	1	1	0	0	0	0
$(P \to QR)\,\bar{P}Q$	0	0	1	1	0	0	0	0

Отсюда видно, что сложное предложение истинно на двух наборах значений аргументов *P, Q, R,* а именно: (0, 1,0) и (0, 1, 1), а на остальных наборах оно ложно.

В логике высказываний дается следующее определение формулы:

1) переменные высказывания суть формулы;

2) если *A* и *B* — формулы, то (AB), $(A \vee B)$, $(A \to B)$, $(A \sim B)$ и $\bar{A}$ также формулы.

Это определение имеет *рекурсивный характер* в том смысле, что первая его часть определяет элементарные формулы, а вторая позволяет из любых формул образовать новые формулы. При записи формул используются обычные упрощения. Пусть, например, требуется получить формулу $(A \to A\bar{B}) \to ((C \vee D) \to AB)$.

Выбираем необходимое множество элементарных формул *A, B, C, D*. Затем последовательно получаем формулы

$$\overline{AB},\ A \to \overline{AB},\ (C \vee D) \to AB,\ (A \to \overline{AB})((C \vee D) \to AB).$$

Как видно, процесс образования формулы происходит путем расширения их множества до тех пор, пока это множество не будет содержать требуемую формулу. Все формулы, построенные в указанном процессе, называются *частями результирующей формулы.*

Если имеется некоторая высказывательная формула, то можно построить соответствующее сложное предложение, заменяя буквы простыми предложениями (одинаковые вхождения букв замещаются одним и тем же предложением). Полученное таким путем предложение называется *подстановкой в данную формулу.* Так, полагая *P* — «идет снег», *Q* — «2×2=4» и *R* — «слоны зеленые», по формуле $P \to QR$ получаем подстановку: «Если идет снег, то 2×2 = 4 и слоны зеленые». Истинность этого высказывания определяется только приведенной выше таблицей и никоим образом не связана с конкретным содержанием как простых предложений, так и полученного в результате их объединения сложного предложения Как видно из таблицы, истинностная функция истинна на всех наборах значений аргументов, кроме наборов {1,0,0}, {1,0, 1} и {1, 1, 0}. Например, при *P* = 0, *Q*= 0 и *R* = 1, получим истинное высказывание: «Если не идет снег, то 2 × 2 × 4 и слоны зеленые».

4. Сложные высказывания и «здравый смысл». При первом знакомстве с логикой высказываний трудно без чувства юмора принять подобные предложения. Наш опыт подсказывает, что подвергать сомнению истину «2×2 = 4» так же нелепо, как и утверждать, что «слоны зеленые». Кроме того, между посылкой «идет снег» и ее следствием нет причинной связи. Поэтому с точки зрения «здравого смысла» такие высказывания кажутся несуразными и возможность их появления в логике высказываний следовало бы исключить.

Однако необходимо преодолеть психологический барьер и понять, что ограничения, основанные на «здравом смысле» и причинной связи в логике высказываний не только невозможны, но и нежелательны. В (1) уже указывалось на относительность истинности или ложности того или иного высказывания. Если бы множество допустимых высказываний было подвергнуто испытанию «здравым смыслом», то возникли бы непреодолимые трудности из-за отсутствия строгого определения, что следует под этим понимать. Человеку, который никогда не видел снега и не слышал о нем, фраза «идет снег» покажется бессмысленной, а высказывание «слоны зеленые» может иметь вполне определенный смысл, если речь идет, например, о выборе цвета для игрушечных слонов. Аналогичные соображения

можно привести и в пользу допущения логической связи между любыми предложениями без учета причинной зависимости между ними.

Поэтому логика высказываний, отвлекаясь от конкретного содержания высказываний, по существу занимается лишь анализом и синтезом высказывательных формул и изучением отношений между высказываниями. Что же касается «здравого смысла», то он должен проявляться при использовании законов логики высказываний в ее конкретных приложениях.

5. Тавтологии. Тождественно истинная формула, т. е. такая формула, которая принимает значения 1 при любых значениях ее компонентов, называется *тавтологией.* Тождественно ложная формула на всех наборах ее компонентов принимает значение 0 и называется *противоречием.* Если в технических приложениях логические функции, выражаемые тавтологиями или противоречиями, практически не представляют интереса, то в логике высказываний они играют первостепенную роль.

Примером тавтологии может служить высказывание: «Если внедрить новую технологию *(P),* то качество продукции улучшится *(Q).* При улучшении качества продукции (*Q*), ее сбыт увеличивается *(R).* Новая технология внедрена *(P).* Следовательно, сбыт продукции увеличился *(R)».* Оно выражается формулой $(P \to Q)(Q \to R)P \to R$.

Чтобы выяснить, является ли данная формула тавтологией, можно составить для нее истинностную таблицу. Так, для приведенной выше формулы имеем:

P	0	0	0	0	1	1	1	1
Q	0	0	1	1	0	0	1	1
R	0	1	0	1	0	1	0	1
$P \to Q$	1	1	1	1	0	0	1	1
$Q \to R$	1	1	0	1	1	1	0	1
$(P \to Q)(Q \to R)P$	0	0	0	0	0	0	0	1
$(P \to Q)(Q \to R)P \to R$	1	1	1	1	1	1	1	1

Можно также воспользоваться зависимостями (1.7) $x_1 \to x_2 = \bar{x}_1 \vee x_2$; $x_1 \sim x_2 = x_1x_2 \vee \bar{x}_1\bar{x}_2 = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2)$ и преобразовать высказывательную формулу к нормальной форме. Если хотя бы один член дизъюнктивной нормальной формы окажется равным 1, то соответствующая ей формула является тавтологией. Если хотя бы один член конъюнктивной нормальной формы окажется равным 0, то соответствующая ей формула является противоречием. Так, для нашего примера имеем:

$$(P \to Q)(Q \to R)P \to R = (\bar{P} \vee Q)(\bar{Q} \vee R)P \to R =$$

$= (\bar{P}\bar{Q} \vee \bar{P}R \vee QR)P \to R = PQR \to R = \overline{PQR} \vee R = \bar{P} \vee \bar{Q} \vee \bar{R} \vee R =$
$= \bar{P} \vee \bar{Q} \vee (\bar{R} \vee R) = \bar{P} \vee \bar{Q} \vee 1 = 1.$

Очевидно, формула не является тавтологией, если она принимает значение 0 хотя бы на одном наборе значений переменных. Этим обстоятельством можно воспользоваться для распознавания тавтологий сокращенным методом «обратного рассуждения», заключающемся в поиске таких переменных, при которых формула оказывается ложной. Так, приведенная выше формула может принять значение 0, если и только если R ложно, а $(P \to Q)(Q \to R)P$ истинно. При этом должны быть истинны $P \to Q$, $Q \to R$ и P. При истинном P формула $P \to Q$ истинна только при истинном Q. В свою очередь, при истинном Q формула $Q \to R$ истинна только при истинном R. Таким образом, анализируемая формула может быть ложной, если и только если R одновременно и истинно и ложно, что невозможно в силу закона противоречия. Следовательно, она является тавтологией.

Для указания на то, что данная формула является тавтологией, используется знак $\models$, который помещается перед формулой, например: $\models (P \to Q)(Q \to R)P \to R$.

6. Законы логики высказываний. Различные подстановки в тавтологию, независимо от их конкретного содержания, всегда являются истинными предложениями в силу одной только своей логической структуры. Иначе говоря, тавтологии можно рассматривать как некоторые *логически истинные схемы* рассуждений или утверждений. Поэтому они играют роль *законов (теорем) логики высказываний,* претендующих на установление методов построения правильных умозаключений.

Существует бесконечное множество тавтологий, а значит, и законов логики высказываний. Наиболее часто используемые из них следующие: $P \to P$ *(закон тождества),* $P \vee \bar{P}$ *(закон исключения третьего),* $\overline{P\bar{P}}$ *(закон противоречия),* $\bar{\bar{P}} \sim P$ *(закон двойного отрицания),* $P \to (Q \to P)$ *(добавление антецедента* или *verum ex quodlibet* — истина из чего угодно), $\bar{P} \to (P \to Q)$ *(ex falso quodlibet* — из ложного что угодно), $(P \to Q)P \to Q$ *(закон отделения* или *modus ponens),* $(P \to Q)\bar{Q} \to \bar{P}$ *(modus tollens),* $(P \to Q)(Q \to R) \to (P \to R)$ *(закон силлогизма),* $(P \to Q) \to (\bar{Q} \to \bar{P})$ *(закон контрапозиции).*

Каждый из законов логики высказываний отображает в символической форме некоторую схему доказательства. Например, в соответствии с законом отделения, если истинно, что некоторое высказывание P имплицирует высказывание Q и, кроме того, P истинно, то истинно и Q. Modus tollens применяется при доказательстве от

противного: желая доказать утверждение P, предполагается, что P ложно, и показывается, что P имплицирует некоторое высказывание Q, о котором известно, что оно ложно ($\bar{Q}$ истинно). Отсюда заключается, что P истинно.

7. Равносильность. Две формулы называются *равносильными,* если уа всех наборах значений входящих в них переменных эти формулы принимают одинаковые значения. Для обозначения этого отношения часто употребляют символ $\leftrightarrow$, так что равносильность формул A и B символически записывается как $A \leftrightarrow B$. Легко видеть, что равносильность — это отношение эквивалентности: оно рефлексивно $(A \leftrightarrow A)$, симметрично (если $A \leftrightarrow B$, то $B \leftrightarrow A$) и транзитивно (из $A \leftrightarrow B$ и $B \leftrightarrow C$ следует, что $A \leftrightarrow C$). Поэтому равносильность называют также *логической эквивалентностью.*

Равносильность формул логики высказываний вытекает из тождественности соответствующих формул алгебры логики. Так, в соответствии с булевой алгеброй и тождественных преобразований получаем следующие равносильности:

$\bar{\bar{A}} \leftrightarrow A;\ A \vee A \leftrightarrow A;\ AA \leftrightarrow A;\ A \vee B \leftrightarrow B \vee A;\ AB \leftrightarrow BA;\ A \vee (B \vee C) \leftrightarrow (A \vee B) \vee C;\ A(BC) \leftrightarrow (AB)C;\ A(B \vee C) \leftrightarrow AB \vee AC;\ A \vee BC \leftrightarrow (A \vee B)(A \vee C);\ \overline{A \vee B} \leftrightarrow \bar{A}\bar{B};\ \overline{AB} \leftrightarrow \bar{A} \vee \bar{B};\ A \vee AB \leftrightarrow A;\ A(A \vee B) \leftrightarrow A;\ A \vee \bar{A}B \leftrightarrow A \vee B$ и т. д. Кроме того, с помощью отношения равносильности

$A \to B \leftrightarrow \bar{A} \vee B;\ A \sim B \leftrightarrow AB \vee \bar{A}\bar{B} \leftrightarrow (A \vee \bar{B})(\bar{A} \vee B);\ A \vee B \leftrightarrow \bar{A} \to B;$
$AB \leftrightarrow \overline{A \to \bar{B}};\ A \sim B \leftrightarrow (A \to B)(B \to A).$

выражаются различные связки между формулами:

Эти и подобные им равносильные соотношения можно использовать для преобразования и упрощения структуры сложного высказывания. Так, для примера из (3) имеем:

$$(P \to QR)\bar{P}Q \leftrightarrow (\bar{P} \vee QR)\bar{P}Q \leftrightarrow \bar{P}Q \vee \bar{P}QR \leftrightarrow \bar{P}Q.$$

Между отношением равносильности и эквиваленцией формул существует следующая связь: если A и B — равносильны, то $A \sim B$ — тавтология, и обратно, если $A \sim B$ — тавтология, то A и B — равносильны. Это сокращенно записывается так: $\models A \sim B$, если и только если $A \leftrightarrow B$. Справедливость этого утверждения следует непосредственно из определения равносильности и таблицы истинности для эквиваленции. Действительно, если $A \leftrightarrow B$, то A может принимать только то значение, что и B и, следовательно, их эквиваленция $A \sim B$ всегда истинна и является тавтологией. Если $A \sim B$ — тавтология, то A и B могут иметь, только одинаковые значения (0 или 1) и, следовательно, $A \leftrightarrow B$.

Из изложенного ясно, что тавтологии можно получить из равносильности заменой знака $\leftrightarrow$ на $\sim$. Так, из равносильности $A \vee AB \leftrightarrow A$ получаем тавтологию $\models (A \vee AB) \sim A$. Доказательство тавтологий, например $\models (A \to B)(A \to C) \sim (A \to BC)$ можно выполнить с помощью преобразований:
$(A \to B)(A \to C) \leftrightarrow (\bar{A} \vee B)(\bar{A} \vee C) \leftrightarrow \bar{A} \vee \bar{A}B \vee$
$\vee\, \bar{A}C \vee BC \leftrightarrow \bar{A} \vee BC \leftrightarrow A \to BC$.

8. Логическое следствие. Говорят, что формула B является *логическим следствием* формулы A и пишут $A \Rightarrow B$, если B истинно на всех наборах значений переменных, для которых A истинно. Легко убедиться, что $A \Rightarrow B$, если и только если $\models A \to B$. Действительно, в соответствии с определением импликации $A \to B$ ложно только при истинном A и ложном B и, следовательно, если $A \to B$ — тавтология, то из истинности A всегда следует истинность B, т. е. $A \Rightarrow B$. Обратно, если $A \Rightarrow B$, то исключается случай, когда A истинно и B ложно, а значит $A \to B$ истинно на всех наборах значений переменных, т. е. $\models A \to B$.

Логическое следствие $A \Rightarrow B$ означает, что из истинности A следует истинность B, но если A ложно, то относительно B ничего утверждать нельзя. Это отношение обобщается на совокупность высказываний: B есть логическое следствие высказываний A_1, $A_2, \ldots, A_m$, если из истинности всех A_i $(i = 1, 2, \ldots, m)$ следует истинность B. Из определения конъюнкции можно заключить, что это сводится к соотношению $A_1 A_2 \ldots A_m \Rightarrow B$, необходимым идостаточным условием которого является тавтология $\models A_1 \times A_2 \ldots A_m \to B$.

Пусть, например, даны высказывания $(A \to B)(C \to D)$, $BD \to E$, $\bar{E}$ и необходимо установить, является ли высказывание $\bar{A} \vee \bar{C}$ логическим следствием. Это сводится к доказательству тавтологии $\models ((A \to B)(C \to D))(BD \to E)\bar{E} \to (\bar{A} \vee \bar{C})$. Воспользовавшись методом «обратного рассуждения», положим, что следствие $\bar{A} \vee \bar{C}$ ложно (A и C истинны) при истинном значении всех посылок. Тогда, как следует из первой посылки, B и D должны быть истинны, а из истинности BD и второй посылки следует истинность E. Но это противоречит третьей посылке $\bar{E}$, что и доказывает данную тавтологию.

Между логическим следствием и логической эквивалентностью имеется связь, которая вытекает из соотношения $A \sim B \leftrightarrow (A \to B) \wedge$ $\wedge (B \to A)$, приведенного в (7). Оно означает: $A \sim B$, если и только

если $A \to B$ и $B \to A$. Пусть $A \sim B$ — тавтология, тогда $A \to B$ и $B \to A$ —также тавтологии, т. е. $\models A \sim B$, если и только если $\models A \to B$ и $\models B \to A$. А это равносильно утверждению: $A \leftrightarrow B$, если и только если $A \Rightarrow B$ и $B \Rightarrow A$.

Логическое следствие есть отношение порядка; так, оно рефлексивно $(A \Rightarrow A)$, траизитивпо (если $A \Rightarrow B$ и $B \Rightarrow C$, то $A \Rightarrow C$) и антисимметрично (из $A \Rightarrow B$ и $B \Rightarrow A$ следует $A \leftrightarrow B$).

9. Правила вывода. Формальная теория вывода ставит своей главной задачей образование из некоторой совокупности исходных тавтологий новых формул, которые также являются тавтологиями. Эта задача решается с помощью *правил вывода:*

1) если A — тавтология, то, заменяя в ней букву X всюду, где она входит, произвольной формулой B, получаем также тавтологию *(правило подстановки);*

2) если A и $A \to B$ суть тавтологии, то B — также тавтология *(правило заключения).*

Первое из этих правил почти очевидно, а второе непосредственно следует из закона *modus ponens* (6).

Формула называется *выводимой в исчислении высказываний,* если она может быть получена из конечной совокупности исходных формул путем конечного числа шагов применения правил вывода. Вообще говоря, не все тождественно истинные формулы могут быть выведены из произвольного множества тавтологий. В то же время строго доказано, что можно выбрать такую конечную совокупность исходных тавтологий *(аксиом исчисления высказываний)*, из которой выводимы все тождественно истинные формулы. Это важное положение решает проблему *полноты исчисления высказываний.*

Предложено много различных систем аксиом исчисления высказываний. Одна из них включает следующие тавтологии:

1) $A \to (B \to A)$;

2) $((A \to B) \to A) \to A$; 3) $(A \to B) \to ((B \to C) \to (A \to C))$; 4) $AB \to A$; 5) $AB \to B$; 6) $(A \to B) \to ((A \to C) \to (A \to BC))$; 7) $A \to (A \vee B)$; 8) $B \to (A \vee B)$; 9) $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$; 10) $(A \sim B) \to (A \to B)$; 11) $(A \sim B) \to (B \to A)$; 12) $(A \to B) \to ((B \to A) \to (A \sim B))$;

13) $(A \to B) \to (\bar{B} \to \bar{A})$; 14) $A \to \bar{\bar{A}}$; 15) $\bar{\bar{A}} \to A$.

Выведем, например, тавтологию $AB \to BA$. Подстановка в аксиому (6) вместо A формулы AB дает $\models (AB \to B) \to ((AB \to C) \to (AB \to BC))$, что после подстановки A вместо C приводится к $\models (AB \to B) \to ((AB \to A) \to (AB \to BA))$. Посылка в этой формуле есть аксиома (5), поэтому на основе правила заключения

$\models (AB \to A) \to (AB \to BA)$. Так как посылка в полученной тавтологии является аксиомой (4), то, применяя еще раз правило заключения, получаем $\models AB \to BA$, что и требовалось доказать.

Формализация процесса вывода имеет большое теоретическое значение и позволяет построить схему доказательства, которая может быть реализована на вычислительных машинах. Однако сложность аксиоматического подхода к выводу тавтологий заставляет искать и применять специальные правила, которые сокращают многократное применение основных правил вывода.

10. Дедуктивный метод. Более краткий и простой способ вывода основан на *теореме дедукции:* если формула B является логическим следствием формул $A_1, A_2, \ldots, A_m$, т. е. $A_1, A_2, \ldots, A_m \Rightarrow B$, то $\models A_1 \to (A_2 \to (\ldots \to (A_m \to B) \ldots))$. При этом говорят, что формула B *выводима из формул* $A_1, A_2, \ldots, A_m$.

Дадим алгебраическое доказательство теоремы дедукции, рассматривая в соответствии с (8) логическое следствие $A_1, A_2, \ldots, A_m \Rightarrow B$ как $A_1 A_2 \ldots A_m \Rightarrow B$. Преобразуем по формулам из (7) тавтологию

$$A_1 A_2 \ldots A_m \to B \leftrightarrow \overline{A_1 A_2 \ldots A_m} \vee B \leftrightarrow \overline{A}_1 \vee \overline{A}_2 \vee \cdots \vee \overline{A}_m \vee B \leftrightarrow \overline{A}_1 \vee$$

$$\vee\, \overline{A}_2 \vee \cdots (A_m \to B) \leftrightarrow (A_1 \to (A_2 \to (\cdots \to (A_m \to B) \ldots))).$$

Так как исходная формула — тавтология, то полученная логически эквивалентная ей формула также является тавтологией, что и требовалось доказать.

Значение теоремы дедукции состоит в том, что логическое следствие B из совокупности посылок $A_1, A_2, \ldots, A_m$ представимо в виде тавтологий типа $\models A_1 A_2 \ldots A_p \to (A_{p+1} \to \ldots (A_m \to B) \ldots)$.

Справедливо и обратное утверждение: если имеется тавтология, содержащая цепочку импликаций типа $(A_1 \to (A_2 \to \cdots (A_p \to (A_{p+1} \to \cdots \to \cdots (A_m \to B) \ldots))))$, то она может быть представлена эквивалентной формулой $\models A_1 A_2 \ldots A_p \to (A_{p+1} \to \cdots (A_m \to B) \ldots)$, которой соответствует соотношение $A_1 A_2 \ldots A_p \Rightarrow (A_{p+1} \to \ldots (A_m \to B) \ldots)$. Из теоремы дедукции и определения логического следствия вытекают следующие положения:

1) $A_1, A_2, \ldots, A_m \Rightarrow A_i \ (i = 1, 2, \ldots, m)$,

т. е. любая из совокупности посылок является логическим следствием этой совокупности;

2) если $A_1, A_2, \ldots, A_m \Rightarrow B_j$ $(j = 1, 2, \ldots, n)$ и $B_1, B_2, \ldots, B_n \Rightarrow B$, то $A_1, A_2, \ldots, A_m \Rightarrow B$.

С помощью этих правил можно представить доказательство того, что формула B есть логическое следствие формул $A_1, A_2, \ldots, A_m$ в виде *цепочки формул,* последней из которых является B. Промежуточные формулы $B_1, B_2, \ldots, B_n$ получаются на основании известных логических законов, аксиом и эквивалентностей. На основе теоремы дедукции используемые тавтологии и результирующее соотношение преобразуются к требуемой форме.

В качестве примера докажем, что $(A \vee B) \to C$, $C \to (D \vee E)$, $E \to F$, $\bar{D}\bar{F} \Rightarrow \bar{A}$. Из первой пары посылок на основе закона силлогизма получаем $(A \vee B) \to (D \vee \underline{E})$. Из последней посылки следует $\bar{D}$ и $\bar{F}$. Из посылки $E \to F$ и $\bar{F}$ выводим (modus tollens) $\bar{E}$. Из $\bar{D}$ и $\bar{E}$ получаем $\bar{D}\bar{E} \leftrightarrow \overline{D \vee E}$, что совместно с $(A \vee B) \to (D \vee E)$ в соответствии с modus tollens дает $\overline{A \vee B} \leftrightarrow \bar{A}\bar{B}$, откуда выводим $\bar{A}$. Наглядно этот процесс вывода изображается диаграммой, показанной на рис. 1.

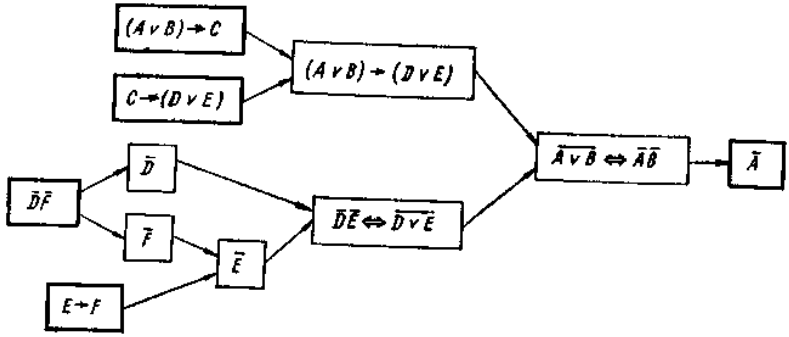


Рис. 1. Диаграмма вывода $\bar{A}$ из посылок $(A \vee B) \to C$, $C \to (D \vee E)$, $E \to F$, $\bar{D}\bar{F}$.

Если в качестве логического следствия выводится конъюнкция некоторого высказывания и его отрицания $A \wedge \bar{A}$, то это свидетельствует о *противоречивости* посылок (из нее выводится произвольное высказывание, как истинное, так и ложное).

5.8. Логика предикатов

1. Высказывания и предикаты. В то время как логика высказываний проявляет интерес только к логической связи между предложениями, логика предикатов проникает и в структуру самих предложений в смысле связи того, о ком или о чем идет речь *(субъект)* с тем, что говорится о данном предмете *(предикат).* Поэтому язык логики предикатов лучше приспособлен для выражения логических связей между различными понятиями и утверждениями. Как известно, *n-местный предикат* $P(x_1, x_2, \ldots, x_n)$ является неоднородной двузначной логической функцией. Аргументы $x_1, x_2, \ldots, x_n$ представляют собой объекты из множеств их определения $X_1, X_2, \ldots, X_n$, т. е. $x_1 \in X_1, x_2 \in X_2, \ldots, x_n \in X_n$ и называются *предметными переменными.* Конкретные значения аргументов называют *предметными постоянными.* Предметные переменные и предметные постоянные образуют класс логических понятий, называемых *термами.*

При замещении аргумента x_k (предметной переменной) некоторым его значением a (предметной постоянной) n-местный предикат $P(x_1, x_2, \ldots, x_n)$ превращается в $(n - 1)$-местный предикат $P(x_1, \ldots, x_{k-1}, a, x_{k+1}, \ldots, x_n)$ и от переменной x_k он уже не зависит. Приписав значения всем переменным $x_1, x_2, \ldots, x_n$ из соответствующих областей определения, мы получим высказывание, которое можно рассматривать как 0-*местный предикат.*

Например, трехместный предикат $P(x_1, x_2, x_3)$ = «x_1 есть сумма x_2 и x_3» при подстановке $x_1 = 5$ переходит в двуместный предикат $P(5, x_2, x_3)$ — «5 есть сумма x_2 и x_3», а при дальнейшей подстановке $x_2 = 2$ — в одноместный предикат $P(5, 2, x_3)$ = «5 есть сумма 2 и x_3». Очевидно, при $x_3 = 3$ он становится истинным высказыванием, а при всех $x_3 \neq 3$ ложным высказыванием.

2. Кванторы. В логике предикатов большое значение имеют две операции, называемые *кванторами,* с помощью которых выражают отношения общности и существования. Пусть $P(x)$ — предикат, определенный на множестве M. Утверждение, что все $x \in M$ обладают свойством $P(x)$, записывают с помощью *квантора общности* $\forall x$ в виде $\forall x P(x)$, что читается «для всех x, P от x». Утверждение, что существует хотя бы один объект x из M, обладающий свойством $P(x)$, записывают с помощью *квантора существования* $\exists x$ в виде $\exists x P(x)$, что читается «существует такое x, что P от x».

Хотя в выражениях $\forall x P(x)$ и $\exists x P(x)$ и встречается буква x, но они не зависят от значений этой переменной. Кванторы $\forall x$ и $\exists x$

связывают переменную x, превращая одноместный предикат в высказывание. Очевидно, $\forall x P(x)$ истинно только при условии, что *P(x)* тождественно истинный предикат, а во всех остальных случаях это высказывание ложно. Высказывание $\exists x P(x)$ всегда истинно, кроме единственного случая, когда *P(x)* — тождественно ложный предикат.

Рассмотрим, например, предикат *P(x)* = «*x* — простое число», определенный на множестве натуральных чисел. Подставляя вместо *x* числа натурального ряда, получаем счетное множество высказываний. Некоторые из них, например *P* (1), *P* (2), *P* (3), *P* (5) и т. д., являются истинными. Высказывание $\forall x P(x)$ — «все натуральные числа простые» — ложно, а $\exists x P(x)$ — «некоторые из натуральных чисел — простые» — истинно.

Между кванторами $\forall x$ и $\exists x$ имеют место соотношения, обобщающие законы де Моргана: $\overline{\forall x P(x)} = \exists x \overline{P(x)}$; $\overline{\exists x P(x)} = \forall x \overline{P(x)}$.

3. Связанные и свободные переменные. Применение квантора к *n*-местному предикату превращает его в *(n*—1)-местный предикат. Кванторы можно также применять к нескольким различным переменным (по одному квантору какого-либо типа к каждой переменной). Если к *n*-местиому предикату применяется *k* кванторов, то он превращается в *(n — k)*-местный предикат, а при *n = k* — в высказывание. Переменные, к которым применяются кванторы, называются *связанными,* а остальные переменные — *свободными.* Например, из двухместного предиката *P(x, y)* с помощью кванторов получаем одноместные предикаты $\forall x P(x, y)$; $\exists x P(x, y)$; $\forall y P(x, y)$ и $\exists P(x, y)$, а также высказывания $\forall x \forall y P(x, y)$; $\forall x \exists y P(x, y)$; $\exists x \exists y P(x, y)$ и т. п.

Порядок следования одноименных кванторов не имеет значения, но разноименные кванторы переставлять нельзя. Так, $\forall x \forall y P(x, y)$ эквивалентно $\forall y \forall x P(x, y)$, но высказывания $\forall x \exists y P(x, y)$ и $\exists y \forall x P(x, y)$, вообще говоря, различны. В этом можно убедиться на примере предиката *P(x, y)* = «*x* делит *y*», который в первом случае превращается в высказывание «для всякого *x* существует такое *y,* что *x* делит *y*» (истинно), а во втором — «существует такое *y,* что любое *x* делит *y*» (ложно).

Квантор связывает переменную в области своего действия. Эта область обычно заключается в скобки, если она содержит не один предикат, а совокупность предикатов, связанных символами логических операций. Выражения, которые можно образовать применением к предикатам сентенциональных связок и кванторов, представляют собой *формулы логики предикатов.* Переменная *свободна в формуле,*

если хотя бы на одно ее вхождение не распространяется действие квантора. Переменная *связана в формуле,* если она связана по меньшей мере одним квантором. Например, в формуле $\exists y \forall x P(x, y) \rightarrow \forall z Q(z))$ вхождение каждой из переменных связано, а в формуле $\forall x(P(x, y) \vee \exists y Q(y)) \vee R(x)$ переменная x одновременно и свободная и связанная.

4. Категорические высказывания. Перевод предложений с русского или какого-либо другого языка на символический язык логики предикатов вызывает определенные трудности из-за отсутствия механических правил. Он основан не столько на форме обычных предложений, сколько на выявлении их смысловой связи.

В традиционной логике большое внимание уделяется четырем типам *категорических высказываний,* которые обычно обозначаются заглавными латинскими буквами *A, E, I, O:*

A — общеутвердительное высказывание «Всякое *S* суть *P*»: $\forall x(S(x) \rightarrow P(x))$, что означает: «Для всех *x,* если *x* обладает свойством *S,* то *x* обладает и свойством *P*»;

E — общеотрицательное высказывание «Никакое *S* не есть *P*»: $\forall x(S(x) \rightarrow \overline{P(x)})$, чю означает «Для иссх *x,* если *x* обладает свойством *S,* то он не обладает свойством *P*»;

I — частноутвердительное высказывание «Некоторые *S* суть *P*»: $\exists x(S(x) \wedge P(x))$, что означает «Существует такой объект *x,* обладающий свойством *S,* который также обладает свойством *P*»;

O — частноотрицательное высказывание «Некоторые *S* не суть *P*»: $\exists x(S(x) \wedge \overline{P(x)})$, что означает «Существует такой объект *x,* который обладает свойством *S* и не обладает свойством *P*».

Пусть, например, *S(x) — «x* — селедка» (свойство «быть селедкой») и *P(x) = «x* — рыба» (свойство «быть рыбой»). Тогда четырем типам категорических высказываний соответствуют следующие утверждения: *A* = «Всякая селедка— рыба»; *E* = «Никакая селедка не является рыбой»; *I* = «Некоторые селедки — рыбы»; *O* = «Некоторые селедки не являются рыбами».

На основе правил преобразования высказываний (5.6.7) и зависимостей между кванторами (2) можно записать: $\forall x (S(x) \rightarrow P(x)) \leftrightarrow$ $\leftrightarrow \overline{\exists x (\overline{S(x)} \vee P(x))} \leftrightarrow \overline{\exists x} (S(x) \wedge \overline{P(x)})$. Аналогично преобразуются и другие типы высказываний, в результате чего получаем зависимости:

$$\forall x\,(S(x) \to P(x)) \leftrightarrow \overline{\exists} x\,(S(x) \wedge \overline{P(x)});$$
$$\forall x\,(S(x) \to \overline{P(x)}) \leftrightarrow \overline{\exists} x\,(S(x) \wedge P(x));$$
$$\overline{\forall} x\,(S(x) \to \overline{P(x)}) \leftrightarrow \exists x\,(S(x) \wedge P(x));$$
$$\overline{\forall} x\,(S(x) \to P(x)) \leftrightarrow \exists x\,(S(x) \wedge \overline{P(x)}).$$

Как видно из приведенных равносильностей, высказывания *A* и *O*, а также *E* и *I* являются отрицаниями друг от друга (если одно из них истинно, то другое ложно и обратно) и называются *противоположными.* Из коммутативности операции конъюнкции следует, что суждения *E* и *I* допускают перестановку предикатов *S(x)* и *P(x),* т. е.

$$\overline{\exists} x\,(S(x) \wedge P(x)) \leftrightarrow \overline{\exists} x\,(P(x) \wedge S(x));$$
$$\exists x\,(S(x) \wedge P(x)) \leftrightarrow \exists x\,(P(x) \wedge S(x)).$$

5. Непосредственные заключения. Приняв одно из категорических высказываний в качестве посылки, а другое — в качестве следствия, можно построить так называемые *непосредственные заключения.* Истинность или ложность заключения зависит только от его формы или, как говорят, от его *модуса.*

Обычно категорические высказывания сокращенно обозначают совокупностью трех букв *SaP, SeP, SiP, SoP,* где *a, e, i, o* указывают на тип высказывания *(A, E, I, O*); *S* и *P —термины,* означающие свойства (в таком порядке, в каком они входят в высказывание). Например, непосредственное заключение $\forall x\,(S(x) \to P(x)) \to \exists x\,(P(x) \wedge S(x))$ в принятых обозначениях запишется как *SaP →SiP.*

Простой анализ показывает, что *SiP* является логическим следствием *SaP,* а *SoP* — логическим следствием *SeP.* Высказывания *SaP* и *SeP* могут одновременно быть ложными, но не истинными и поэтому называются *противоречивыми.* Высказывания *SiP* и *SoP* могут быть одновременно истинными, но не ложными и поэтому называются *антипротиворечивыми.*

Традиционная схема отношений между категорическими высказываниями, называемая *логическим квадратом,* показана на рис. 1.

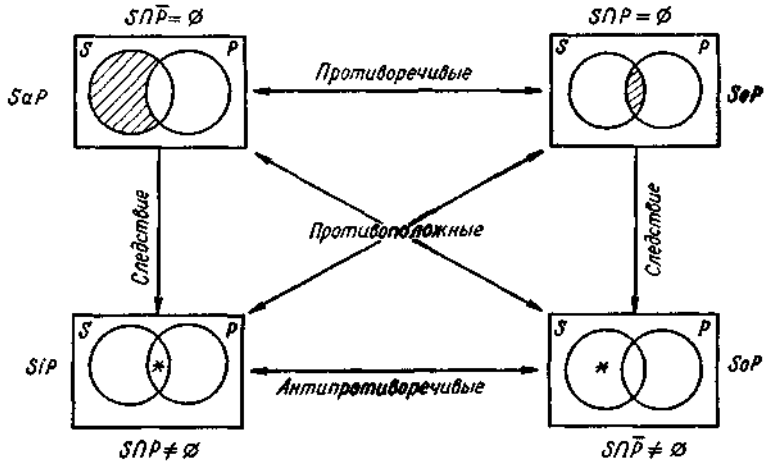


Рис. 1. Логический квадрат.

Там же приведены диаграммы Венна для каждого из четырех типов высказываний. Они непосредственно вытекают из правых частей выражений в (4) и теоретико-множественной интерпретации логических операций над предикатами, причем заштрихованные области соответствуют пустым множествам, а отмеченные звездочкой (*) — непустым множествам. Так как $S \cap \bar{P} = \varnothing$, если и только если $S \subset P$, то высказывание *SaP* соответствует отношению включения множеств $S \subset P$. В случае высказывания *SeP* множества *S* и *P* являются непересекающимися, а в случае высказывания *SiP* множества *S* и *P* должны иметь непустую общую часть. Наконец, высказывание *SoP* в силу тождества $S \cap \bar{P} = S \setminus P$ соответствует дополнению *S* до *P*.

Поскольку каждый из четырех типов высказываний может быть как посылкой, так и следствием, то всего можно образовать 4 • 4 = 16 модусов непосредственных заключений с одинаковым положением терминов *S* и *P* в посылках и следствиях. Изменив порядок следования терминов в следствиях *(SP* на *PS),* получим еще 16 модусов. Итак, имеется всего 32 существенно различных модусов непосредственных заключений. Анализ (например, с помощью диаграмм Венна) показывает, что только 10 из них являются тавтологиями, т. е. *правильными модусами.* Кроме четырех модусов, в которых посылки и следствия являются одинаковыми высказываниями, и двух модусов, допускающих перестановку терминов ($SeP \Rightarrow PeS$, $SiP \Rightarrow PiS$), правильными являются модусы:

$$SaP \Rightarrow SiP;\ SeP \Rightarrow SoP;\ SaP \Rightarrow PiS;\ SeP \Rightarrow PoS.$$

К таким выводам приходим, если, следуя традиционной формальной логике, считать, что термины *S* и *P* всегда соответствуют непустым множествам, т. е. предикаты *S(x)* и *P(x)* не могут быть тождественно ложными. Если же, например, *S(x)* — тождественно ложно *(S* = 0), то высказывания *SaP* и *SeP* всегда истинны, а *SiP* и *SoP* — ложные (это хорошо видно из рис. 1). Тем самым нарушается правильность ряда модусов традиционной логики.

Пусть, например, *S(x)* = *«x* — летающие черепахи», а *P(x)* означает «жить в зоопарке». Тогда категорические высказывания четырех типов суть следующие: *SaP* — «Все летающие черепахи живут в зоопарке», *SeP* = «Никакие летающие черепахи не живут в зоопарке», *SiP* = «Некоторые летающие черепахи живут в зоопарке», *SoP* = «Некоторые летающие черепахи не живут в зоопарке». Первые два высказывания истинны, что ясно из их эквивалентного представления: *SaP* = «Не существует такого объекта *x,* который был бы летающей черепахой и не жил в зоопарке» и *SeP* = «Не существует такого объекта *x,* который был бы летающей черепахой и жил в зоопарке». Истинность этих высказываний следует уже из того, что действительно «не существует такого объекта, который был бы летающей черепахой», т. е. в силу тождественной ложности предиката *S(x).* По этой же причине ложными являются два других высказывания *SiP* и *SoP*.

Ясно, что при тождественно ложном *S(x)* высказывание *I* не является следствием *A* и высказывание *O* не является следствием *E,* т. е. модусы $SaP \Rightarrow SiP;\ SeP \Rightarrow SoP;\ SaP \Rightarrow PiS;\ SeP \Rightarrow PoS$ перестают быть правильными. Теряют смысл и некоторые отношения между высказываниями, изображенные на логическом квадрате. Традиционная логика находит выход из этого положения, не допуская тождественно ложных предикатов, а значит, и пустых множеств. Но современная логика предикатов не может встать на такую точку зрения, которая сильно сузила бы область ее применения. О целесообразности допущения пустых множеств уже говорилось. Рассматривая пустое множество как подмножество любого множества, мы не нарушаем теоретико-множественных соотношений для различных типов категорических высказываний (рис. 1).

6**. Категорические силлогизмы**. Так называют суждения типа *XY*→Z, где *X, Y* и *Z* — категорические высказывания. Из истинности конъюнкции *XY* (она истинна только при истинных *X* и *Y*) на основании modus ponens можно выводить истинность высказывания *Z.* Если $\models XY \rightarrow Z$, то $XY \Rightarrow Z$ — *правильный силлогизм.*

Во всяком силлогизме *X* — *большая посылка,* содержащая термины *M* и *P; Y* — *малая посылка,* содержащая термины *M* и *S*, и Z — *заключение,* в котором *S* играет роль *подлежащего* и *P* — *сказуемого.* Таким образом, в силлогизме участвуют три термина, называемые: *S* — *малый термин, M* — *средний термин* и *P* — *большой термин,* причем некоторое суждение от *S* и *P* выводится из двух высказываний — посылок, в которых участвует средний термин *M,* отсутствующий в заключении. Например, $MaP \cdot SaM \rightarrow SaP$ означает: «Если все *M* суть *P* и все *S* суть *M,* то все *S* суть *P»,* что принято записывать в виде:

$$\begin{array}{l} \text{Все } M \text{ суть } P \\ \text{Все } S \text{ суть } M \\ \hline \text{Все } S \text{ суть } P \end{array}$$

В зависимости от порядка следования терминов в посылках совокупность силлогизмов распадается на четыре группы, называемые *фигурами* силлогизмов:

$$\begin{array}{l} MP \\ SM \\ \hline SP \end{array}; \quad \begin{array}{l} PM \\ SM \\ \hline SP \end{array}; \quad \begin{array}{l} MP \\ MS \\ \hline SP \end{array}; \quad \begin{array}{l} PM \\ MS \\ \hline SP \end{array}.$$

В данной фигуре каждое из высказываний может относиться к одному из четырех типов *A, E, I, O,* поэтому из нее можно образовать $4^3 = 64$ модуса, а общее количество модусов для всех четырех фигур равно $64 \cdot 4 = 256$. Основная задача теории силлогизмов состоит в выделении множества правильных модусов, т. е. таких, которые при любых конкретных терминах позволяют из истинных посылок делать истинные заключения. Можно доказать, что из 256 модусов правильными являются только 15. Для наименования правильных модусов применяются слова, содержащие три из четырех букв *a, e, i, o*, которые указывают последовательно на типы высказываний посылок и заключения. Они выглядят (по фигурам) следующим образом:

	Barbara	Celarent	Darii	Ferio
1)	MaP SaM $\overline{SaP}$	MeP SaM $\overline{SeP}$	MaP SiM $\overline{SiP}$	MeP SiM $\overline{SoP}$
	Cezare	Camestres	Festino	Baroco
2)	PeM SaM $\overline{SeP}$	PaM SeM $\overline{SeP}$	PeM SiM $\overline{SoP}$	PaM SoM $\overline{SoP}$
	Datisi	Feriso	Disamis	Bocardo
3)	MaP MiS $\overline{SiP}$	MeP MiS $\overline{SoP}$	MiP MaS $\overline{SiP}$	MoP MaS $\overline{SoP}$
	Calemes	Fresison	Dimatis	
4)	PaM MeS $\overline{SeP}$	PeM MiS $\overline{SoP}$	PiM MaS $\overline{SiP}$	

Традиционная логика признавала правильными еще девять модусов, которые имеют место при условии, что терминам соответствуют непустые множества объектов. Правильность модусов доказывается на основе законов логики высказываний. Так, для модуса Celarent имеем:

$$\forall x\,(M(x) \to \overline{P(x)}) \quad \forall x\,(S(x) \to M(x)) \to \forall x\,((S(x) \to M(x)) \wedge (M(x) \to \overline{P(x)})).$$

Согласно закону силлогизма $((A \to B)(B \to C)) \Rightarrow (A \to C)$, если для всякого x выражение $(S(x) \to M(x))(M(x) \to \overline{P(x)})$ истинно, то истинно и выражение $S(x) \to \overline{P(x)}$. Таким образом, в сокращенной записи имеем $MeP \cdot SaM \Rightarrow SeP$, что и представляет собой силлогизм Celarent. Аналогично доказываются и другие правильные силлогизмы. Придавая терминам *S, M, P* конкретное содержание, из истинных посылок всегда будем получать истинные заключения. Например, в соответствии с модусом Festino имеем:

Никакие черепахи не летают
Некоторые животные летают
Некоторые животные — не черепахи

В то время как правильность модуса требует строгого доказательства, для установления неправильности какого-либо модуса достаточно привести опровергающий его контрпример. Так, модус $MaP \cdot MiS \to SaP$ опровергается ложным суждением:

Всякое четное число делится на 2
Некоторые четные числа — простые
Всякое простое число делится на 2

Правильный вывод из приведенных посылок «Некоторые простые числа делятся на 2» (множество таких простых чисел содержит единственный элемент 2) следует в соотвегствии с модусом Datisi.

7. Символизация языка. Логика предикатов располагает более общими и универсальными методами обоснования правильных выводов, чем формальная логика. Первым этапом построения какого-либо доказательства или теории является символизация исходных положении, подвергающихся логическому анализу или принимаемых в качестве аксиом дайной теории. Этот процесс обычно сводится к переводу некоторых высказываний на символический язык логики предикатов. Приведем некоторые примеры.

Рассмотрим сложное высказывание, выраженное на обычном языке: «Некоторые студенты выполнили все задания. Ни один студент не выполнял графиков. Следовательно, ни одно задание не являлось графиком». В первом предложении участвуют одноместные предикаты — свойства *P(x)* = «*x* — студент», *Q(y)* = «*y* — задание» и двуместный предикат *R(x, y)* = «*x* — выполнил *y*». Так как в нем говорится о «некоторых студентах», то соответствующая форма будет $\exists x(P(x) \wedge A(x))$, где *A(x)* — сложное высказывание, характеризующее предикат *P(x),* а именно: «выполнили все задания». Поскольку речь идет о «всех заданиях», то переменная *y* связывается квантором общности и высказывание *A(x)* представляется формулой $\forall y(Q(y) \to R(x, y))$, которая дословно переводится «для всякого *y,* если *y* — задание, то *x* выполнили *y*», смысл которого соответствует фразе «выполнили все задания». Итак, символическая запись первого предложения имеет вид: $\exists x(P(x) \wedge \forall y(Q(y) \to R(x, y)))$.
Аналогично записывается и второе предложение

$$\forall x(P(x) \to \forall y(S(y) \to \overline{R(x, y)})),$$

где *S(y)* = «*y* — график». Заключение «Ни одно задание не являлось графиком» представляет собой категорическое высказывание типа *QeS.* Таким образом, получаем окончательно:

$$\exists x(P(x) \wedge \forall y(Q(y) \to R(x, y))) \wedge \forall x(P(x) \to$$

$$\to \forall y(S(y) \to \overline{R(x, y)})) \to \forall x(Q(x) \to \overline{S(x)}).$$

Рассмотрим примеры символической записи свойств и определений Пусть *P(x, y)* — бинарное отношение, определенное на некотором

множестве *M.* Рассматривая его как двуместный предикат, записываем основные свойства отношений: $\forall x P(x, x)$ — рефлексивность, $\forall x \forall y (P(x, y) \to P(y, x))$ — симметричность, $\forall x \forall y \forall z \times$ $\times (P(x,y) \wedge P(y, z) \to P(x, z))$ транзитивность, $\forall x \forall y (P(x, y) \wedge$ $\wedge P(y, x) \to (x = y))$ — антисимметричность и т. д. С помощью этих и подобных им выражений определяются любые типы бинарных отношений, обладающих некоторой совокупностью свойств. Так, отношение эквивалентности определяется как двуместный предикат, удовлетворяющий формуле: $\forall x P(x, x) \wedge \forall x \forall y (P(x, y) \to$ $\to P(y, x)) \wedge \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \to P(x, z))$.

Символический язык логики предикатов широко используется в теории распознавания. Поэтому необходимо научиться уверенно расшифровывать формулы, записанные на этом языке. Пусть, например, $\forall x (P(x) \to \exists y (Q(y) \wedge R(x, y)))$, где *P(x)=«x* — простое число», *Q(x) = «x* — четное число», *R(x, y) = «R* делится на *x».* Это общеутвердительное высказывание, в котором *P(x)* играет роль подлежащего, а $\exists y (Q(y) \wedge R(x, y))$ — сказуемого. В свою очередь, сказуемое является частноутвердительным высказыванием относительно переменной *y (x* — свободная переменная) и означает: «Существует такое четное число *y,* которое делится на *x».* Тогда исходная формула расшифровывается следующим образом: «Для всех *x,* если *x* — простое число, существует такое чатное число *y,* которое делится на *x*» или проще: «Для всякого простого числа можно подыскать такое четное число, которое делится на это простое число»

8. Оценочная процедура Истипное значение формулы в логике предикатов можно установить с помощью *оценочной процедуры.* Она сводится к определению значений входящих в данную формулу предикатов при замещении свободных переменных элементами из множества их определения. При этом последовательно используются общие свойства сентенциональных связок и кванторов. Исходными данными являются неоднородные функции, представляющие предикаты, и конкретные значения высказываний и свободных переменных, для которых требуется найти значение формулы

Проиллюстрируем рассматриваемую процедуру на примере формулы $\forall x (P(x, y, z) \to \exists y Q(x, y)) \vee Q(x, y) \bar{S}$, где предикаты заданы на двухэлементном множестве $\{a, b\}$ таблицами соответствия:

$P(x, y, z)$

y	a	a	b	b
z	a	b	a	b
x a	0	1	1	0
x b	0	1	0	1

$Q(x, y)$

y	a	b
a	0	0
b	0	1

Пусть $S = 0$; $x = b$; $y = a$; $z = a$. Подставляя эти значения в формулу, получаем $\forall x(P(x, a, a) \to \exists y Q(x, y)) \vee Q(b, a) \cdot 1$. Так как $Q(b,a) = 0$, то формула упрощается к виду $\forall x(P(x, a, a) \to \exists y Q(x, y))$. Это выражение представляет собой высказывание, для установления значения которого необходимо выяснить, является ли одноместный предикат в скобках истинным для всех значений x. Соответствующая таблица имеет вид:

x	$P(x, a, a) \to \exists y Q(x, y)$	
a	0	0
b	0	1

Здесь значения $P(x, a, a)$ взяты из первого столбца таблицы для $P(x, y, z)$. Значения $\exists y Q(x, y)$ получены на основе таблицы для $Q(x, y)$. Так как первая ее строка содержит только нули, то $\exists y Q(x, y)$ при $x = a$ получает значение 0. Во второй строке имеется единица, откуда заключаем, что $\exists y Q(x, y)$ при $x = b$ имеет значение 1. Истинностные значения выражения $P(x, a, a) \to \exists y Q(x, y)$ помещены в таблице под знаком импликации (так часто поступают для сокращения места).

Как видим, это выражение тождественно истинно относительно переменной x, следовательно, $\forall x(P(x, a, a) \to \exists y Q(x, y))$ также истинно, т. е. исследуемая формула имеет значение 1. Аналогично можно определить истинностные значения формулы и при других значениях переменных x, y, z и высказывания S. Рассмотренная процедура трудоемка даже для сравнительно простых формул, особенно, если требуется найти истинностные значения на всевозможных наборах (при этом необходимо выполнить эту процедуру для всех функций $P(x, y, z)$ и $Q(x, y)$).

9. Общезначимость. Особый интерес представляют общезначимые формулы, которые истинны (принимают значения 1) при каждом приписывании значений входящих в них свободных переменных и предикатов. Если A — общезначимая формула, то она, как и тавтологии, обозначается $\models A$.

Для доказательства общезначимости формул используется аппарат логики высказываний, дополненный теоремами для выражений, содержащих кванторы. Приведем некоторые из них.

1) Пусть $Q(x)$ — формула, свободная для y; тогда: а) $\models \forall x Q(x) \to \to Q(y)$; б) $\models Q(y) \to \exists x Q(x)$;

2) Пусть R — формула, не содержащая свободных вхождений переменной x, и $Q(x)$ — какая-либо формула; тогда: а) если $\models R \to \to Q(x)$, то $\models R \to \forall x Q(x)$; б) если $\models Q(x) \to R$, то $\exists x Q(x) \to R$.

3) $\models Q(x)$, если и только если (следствие $\models \forall x Q(x)$ из теорем 1 и 2).

На основе этих теорем строятся правила вывода, которые, наряду с правилами исчисления высказываний (правила подстановки и заключения, теорема дедукции и др.). используются для доказательства логических следствий.

Правило универсальной конкретизации (УК): из $\forall x Q(x)$, которая свободна для y, выводится $Q(y)$ подстановкой в $Q(x)$ вместо x переменной y (теорема 1 а).

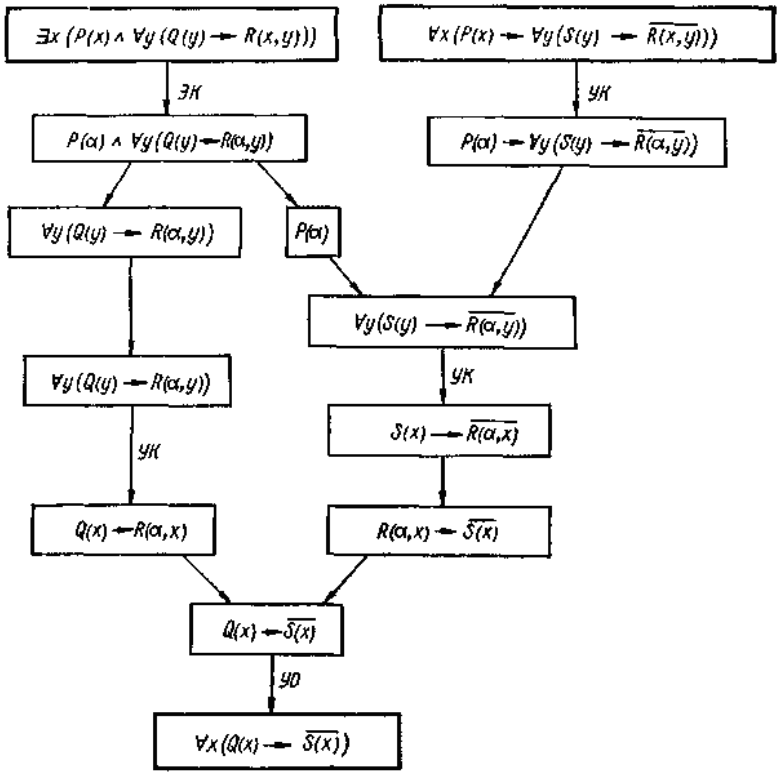


Рис. 2. Диаграмма вывода $\forall x (Q(x) \to \overline{S(x)})$ из посылок

$$\exists x\,(P(x) \wedge \forall y\,(Q(y) \to R(x,\ y))) \text{ и}$$
$$\forall x\,(P(x) \to \forall y\,(S(y) \to \overline{R(x,\ y)})).$$

Правило универсального обобщения (УО): если *Q(x)* — следствие посылок, ни одна из которых не имеет свободных вхождений *x,* то из нее выводится у *xQ(x)* (теорема 2 а).

Кроме того, можно использовать еще два правила, представляющие собой аналоги приведенных выше правил для квантора существования.

Правило экзистенциальной конкретизации (ЭК) позволяет перейти от $\exists\, xP(x)$ к *P((x),* где α — неизвестный, но вполне определенный элемент такой, что, если $\exists\, xP(x)$ истинно, то *P(a)* также истинно.

Правило экзистенциального обобщения (ЭО) позволяет перейти от *P*(α) к $\exists\, xP(x)$, т. е., если существует такое α, что *P(α)* истинно, то истинно и $\exists\, xP(x)$.

В логику предикатов полностью переносятся все тавтологии, в частности соотношения: а) $| = A \sim B$, если и только если $A \Leftrightarrow B$; б) $| = A \to B$, если и только если $A \Rightarrow B$.

10. Доказательство логического следствия. Исходя из понятия общезначимости, можно дать следующее определение *логического следствия в логике предикатов:* формула *B* есть логическое следствие формул $A_1, A_2, \ldots, A_m$, т. е. $A_1, A_2, \ldots, A_m \Rightarrow B$, если для каждого множества определения и для каждого приписывания формулам A_i $(i = 1, 2, \ldots, m)$ в этом множестве формула *B* истинна при условии, что все A_i истинны. При этом для всех свободных вхождений некоторой переменной *x* в какие-нибудь A_i выбирается одно и то же значение *x* из множества определения, т. е. такое *x* по существу рассматривают как постоянную.

Следуя общей схеме рассуждений, изложенной и (5.6. 10), а также дополнительным правилам вывода (9), рассмотрим пример из (7), где $\exists\, x(P(x) \wedge \forall\, y(Q(y) \to \overline{R(x,\ y)}))$ и $\forall\, x\,(P(x) \to \forall\, y(S(y) \to \to \overline{R(x, y)}))$ — посылки и $\forall\, x(Q(x) \to \overline{S(x)})$ — заключение. Процесс доказательства представляется диаграммой, показанной на рис. 2. Применение правил вывода, специфических для логики предикатов, указано здесь их сокращенными обозначениями. Остальные правила заимствованы из логики высказываний.

5.9. Формальное описание и преобразование распознающих процессов

5.9.1. Исчисление высказываний как язык описания распознающих процессов

Развитие методов построения логических схем позволило создать методы построения систем автоматизированных распознающих систем (АРС). Рассмотрим аппарат математической логики, открывший широкие возможности для построения схем автоматизированных распознающих процессов.

Основные законы алгебры логики

Алгебра логики подчиняется законам, иногда совпадающим с законами обычной алгебры, а иногда — своим своеобразным законам. Рассмотрим основные законы алгебры логики.

Законы множеств:

$$0 \cdot a = 0;$$
$$0 + a = a;$$
$$0 \cdot abc \ldots w = 0,$$

т. е. произведение любого числа переменных обращается в нуль, если какая-либо одна переменная имеет значение 0, независимо от значений других переменных;

$$1 \cdot a = a;$$
$$1 + a = 1;$$
$$1 + a + b + c + d + \ldots + w = 1,$$

т. е. сумма любого числа переменных обращается в единицу, если одна из ее переменных имеет значение 1, независимо от значений других переменных.

Законы перемещения:

$$ab = ba;$$
$$a + b = b + a,$$

т. е. результаты выполнения операций умножения и сложения не зависят от того, в каком порядке следуют переменные.

Законы тавтологии (повторения):

$$aa = a,$$
$$aaa \ldots = a^n = a,$$
$$a + a = a,$$
$$a + a + a + \ldots + a = na = a.$$

Здесь можно сказать, что истина или ложь всегда остается истиной (или ложью), сколько ее не повторяй.

Законы дополнительности:

а) логическое противоречие:

$$a\,\bar{a} = 0$$

т. е. произведение любой переменной и ее инверсии есть 0. Как пример можно привести строку из известной песни «Речка движется и не движется» — заведомая ложь,

б) закон исключенного третьего:

$$a + \bar{a} = 1,$$

т. е. сумма любой переменной и ее инверсии есть 1. Так, утверждая, что «студент сдаст экзамен или не сдаст», мы всегда будем правы.

Законы инверсии (Де Моргана):

$$\overline{ab} = \bar{a} + \bar{b},$$

т. е. инверсия произведения равна сумме инверсий;

$$\overline{a+b} = \overline{ab},$$

а инверсия суммы есть произведение инверсий.

Здесь записаны законы для двух переменных, но они справедливы для любого числа переменных.

Законы распределительные (дистрибутивные)}

а) произведения относительно суммы:

$$a(b + c) = ab + ac.$$

Справедливость этого закона можно подтвердить высказываниями. Например: «Я зайду за Вами *И* мы пойдем в театр *ИЛИ* в кино». Так можно формулировать левую часть приведенного выше выражения, а правая часть тогда может быть прочитана так: «Я зайду за Вами *И* мы пойдем в театр *ИЛИ* я зайду за Вами *И* мы пойдем в кино». Смысл один и тот же, но правая часть несколько длиннее:

б) суммы относительно произведения:

$$a + bc = (a + b)(a + c).$$

Справедливость этого закона можно доказать, опираясь на предыдущие. Раскрыв скобки, получим

$$a + bc = aa + ac + ba + be = a + ac + ba + bc.$$

Из первых двух членов вынесем за скобки переменную *a:*

$$a(1+c),$$

но

$$1 + c = 1, \text{ а } a \cdot 1 = a.$$

Рассматривая следующее выражение $ba + bc$, мы устанавливаем, что и оно равно a. Тогда вся правая часть превращается в $a + bc$, т. е. в такое же выражение, как и в левой части.

Законы склеивания:

$$ab + a\bar{b} = a;$$

$$(a + b)(a + \bar{b}) = a.$$

Эти законы легко подтверждаются на основании рассмотренных ранее законов, например:

$$ab + a\bar{b} = a(b + \bar{b}) — a \cdot 1 = a.$$

Законы поглощения:

$$a(a + b) = a;$$
$$a(a + b)(a + c) \ldots (a + w) = a;$$
$$a + ab = a;$$
$$a + ab + ac + .., + aw = a;$$
$$a(\bar{a} + b) = ab;$$
$$a + \bar{a}\, b = a + b.$$

Эти законы можно легко доказать с помощью других законов алгебры логики, например, умножая в последнем выражении первый член a на $(1 + b)$, получаем

$$a(1 + b) + \bar{a}\, b = a + ab + \bar{a}\, b = a + b\,(a + \bar{a}) = a + b \cdot 1 = a + b.$$

Так как для логического сложения и умножения характерны все свойства сложения и умножения алгебры чисел, то над многочленами алгебры высказываний можно производить те же действия, что и над многочленами алгебры чисел. Но логическое сложение и умножение обладают и некоторыми необычными свойствами и это приводит к необычности действий над логическими многочленами. Разъясним это на примере.

Пример. Пусть необходимо умножить $(a + b)$ на $(a+c)$. Умножаем по обычным правилам умножения многочлена на многочлен:

$$(a + b)(a + c) = aa + ac + ab + bc.$$

Так как в алгебре высказываний $aa = a$, то

$$(a + b)(a + c) = a + ac + ab + bc.$$

Но работу над полученным произведением можно продолжить. Рассмотрим два первых слагаемых a и ac. Сгруппируем их и общий множитель a вынесем за скобки:

$$a + ac = a(1 + c),$$

и далее

$$a \cdot 1 = a.$$

Также поступим и с суммой $a + ab = a$. Тогда окончательно

$$(a + b)(a + c) = a + bc.$$

Результат оказался проще, чем мы ожидали, так как выражение $a + ab$, согласно закону поглощения, заменили множителем a.

Таким образом, если высказывание логически складывают с логическим произведением, в состав которого оно входит, то оно поглощает это произведение. Отметим еще одну важную особенность алгебры высказываний. Если в формуле $a + ab$ заменить знак + на знак

× и знак × на знак +, то полученное новое высказывание $a(a + b)$ будет эквивалентно заданному. В этом легко убедиться, раскрыв скобки.

Это свойство распространяется на любую формулу алгебры высказываний.

Например,

$$a(b + c) = ab + ac.$$

К обеим частям применим упомянутую замену знаков и получим:

$$a + bc = (a + b)(a + c).$$

Еще пример. Дано $ab + a\bar{b}$. В левой части заменим знаки:

$$ab + a\bar{b} = (a + b)(a + \bar{b}) = a$$

или

$$(a + b)(a + \bar{b}) = a.$$

Эту особенность преобразования формул в алгебре высказываний называют законом двойственности. Опираясь на закон двойственности, легко преобразовать эквивалентные высказывания.

Упрощение логических выражений

Следует иметь в виду, что каждое логическое высказывание можно воплотить с помощью логических элементов в конкретный действующий автоматический механизм. Для этого каждое логическое сложение, т. е. знак плюс в формуле высказывания, следует осуществить логическим элементом *ИЛИ,* каждое логическое умножение —- элементом *И,* а каждое отрицание или инверсию — элементом *НЕ* (рис. 1).

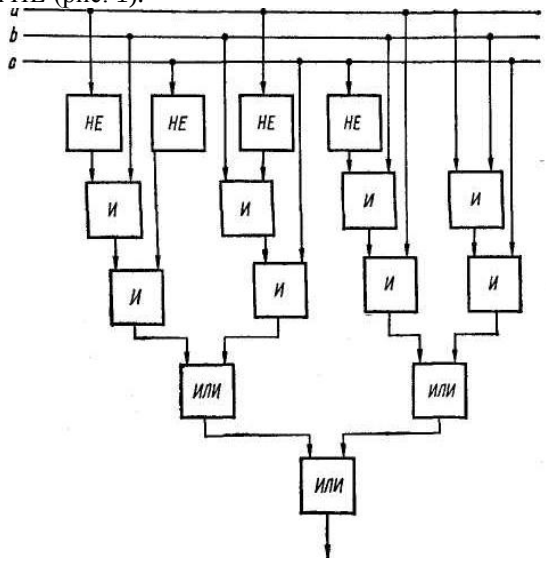


Рис. 1. Логическая схема.

Рассмотрим выражение

$$x = abc + \bar{a}\,bc + ab\,\bar{c} + \bar{a}\,b\,\bar{c}.$$

Заметим, что первое и третье, второе и четвертое слагаемые склеиваются по букве *a.*

Действительно,

$$abc + \bar{a}\,bc = bc$$

и

$$ab\,\bar{c} + \bar{a}\,b\,\bar{c} = b\,\bar{c}.$$

Имеем

$$x = bc + b\,\bar{c}.$$

В этой формуле можно произвести склеивание по букве *c* и тогда окончательно записанное ранее сложное выражение принимает очень простой вид: $x = b$.

Значит, математическая обработка выражения, построенного по законам математической логики, устранила необходимость в 15 логических элементах.

Мы произвели упрощение логического выражения, используя основные законы математической логики, путем последовательных рассуждений. Пример, который мы рассмотрели, относительно прост, да и то разные люди могли бы его решать по-разному и получать разные результаты. Тем более, такое явление может иметь место при упрощении более сложных логических выражений, в которых участвует большое число переменных, и которые выражаются более сложными зависимостями.

Задача упрощения логических выражений, или, как говорят, их «минимизация», является одной из наиболее сложных в алгебре логики. Есть много различных способов минимизации.

Рассмотрим один из распространенных способов минимизации логических выражений с помощью карт Карно.

Карты Карно (Karnaugh) наглядно изображают логические функции. Карта Карно (рис. 2) разделена на квадратики, и каждому из них отвечает определенная комбинация значений всех входных переменных. Кроме того, каждая сторона квадратика представляет собой границу между значениями переменных (верхний и нижний, равно как и боковые квадратики карты, являются соседними).

Обозначения входных переменных указываются сверху и сбоку карты и относятся ко всему столбику или строке квадратиков, причем значения этих входных переменных в них принимаются равными единице. В соседних с обозначенными в столбцах или строках входные переменные соответственно равны нулю. В квадратиках

записывается значение самой функции при данных комбинациях значений входных переменных. Значения входных переменных не принято записывать в квадратиках, они подразумеваются, поэтому на карте остается только значение функции (рис. 2, *в, г, д, е).* Из примеров, приведенных на рисунке для двух, трех и четырех переменных, видно, что прибавление каждой новой переменной удваивает карту.

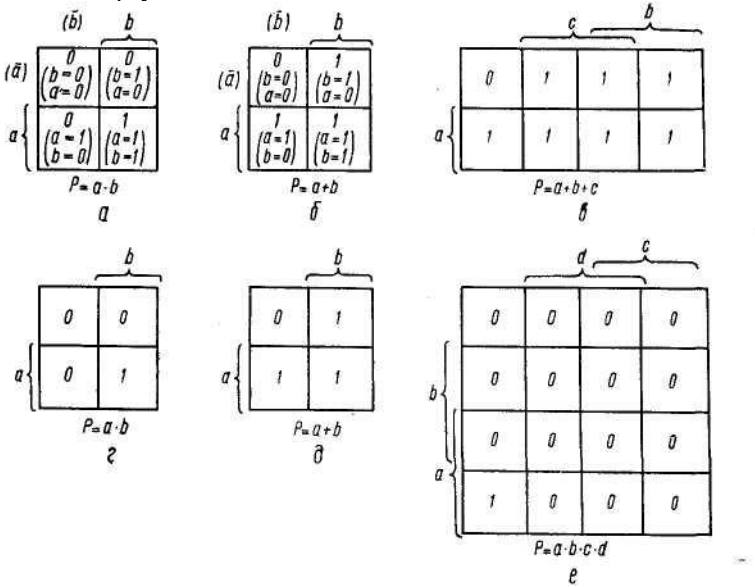


Рис. 2. Карты Карно.

Применяя карты Карно для изображения алгебраического выражения функций, можно записать функции либо в виде суммы произведений, либо в виде произведения сумм.

Выражение суммы произведений определяется суммой произведений значений всех входных переменных (прямых и инверсных) в каждом из квадратиков карты, содержащих единицу. Так, например, для карты, показанной на рис. 2, *а, г, P=ab,* а для карты, показанной на рис. 2, *б, д, —*$P = ab + a\bar{b}$.

Выражения сомножителей в произведении сумм определяются суммами инверсных значений входных переменных в каждом из нулевых квадратиков. Так, например, для карты, представленной на рис. 2, *а, в,*

$$P=(a + b)(a + \bar{b})(\bar{a} + b);$$

для карты, показанной на рис. 2, *б, д,*

$$P = a + b.$$

С помощью карт Карно можно получить упрощенное выражение функций, для чего определяют суммы произведений и произведения сумм, объединяя квадратики, в которых значения функции соответственно равны 1 или 0, в контуры. Последние должны иметь форму прямоугольников и содержать четное число квадратиков или только один квадратик.

Из свойств карт Карно вытекает, что при переходе контура из одного квадратика к другому одна из переменных инвертируется. Поэтому выражение контура из двух квадратиков не зависит от этой переменной, а определяется только остальными переменными, т. е. выражения, соответствующие контурам, «не содержат тех переменных, чьи границы пересекаются данным контуром». Так, контур, ограничивающий четыре квадратика, пересекает две границы двух переменных и поэтому соответствующее ему выражение содержит n—2 переменных и т. д.

Для получения наиболее простых выражений, реализуемых минимальным количеством возможно более простых логических выражений, т. е. при минимизации, логическое выражение должно иметь как можно меньше членов, каждый из которых должен содержать как можно меньше переменных.

Правила минимизации выражения логической функции по карте Карно сводятся к следующему. Чем большее число квадратиков с одним значением функции объединяется в общем контуре на карте и чем меньше будет таких контуров, тем проще будет аналитическое выражение функции. При этом все квадратики с одним значением функции должны входить в какой-нибудь контур. Нужно также следить за тем, чтобы какой-либо контур не входил полностью в другие контуры.

Рассмотрим подробнее карту Карно для трех переменных (рис. 3).

Для наглядности в квадратиках указаны значения переменных. Пусть функция должна быть заложена в карту Карно и минимизирована.

Подставив значения переменных, соответствующие левому верхнему квадратику, в выражение функции, получим

$$X = abc + ab\bar{c} + \bar{a}bc + \bar{a}b\bar{c} = 001 + 000 + 101 + 100 = 0 + 0 + 0 + 0 = 0.$$

Это значит, что значение сложного выражения в этом квадратике равно 0, что и записываем.

Определяя таким образом значение выражения для всех квадратиков, получаем карту, показанную на рис. 3, *б.*

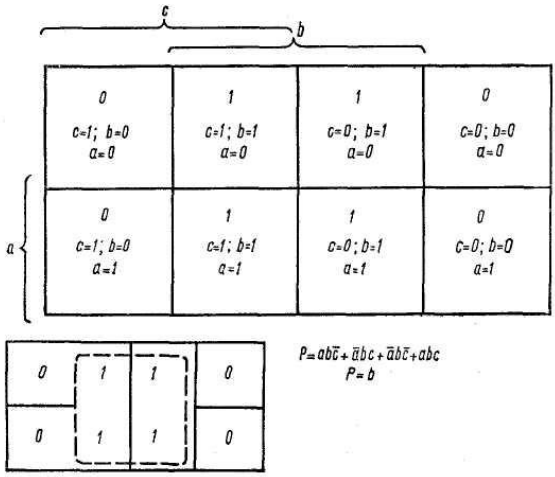


Рис. 3. Карта Карно для трех переменных.

Охватываем контуром средние четыре квадратика со значениями 1. В этом контуре только лишь переменная сохраняет свое значение, равное единице во всех квадратиках. Следовательно, результат минимизации определяется выражением

$$X = b.$$

Такое же значение мы получили ранее более сложным приемом минимизации — аналитическим путем с использованием основных законов алгебры логики,

Разница в затратах труда на минимизацию становится тем значительнее, чем больше переменных в логическом выражении и чем оно сложнее.

Схемы атоматизированных распознающих систем основаны на так называемых двухпозиционных приборах, т. е. устройствах, способных занимать только одно из двух устойчивых положений. Сигнал, поступающий на вход в систему или снимаемый с выхода системы, может либо присутствовать (1), либо отсутствовать (0).

Поэтому в дальнейшем под переменными будем понимать сигналы на входе в схему, а под сложными высказываниями — сигналы на выходе, являющиеся логическими функциями этих переменных. Задача логической части схемы — выработать сигналы на выходе, являющиеся логическими функциями сигналов на входах.

Для решения задачи составления схемы необходимы электрические, электронные или пневматические устройства, осуществляющие элементарные логические связи *И, ИЛИ, НЕ.* Такие устройства

называются элементами. Допустим, что наличию сигнала соответствует 1, т. е, истина, а отсутствию сигнала — 0, т. е. ложь. В электрических системах истине соответствует подача тока, а отсутствие тока — лжи. В пневматических системах наличие сигнала означает подачу сжатого воздуха под давлением, а отсутствие сигнала - соединение с атмосферой.

Логический элемент типа *И* должен иметь два или больше входа и один выход, с которого сигнал снимается.

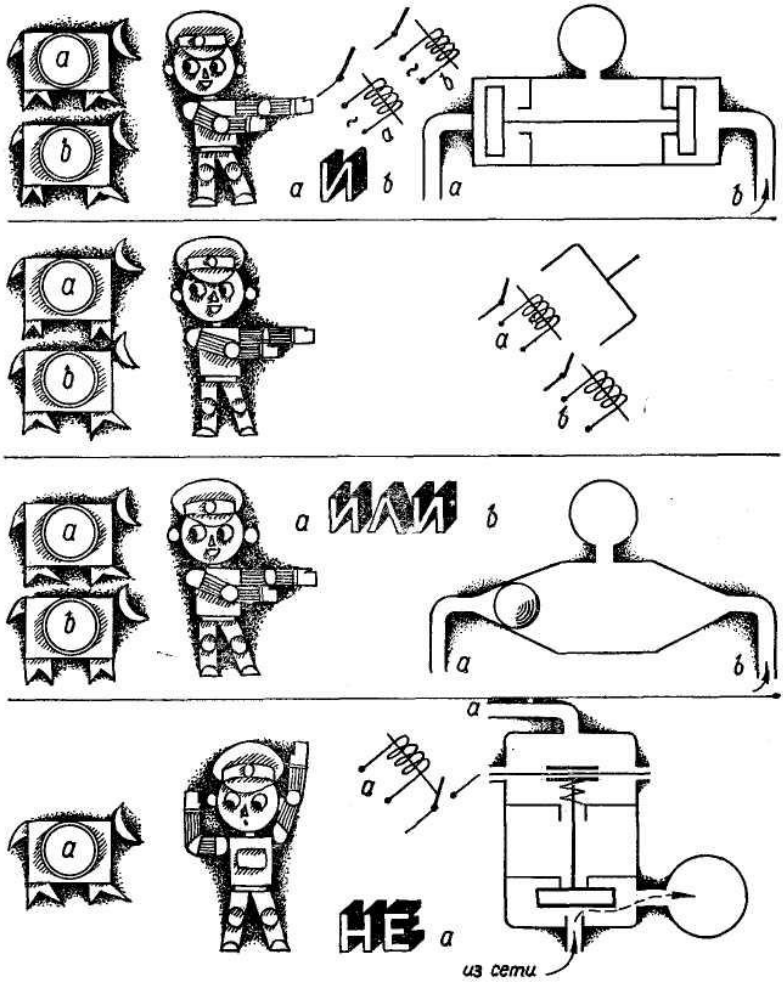


Рис. 4. Логические элементы И, ИЛИ, НЕ.

В электрическом устройстве, в котором два реле *a* и *b* включены последовательно (рис. 4), сигнал на выходе появится при подаче напряжения на катушки обоих реле. В этом случае элемент выполнит логическую операцию умножения $P = ab$.

В пневматическом устройстве сигналы, т. е. сжатый воздух, поступают от двух пневматических кнопок *a* и *b*. Если нажать на одну из кнопок и подать воздух под давлением на один из входов, то две тарели клапана, сидящие на одной оси, передвинутся в одну сторону и свяжут выход клапана с атмосферой через вторую кнопку. Если нажать обе кнопки, то независимо от того, какое положение займут тарели, сжатый воздух пройдет на выход.

Элемент *ИЛИ* также имеет два входа и один выход. Если два реле *a* и *b* соединены параллельно, то сигнал на выходе появится, если подать напряжение на любое реле, и тогда элемент выполняет операцию логического сложения $P=a + b$.

В пневматическом варианте, если обе кнопки отпущены, выход элемента связан с атмосферой по крайней мере через одну из кнопок. Нажмем, например, кнопку *a*. Под действием давления сжатого воздуха шарик переместится вправо, прижмется к резиновому кольцу и не даст воздуху выходить в атмосферу через кнопку *b*, вследствие чего воздух поступит на выход клапана—элемента. Если отпустим кнопку *a* и нажмем кнопку *b*, то шарик переместится влево, не давая воздуху выходить в атмосферу через кнопку *a*, и на выходе тоже появится сигнал. Нажмем обе кнопки вместе, и, в каком бы ни был положении шарик, появится сигнал на выходе.

Логическим элементом *НЕ* в релейном варианте является переключатель. Когда напряжение в катушке *a* отсутствует, на выходе *P* протекает ток, т. е. имеется сигнал на выходе. В пневматическом варианте этот элемент выглядит несколько сложнее; оно носит название пневматического реле. Полость над мембраной *1*, с которой соединяется пневматическая кнопка *a*, является полостью управления реле: она же представляет собой вход. Если кнопка *a* отпущена, т. е. сигнал на входе отсутствует, то тарель *6* с толкателем *3* под действием пружины *2* находится в положении, показанном на рисунке. Резиновая прокладка *5* прижимается к верхнему седлу корпуса и закрывает проход к отверстию *4*, ведущему в атмосферу. Сжатый воздух, подводимый к нижнему отверстию 7, свободно проходит на выход реле *8*. При отсутствии сигнала на входе имеется сигнал на выходе.

Теперь нажмем кнопку *a*. Сжатый воздух начнет поступать в полость управления реле. Под действием силы давления воздуха мембрана прогибается, и тарель с толкателем движутся вниз, сжимая пружину. Резиновая прокладка отходит от верхнего седла, освобождая проход к

отверстию *4,* а другая прокладка *6* перекрывает нижнее отверстие 7. Выход реле *8* оказывается связанным с атмосферой через отверстие *4.* Имеется сигнал на входе — нет сигнала на выходе.

Логические элементы, из которых складываются логические системы, чаще всего основываются на применении электронных, пневматических, пневмонических схем. В обычных пневматических устройствах процессы совершаются во много раз медленнее, чем в электронных, и поэтому логические машины на пневматике работают значительно медленнее. Однако применение пневмоники разрешает создавать устройства на сжатом воздухе, способные выполнять до двух тысяч и более операций в секунду. Вместе с тем при решении целого ряда задач автоматизации производства и, очень часто, в машиностроении, большая скорость выполнения отдельных операций вообще и не требуется. Зато пневматические устройства имеют ряд качеств, выгодно отличающих их от электронных. Они по самой своей природе взрывобезопасны, просты и надежны. Для их обслуживания и ремонта не требуется высокой квалификации.

Пример использования языка исчисления высказывний

При решении задачи распознавания на основе языка исчисления высказывний, в АРС необходимо последовательно выполнить ряд следующих приемов:

1) составить подробные требования к процессам распознавания;

2) установить число входов и выходов распознаваемого объекта;

3) составить таблицу функционирования распознаваемого объекта по форме;

4) на основании таблицы составить структурную формулу модели распознаваемого объекта;

5) осуществить минимизирование логической функции, т. е. структурной формулы модели распознаваемого объекта;

6) составить функциональную схему распознаваемого объекта по минимизированной логической функции.

Когда функциональная схема составлена, можно считать, что задача по распознанию объекта будет решена.

Рассмотрим пример. На одном заводе имеются три цеха *А, В* и *С.* Электроэнергией их обеспечивает небольшая электрическая станция, на которой установлено два электрогенератора *X* и *Y.* Мощность генератора *X* в два раза выше, чем генератора *Y.*

Если в энергии нуждается один из цехов, то достаточно включить генератор *Y,* если же любые два цеха — генератор *X.* Снабжение электроэнергией всех трех цехов сразу обеспечивает одновременная работа двух генераторов. На электрической станции дежурный следит

за сигналами из цехов *A, B* и *C* и соответственно регулирует включение того или иного генератора.

Стоит вопрос, нельзя ли сформировать рекомендации на создание распознающего автомата, который заменил бы дежурного по заводской электрической станции и, получая сигналы от цехов *A, B, C,* сам бы решал, какой из генераторов включать? Приведенное задание ЛРО рассматривают как словесное задание автомата, которое содержит лишние высказывания о его работе.

ЛРО, ознакомившись с таким заданием, анализирует его с помощью следующих рассуждений: будущий распознающий автомат должен получать сигналы из трех цехов *A, B* и *C,* а это значит, что у него три входа. Сигналы, вырабатываемые распознающим автоматом, направляются в два адреса: на генератор *X* и на генератор *Y.* Значит, у него два выхода. Теперь можно составить таблицу работы распознающего автомата.

A	B	C	X	Y	A	B	C	X	Y
1	1	1	1	1	0	1	0	0	1
1	1	0	1	0	0	1	1	1	0
1	0	1	1	0	0	0	1	0	1
1	0	0	0	1	0	0	0	0	0

Если в энергии нуждаются три цеха, включены оба генератора, если два цеха — только генератор *X* или генератор *Y.*

Пользуясь составленной таблицей, следует составить структурную формулу распознающего автомата. Для этого следует брать те строки в таблице, в которых выход имеет значение, равное единице. В таблице таких строк четыре для выхода *X* и четыре для выхода *Y.* Составим формулу для выхода *X.*

На выходе *X* появится сигнал при поступлении сигналов от цехов *A, B, C* одновременно или от любых двух цехов одновременно — всего в четырех случаях. Теперь несложно составить структурную формулу, по которой должен действовать дежурный или заменяющий его распознающий автомат:

$$X = abc + ab\bar{c} + a\bar{b}c + \bar{a}bc. \qquad (C)$$

Аналогично составляют формулу распознающего автомата, вырабатывающего сигнал на выход Y; эта формула будет иметь вид

$$Y = abc + a\bar{b}\,\bar{c} + \bar{a}\,b\bar{c} + \overline{ab}\,c. \qquad (Д)$$

Теперь следует минимизировать полученные выражения.

Составим карту Карно для формулы *(С)*. Используя правила минимизации, получаем новую формулу для распознающего автомата, управляющего генератором *X* (рис. 5):

$$X = ab + ac + bc.$$

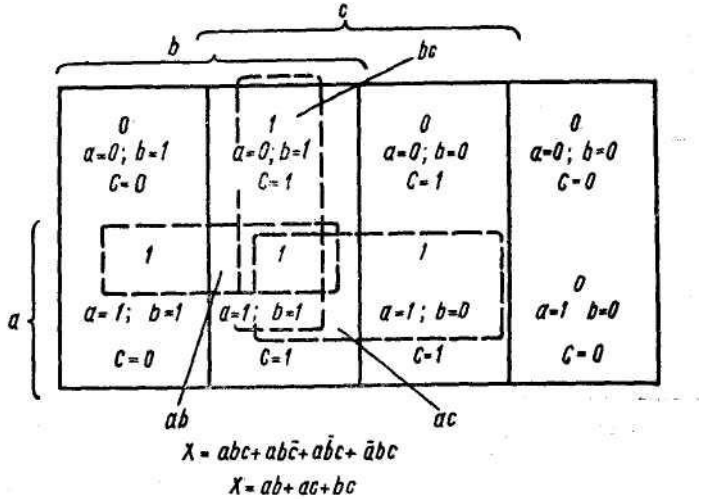


Рис. 5. Минимизация логического выражения для менее мощного генератора.

Наносим на карту Карно формулу *(Д)*. Так как единицы и нули на карте чередуются и нет ни одной пары смежных, которые можно было бы взять в контур, то выражение не поддается минимизации (рис. 6).

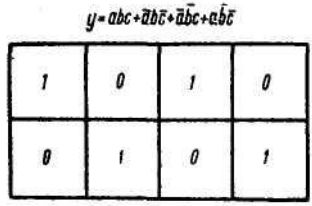


Рис. 6. Карта Карно для более мощного генератора.

Структурная формула распознающего автомата, управляющего включением в работу генератора *Y*, остается прежней:

$$Y = abc + a\bar{b}\,\bar{c} + \bar{a}\,b\,\bar{c} + \overline{ab}\,c.$$

Функциональная схема распознающего автомата приведена на рис. 7.

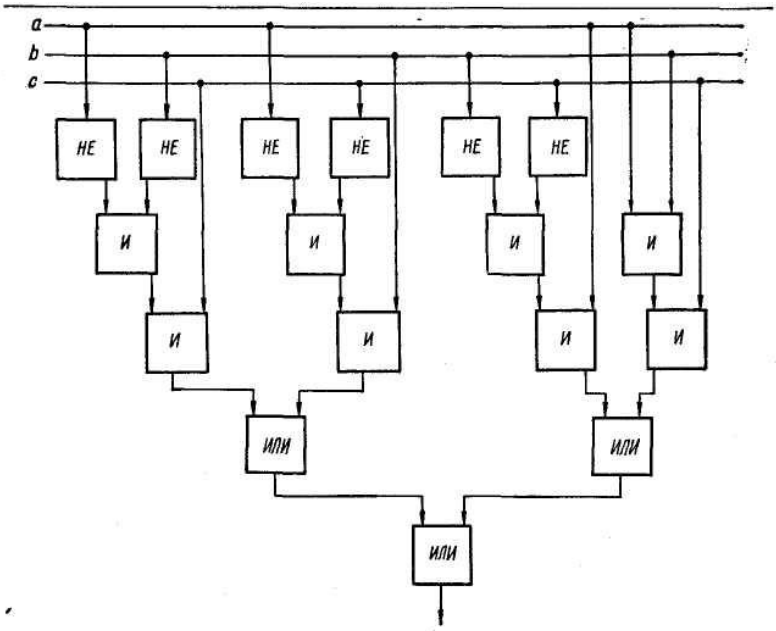


Рис. 7. Функциональная схема распознающего автомата, управляющего снабжением электроэнергией трех цехов

5.9.2. Исчисление предикатов как язык описания процессов распознавания

5.9.2.1. Введение в исчисление предикатов

Для осуществления автоматизации процессов распознавания необходим, как мы уже говорили, адекватный язык. Этот язык должен служить не только и не столько средством представления знаний (информации), сколько средством логического анализа задач распознавания.

Обычные человеческие языки, развивавшиеся под влиянием практических потребностей простоты общения (что далеко не всегда совместимо с точностью и надежностью логического анализа!), для этой цели плохо подходят. По этой причине желательно, даже практически необходимо, использовать в качестве языка логического анализа процессов распознавания специально созданный формализованный язык. Такой язык в противоположность обычному языку должен следовать за логической формой и воспроизводить ее даже в ущерб краткости и легкости общения, если это будет

необходимо. Главной отличительной чертой такого формализованного языка является наличие в нем особой системы логического вывода или дедукции.

В качестве такого формальзованного языка, удовлетворяющего указанным требованиям, мы возьмем *исчисление предикатов.* В терминах исчисления предикатов можно сформулировать многие предложения и утверждения, выраженные на естественных языках, а также формализовать процесс рассуждений и доказательств, который является базовым процесса распознавания. Благодаря этому может быть устранен или во всяком случае резко снижен «языковый барьер» между АРС и человеком.

Для того чтобы описать исчисление предикатов, мы должны воспользоваться какой-то частью обычного языка и в терминах этого языка образовать словарь и сформулировать правила формализованного языка, включая правила логического вывода. Исчисление предикатов (точнее, его синтаксис и семантика) определяется следующим очень экономным словарем символов и правилами их соединения и интерпретации:

1. Имена. Это — заимствованные из обычных языков выражения, служащие для непосредственного обозначения предмета. Примерами имен являются: «консультант», «мозг», «робот», «манипулятор», «процесс распознавания», «источник энергии», «искусственный интеллект» и т. д.

Следует отметить, что в одном или различных языках разные имена могут быть синонимами и выражать один и тот же смысл. С другой стороны, одно имя в различных языках или даже в одном языке (при омонимии) может выражать разный смысл.

Полное понимание языка требует знания смысла всех слов языка. Естественно потребовать, чтобы каждое имя имело точно один смысл. Такая однозначность обеспечивается в логике предикатов. А вот в обычных языках, как мы знаем, дело обстоит совсем не так.

2. Константы и переменные. *Константа* — это собственное имя. Примерами констант являются собственные имена чисел, людей, роботов, процессов, объектов.

Переменная — это символ, содержание которого совпадает с содержанием константы, за исключением лишь того, что единственный денотат константы заменен здесь возможностью различных значений переменной. С каждой переменной связана некоторая непустая область ее возможных значений. Поэтому к содержанию переменной относится в некотором смысле и содержание собственного имени области ее значений. Нужно особо подчеркнуть, что переменная в исчислении предикатов есть определенного рода

символ, а не предмет (например, число), который этот символ обозначает.

3. Функции и термы. *Функция*— это операция, которая будучи применена к чему-то как к аргументу, дает некоторый объект в качестве значения функции для данного аргумента. В природе всякой функции лежит свойство быть применимой лишь к некоторым предметам.

Предметы, к которым функция применима, составляют область определения функции, а ее значения составляют область значений функции. Сама функция состоит в определении некоторого значения для каждого аргумента из области ее определения. Например, функция распознавания состоит в определении номера класса, к которому принадлежит объект, трактуемый как аргумент.

Для того чтобы обозначить значение функции для некоторого аргумента, обычно пишут имя этой функции и приписывают к нему справа имя аргумента, взятое в скобки. Так, если f — функция, а x принадлежит к области ее определения, то $f(x)$ есть значение функции f для аргумента x. Если функция применима к упорядоченной системе из n аргументов, то она называется n-арной.

Важную роль в дальнейшем играют выражения для функций, значения которых принадлежат той же области, что и их аргументы. Такие выражения называются термами. *Терм* — это выражение, построенное, исходя из символов предметных переменных и констант, с помощью символов функций. Например, если f есть n-арная функция и уже известно, что $x_1, ..., x_n$—термы, то $f(x_1, ..., x_n)$ есть терм. Содержательно терму соответствует имя некоторого предмета.

4. Предложения, высказывания и предикаты. Простейшим выражением в обычных языках является предложение. *Предложение* - это такое соединение слов, которое имеет самостоятельный смысл, т. е. выражает законченную мысль. Каждому предложению сопоставим *высказывание* (выражаемое этим предложением), предполагая при этом, что каждое высказывание или истинно, или ложно и не может быть одновременной истинно и ложно. Таким образом, высказывание можно рассматривать как величину, принимающую только два значения: «истина» (И) или «ложь» (Л).

Предположим теперь, что x представляет собой произвольный предмет из некоторого множества $\{x\}$, а $F(x)$ — какое-либо высказывание о x. Выражение $F(x)$ становится определенным, когда переменная x заменена определенным значением (именем предмета) из множества $\{x\}$. Например, выражение «x есть животное» становится вполне определенным высказыванием, если x— это робот (ложное высказывание) или если x — это собака (истинное высказывание).

Так как с нашей точки зрения каждое определенное высказывание представляет собой И или Л, то выражение *F(x)* означает, что каждому предмету из *{x}* поставлен в соответствие один из двух символов: И или Л. Иначе говоря, *F(x)* представляет собой функцию, определенную на множестве *{x}* и принимающую только два значения: И и Л. Аналогично неопределенное высказывание о двух, трех и более предметах представляет функцию со значениями И и Л от двух, трех и более переменных. Эти неопределенные высказывания (функции одной или нескольких переменных) вида $F(x_1, ..., x_n)$ мы будем называть *логическими функциями* или *предикатами.*

5. Элементарные (атомарные) и правильно построенные формулы. Какой бы ни был символ *n*-местного предиката и каков бы ни был выбор термов $x_1, ..., x_n$, (не обязательно различных), выражение $F(x_1, ..., x_n)$ мы будем называть *элементарной,* или *атомарной формулой.* Из этого определения следует, что, например, имена предметов не являются формулами.

Рассматривая элементарные формулы как величины, способные принимать только значения И и Л, мы определим над ними операции, которые позволяют из данных формул получать новые. Эти операции, по существу, выражают употребительные в обычных языках связки.

Если *A* и *B* — какие-либо данные формулы (т. е. либо элементарные формулы, либо уже построенные сложные формулы), то $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ также являются (сложными) формулами. Если *A* — данная формула, то $\neg A$ — также (сложная) формула. Первые четыре операции — бинарные (двухместные), пятая — унарная (одноместная).

Символы $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$ называются соответственно *конъюнкцией, дизъюнкцией, импликацией, эквивалентностью* и *отрицанием.* Их можно читать, пользуясь словами, приведенными в правой части следующей таблицы:

$\wedge$ — «и»;
$\vee$ — «или», «... или, ... или», «и/или»;
$\rightarrow$ — «влечет», «если..., то...,» «только если»;
$\leftrightarrow$ — «равносильно», «эквивалентно», «тогда и только тогда»,
$\neg$ — «не», «неверно, что».

Прочтение сложных формул может стать неоднозначным, если не ввести скобок, указывающих, в каком порядке формулы связываются между собой. Поэтому мы будем писать $(A \rightarrow B) \rightarrow C$ или $A \rightarrow (B \rightarrow C)$, а не $A \rightarrow B \rightarrow C$. Впрочем, число скобок можно уменьшить, приписав нашим связкам убывающие «ранги» в следующем «порядке старшинства»:

$$\leftrightarrow, \rightarrow, \vee, \wedge, \neg.$$

Там, где возможны были бы два способа построения формулы, связка более высокого ранга имеет большую область действия. Так, $A \to B \wedge C$ означает $A \to (B \wedge C)$. Связка $\neg$ имеет наименьший ранг, так что, например, $\neg A \vee B$ означает $(\neg A) \vee B$, а не $\neg (A \vee B)$.

При построении сложных формул возникает вопрос, как определить значения сложных формул, зная значения простых формул, которые их составляют? Ответ на этот вопрос дается нижеследующей таблицей истинности.

A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$
И	И	Л	Л	И	И	И	И
Л	И	И	Л	Л	И	И	Л
И	Л	Л	И	Л	И	Л	Л
Л	Л	И	И	Л	Л	И	И

Таким образом, $A \leftrightarrow B$ истинно тогда и только тогда, когда A и B имеют одинаковые значения (почему $\leftrightarrow$ и называют «эквивалентностью»); $A \to B$ ложно тогда и только тогда, когда A истинно, а B ложно; $A \vee B$ истинно тогда и только тогда, когда и A, и B истинны; $A \vee B$ ложно тогда и только тогда, когда и A, и B ложны; наконец, $\neg A$ истинно тогда и только тогда, когда A ложно.

Кроме пяти упомянутых символов-связок, в исчислении предикатов употребляются еще два символа, выражающие операции утверждения всеобщности и существования. Символ $\forall x$ называется *квантором всеобщности,* а символ $\exists x$ — *квантором существования.*

Формула $\forall xF(x)$ истинна, когда $F(x)$ истинно для каждого элемента x области $\{x\}$, и ложна в противном случае. Соответствующее ей словесное выражение будет: «для всякого x $F(x)$ истинно». Формула $\exists xF(x)$ истинна, если существует элемент области $\{x\}$, для которого $F(x)$ истинно, и ложна в противном случае. В обычном языке этой формуле соответствует выражение: «существует x такое, что $F(x)$ истинно».

Мы будем говорить, что в формулах $\forall xF(x)$ и $\exists xF(x)$ переменная x связана соответствующим квантором. Ясно, что сами эти формулы от x не зависят. Заметим, что $\neg (\forall xF(x)) \leftrightarrow \exists x \neg F(x)$.

Теперь мы можем дать определение *правильно построенной формулы* (ППФ) на языке процесса распознавания. ППФ называется выражение, которое может быть построено исходя из элементарных

(атомарных) формул с помощью операций перехода от формулы A к формулам $\forall xF(x)$ и $\exists xF(x)$, от формул A и B к формулам $A \wedge B$, $A \vee B$, $A \to B$, $A \leftrightarrow B$, $\neg A$, $\neg B$. Элементарная формула или ее отрицание, входящие в ППФ, называются литерами (или литералами), а дизъюнкция литер называется простым дизъюнктом.

6. Интерпретации. ППФ имеет смысл только тогда, когда имеется какая-нибудь интерпретация входящих в нее символов. Под *интерпретацией* мы будем понимать всякую систему, состоящую из непустого множества D, называемого предметной областью (областью распознавания объектов), и какого-либо соответствия, относящего каждому символу n-местного предиката некоторое n-арное отношение в D, каждому символу функции от n аргументов некоторую n-местную операцию в D и каждой константе — некоторый элемент из D. Например, если D есть множество всех процессов распознавания, то отношение между двумя процессами распознавания, состоящее в том, что первый из них «лучше распознает» (по каким-то определенным параметрам) чем второй, можно отождествлять с множеством всех упорядоченных пар процессов распознавания (x, y) таких, что x проблемнее по части распознавания y. Таким образом, интерпретация осуществляет связь между языком процесса распознавания и описываемой им предметной областью (распознаваемым объектом) реального мира. Она позволяет придать ППФ содержательный смысл.

При заданной интерпретации всякая ППФ (не содержащая свободных переменных) представляет собой высказывание, которое истинно или ложно. Если при данной интерпретации каждая из ППФ A_i, $i = 1, ..., n$, имеет значение И, то будем говорить, что данная интерпретация удовлетворяет системе ППФ $\{A_i\}^n_{i=1}$. ППФ A выводима (логически следует) из некоторой системы ППФ $\{A_i\}^n_{i=1}$, если каждая интерпретация, удовлетворяющая $\{A_i\}^n_{i=1}$, удовлетворяет также и A. Так, очевидно, что ППФ $\forall xF(x)$ выводима из системы ППФ $\{\forall x \neg R(x) \vee F(x), . yxR(x)\}$.

Согласно теореме Гёделя, если некоторая интерпретация удовлетворяет заданной системе ППФ $\{A_i\}^n_{i=1}$, то она удовлетворяет и любой ППФ A, выводимой из этой системы. Умение продемонстрировать, что, ППФ A выводима (логически следует) из системы ППФ $\{A_i\}^n_{i=1}$, когда это на самом деле так, играет важную роль при логическом анализе, и мы сосредоточим на нем свое внимание. Предположим, что A выводима из $\{A_i\}^n_{i=1,}$ Тогда любая интерпретация, удовлетворяющая $\{A_i\}^n_{i=1}$, удовлетворяет A, но не удовлетворяет $\neg A$. Следовательно, никакая интерпретация не удовлетворяет объединению $\{A_i\}^n_{i=1} \vee \neg A$. Если некоторая система ППФ не удовлетворяется ни при какой интерпретации, то она

называется неудовлетворимой. Так, если ППФ A выводима из $\{A_i\}^n_{i=1}$, то объединение $\{A_i\}^n_{i=1} \vee \neg A$ неудовлетворимо. И наоборот, если $\{A_i\}^n_{i=1} \vee \neg A$ неудовлетворимо, то ППФ A должна логически следовать из системы ППФ $\{A_i\}^n_{i=1.}$ Именно эта концепция выводимости лежит в основе понятия логического вывода, или дедукции, в исчислении предикатов.

Универсальным методом логического вывода является так называемый *метод резолюций*, предложенный в 1965 г. Дж. Робинсоном. Этот метод замечателен тем, что он сложный процесс логического вывода сводит к последовательности очень простых операций, каждая из которых может быть легко запрограммирована.

В основе метода резолюций лежат три простых правила вывода *(резольвенции):*

1) если истинны ППФ A и $\neg A \vee B$, то истинна ППФ B (правило *modus ponens);*

2) если истинна ППФ $A \vee A$, то истинна ППФ A (правило факторизации);

3) если истинна ППФ $A(x)$, то истинна ППФ $\forall y A(y)$.

Эти правила применяются к простым дизъюнктам, на которые предварительно «раскладывается» система ППФ $\{A_i\}^n_{i=1} \vee \neg A$, из неудовлетворимости которой следует, что A выводима из $\{A_i\}^n_{i=1}$. Новые дизъюнкты, получаемые в результате применения указанных правил, называются *резольвентами*. При образовании резольвент существенную роль играет процедура *унификации,* которая для двух данных предикатов осуществляет подстановку термов вместо переменных, делающую предикаты одинаковыми. После этого к полученным ППФ применяются правила резольвенции. Например, для неудовлетворимой системы ППФ вида $\{A(x) \vee B(x), \neg B(f(z)), \neg A(f(z))\}$, используя первое правило вывода после подстановки терма $f(z)$ вместо переменной x, получим из первых двух ППФ резольвенту A $(f(z))$, которая в сочетании с третьей ППФ системы дает нулевую формулу. Таким образом, если выбрано два простых дизъюнкта и по одной литере в каждом из них, то применение правила унификации и затем правил вывода дает резольвенту. При доказательстве выводимости ППФ A, рассматриваемой как заключение (теорема), из заданной системы ППФ $\{A_i\}^n_{i=1}$, рассматриваемых как посылки (аксиомы), процесс образования резольвент (в котором могут принимать участие и ранее полученные резольвенты) продолжается, пока не будет получена пустая формула, означающая неудовлетворимость системы $\{A_i\}^n_{i=1} \vee \neg A$ и успех доказательства. Важно отметить, что число резольвент, формируемых при доказательстве любой теоремы из заданной конечной системы аксиом, конечно.

В ряде задач распознавания, которые должны решаться с использованием автоматизированных процессов процессов распознавания, простое доказательство выводимости ППФ *A,* формулирующей задание на распознавание, из системы ППФ $\{\{A_i\}^n_{i=1}$, описывающих условия выполнения этого задания, оказывается недостаточным. Примером такой задачи является задача по разработке процесса распознавания поведения робота при выполнении им заданных функций. В подобного рода задачах нужно знать то значение переменной *x,* при котором данная ППФ *A(x)* логически выводима из некоторой системы ППФ $\{A_i\}^n_{i=1}$. Иными словами, ЛРО (вместе с роботом) хотели бы знать, следует ли логически ППФ $\exists x A(x)$, и если да, то каково то значение *x,* при котором существует решение. Заметим, что умение отыскивать такие значения для переменной, связанной квантором существования, позволяет ставить распознающему объекту, в нашем случае - роботу, вопросы весьма общего характера и осуществлять диалог с ним. Например, оператор мог бы рекомендовать спросить у робота: «Какие действия и в какой последовательности нужно совершать, чтобы собрать из деталей определенную конструкцию?». Ответом на этот вопрос будет не просто констатация факта, что сборка данной конструкции возможна, а развернутый план рекомендаций (технологический маршрут) сборки.

Рассмотрим на простейшем примере, как можно рекомендовать решать подобного рода задачи. Пусть роботу известно, что его манипулятор жестко закреплен на подвижной платформе, а платформа находится в цехе. Рекомендуется спросить: «где находится манипулятор?». В этой задаче сформулированы два «факта», которые можно записать в виде двух правильно построенных формул:

$A_1 \leftrightarrow \forall x P$ (платформа, *x)*$\rightarrow P$(манипулятор, *x), *$A_2 \leftrightarrow P$(платформа, цех),

где двухместному предикату *P(y, z)* придана очевидная интерпретация: *«y* находится в *z*». На вопрос «где находится манипулятор?» робот может дать ответ, если сначала докажет, что правильно построенная формула

$$A_1 \leftrightarrow \forall x P \text{ (манипулятор, } x)$$

выводима из системы ППФ $\{A_i\}^n_{i=1}$, и затем найдет то значение *x* (константу), которое на самом деле «существует» и служит ответом.

Используя описанный выше рекомендованый метод резолюций, робот сначала попытается доказать неудовлетворимость системы $\{A_i\}^n_{i=1} \vee \neg P$ (манипулятор, *x).* (Заметим, что отрицание ППФ *A* есть ППФ $\neg P$ (манипулятор, *x)).* Процесс доказательства неудовлетворимости $\{A_i\}^n_{i=1} \vee \neg A$ представлен на дереве вывода, изображенном на рис. 8.

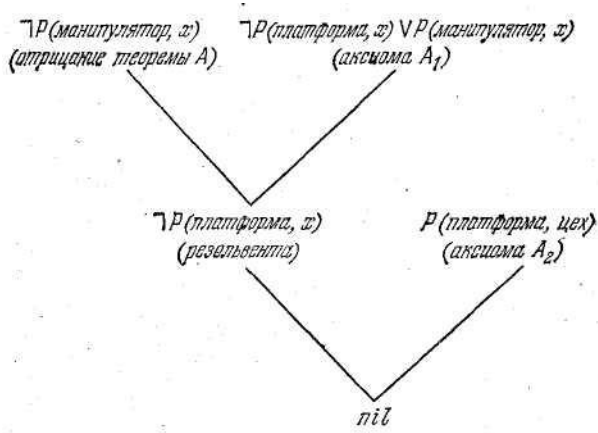


Рис. 8. Дерево вывода.

Из этого дерева вывода можно извлечь также ответ на вопрос оператора: «где находится манипулятор?». Это осуществляется следующим образом. Сначала к отрицанию теоремы добавляется ее отрицание, т. е. сама теорема. В результате получается тавтология (т.е. ППФ, тождественно истинная при всех интерпретациях) вида

$$\neg P(\text{манипулятор}, x) \vee P(\text{манипулятор}, x).$$

Затем в соответствии со структурой дерева вывода, изображенного на рис. 8, вновь формируются резольвенты до тех пор, пока в корне дерева не получится некоторая ППФ, играющая роль ответа на языке робота. В нашем примере получим одну резольвенту

$$\neg P(\text{платформа}, x) \vee P(\text{манипулятор}, x),$$

а в корне дерева — ППФ P (манипулятор, цех), в которой содержится ответ на вопрос «где находится манипулятор?». Заметим, что форма ответа на языке предикатов близка к форме теоремы-вопроса. В нашем случае единственное отличие состоит в том, что в теореме-вопросе содержится переменная, связанная квантором существования, а в ответной ППФ — константа (ответный терм).

Таким образом, описанная система логического вывода, основанная на методе резолюций, представляет собой эффективное средство для автоматического поиска доказательств (отыскания логических следствий) и извлечения ответа в терминах исчисления предикатов. Мы рассмотрели основные понятия этого исчисления и связанного с ним метода резолюций не ради них самих, а чтобы понять и продемонстрировать, как ЛРО, используя этот язык, может логически

рассуждать, обучаться новому и адаптироваться в процессе решения задач распознавания в автоматизированной распознающей системе.

5.9.2.2. Адаптация процессов распознавания логического вывода

Первым и неизбежным этапом применения описанной выше системы логического вывода для автоматизированного формирования рекомендаций по решения задач распознавания, требующих логического анализа, является формулировка этих задач в терминах исчисления предикатов. Для этого нужно, прежде всего, задать предметную область (распознаваемый объект), т. е. совокупность относящихся к решаемой задаче объектов (или процессов), и выделить их существенные свойства, от которых в наибольшей степени зависит успех формирования рекомендаций по решению задач распознавания. Далее нужно, присвоив определенный содержательный (семантический) смысл предикатным и функциональным символам, формализовать данные и условия задачи в виде ППФ, которые должны на них выполняться (т. е. истинность которых считается не требующей доказательства). Очевидно, что эти ППФ выделяют из всевозможных распознаваемых объектов, их свойств и отношений между ними такие распознаваемые объекты, для которых они выполнены.

ППФ, посредством которых мы таким образом выделяем совокупность распознаваемых объектов, называются аксиомами. Если для какой-либо совокупности распознаваемых объектов, их свойств и отношений некоторые аксиомы истинны, то говорят, что данная совокупность распознаваемых объектов удовлетворяет системе этих аксиом или является интерпретацией данной системы аксиом.

Таким образом, аксиомы можно рассматривать как определения системы распознаваемых объектов, их свойств и отношений между ними. Делая логические выводы из аксиом, мы будем получать ППФ, истинные для любой системы распознаваемых объектов, удовлетворяющей данным аксиомам.

Ясно, что соответствие между аксиомами и предметами распознаваемой реальности, т. е. предметной областью, всегда имеет приближенный характер. Поэтому возникает вопрос, как узнать, действительно ли данная система аксиом определяет именно то, что было задумано, что требуется для решения задачи распознавания?

Ответ на этот вопрос связан с понятием *непротиворечивости системы аксиом.* Мы должны быть уверены, что делая всевозможные выводы из данной системы аксиом, не придем к противоречию, т. е. не выведем какие-либо несовместимые ППФ. Появление противоречия означало бы, что рассматриваемой системе аксиом не может удовлетворять никакая совокупность распознаваемых объектов, и,

таким образом, эти аксиомы ничего не описывают. Мы будем говорить, что система аксиом $\{A_i\}^n_{i=1}$ противоречива, если в ней выводима какая-либо ППФ *A,* а также и ее отрицание $\neg A$. Для проверки (доказательства) непротиворечивости системы аксиом достаточно построить какую-нибудь точную интерпретацию этой системы.

Весьма важным является свойство *независимости аксиом.* Какая-либо аксиома *A* называется *независимой* в данной системе аксиом $\{A_i\}^n_{i=1}$, если она невыводима из остальных аксиом этой системы. Для проверки (доказательства) независимости какой-либо аксиомы достаточно найти совокупность распознаваемых объектов, удовлетворяющую всем аксиомам, кроме исследуемой, и не удовлетворяющей этой последней. Иными словами, для проверки независимости аксиомы *A* достаточно найти интерпретацию следующей системы аксиом: $\{A_i\}^n_{i=1}, \neg A$.

Таким образом, система аксиом, которой пользуется робот, должна иметь точную интерпретацию в том мире объектов, свойств и отношений, в котором он функционирует. Этому требованию можно удовлетворить путем правильной формулировки тех задач распознавания, которые робот должен решать. Остановимся на этом вопросе подробнее.

Формулировка задачи распознавания на языке предикатов— это первый и наиболее ответственный этап организации его целенаправленного поведения. На этом этапе от ЛРО требуются глубокие знания не только и не столько исчисления предикатов, сколько существа решаемой задачи распознавания, ее специфических черт, распознаваемого объекта, той цели, которая должна быть достигнута в результате реализации процесса распознавания. Возможна, но совершенно бессмысленна постановка на языке предикатов таких, например, задач: «рекомендую переместиться туда, сам не знаю куда», или « рекомендую найти то, сам не знаю что».

Практически весьма важно, чтобы формулировка задачи распознавания (связанная с заданием системы аксиом и теорем-заданий) была по возможности простой, не «засоренной» массой мелких, второстепенных факторов, так как учет их существенно осложняет логический анализ и делает трудно обозримыми результаты решения. Отметим две типичные трудности, которые всегда подстерегают «формулировщика» задачи. Первая — это возможность «утонуть в деталях и подробностях», т. е. «из-за деревьев не увидеть леса»; вторая — слишком огрубить задачу, или, как принято говорить в подобных случаях, «вместе с водой выплеснуть и ребенка». Ниже на примерах формулировки задач по формированию рекомендаций

планирования поведения робота и распознавания сложных ситуаций мы увидим, что искусство формулировки задач распознавания на языке предикатов есть именно искусство. Здесь нет общих рецептов, а опыт ЛРО в этом трудном деле приобретается постепенно.

При построении аксиом будем различать два типа предикатов, использование которых по-разному сказывается на быстроте формирования рекомендаций. Предикаты первого типа описывают простейшие свойства конкретных объектов (например, «робот находится в точке *x»,* «объект *z* большой» и т. п.). Предикаты второго типа определяют общую картину отношений между различными объектами и их свойствами. Один такой предикат может описывать набор свойств большого числа объектов (например, «если между точками *a* и *b* нет препятствий, то робот может проехать между этими точками по прямой» и т. п.). Количество подобных предикатов, необходимое для описания (с требуемой степенью подробности) данных и условий задачи, обычно невелико. Однако для сложных предикатов существенно возрастает сложность термов, участвующих в их определении.

Многие задачи распознавания часто связаны с изменением во времени свойств распознаваемых объектов, которые обычно известны в начальный момент времени. В таких задачах удобно ввести предикат позиции, определяющий все «интересные» свойства всех распознаваемых объектов. При этом аксиомы, описывающие изменение предиката позиции во времени, в наиболее простой форме могут быть составлены из трех литер, а именно, если имеется некоторая позиция (ситуация, проблема) и если выполняется некоторое дополнительное условие, характеризующее принципиальную возможность применения данной аксиомы, то получится новая позиция (ситуация, проблема). Основным преимуществом такого способа построения аксиом является то, что свойства, связанные между собой, определяются одним предикатом и поэтому меняются одновременно. При этом на каждом шаге формирования рекомендации учитывается все многообразие сложившейся ситуации, вследствие чего уменьшается число резольвент в процессе логического вывода.

Из дальнейшего изложения (и, в частности, из примеров) будет ясно, что эффективность системы логического вывода можно увеличить путем уменьшения числа предикатов и аксиом, определяющих данные и условия задачи. С этой целью разумно использовать ранее доказанные теоремы или ввести более сложные предикаты, образующие новые аксиомы, которые можно рассматривать как результат обучения ЛРО в процессе формирования рекомендаций. Такие аксиомы, описывающие на языке исчисления предикатов

приобретаемый ЛРО опыт, мы будем называть *аксиомами обучения.* Введение аксиом-обучения как бы моделирует феномен мышления, о котором еще Р. Декарт писал в своем «Рассуждении о методе»: «Каждая решенная мною задача становилась образцом, который служил впоследствии для решения других задач». Образно говоря, аксиомы обучения играют роль лемм при доказательстве новых теорем, определяющих целевые условия задачи. Тем самым они позволяют оперировать более крупными «блоками» (фрагментами) доказательств, освобождая от рассмотрения многочисленных деталей, имеющих в данном доказательстве лишь вспомогательное значение. Заметим, что введение аксиом обучения позволяет ЛРО увеличивать и улучшать знания о решаемом классе задач распознавания в процессе их непосредственного решения. Таким образом, аксиомы обучения являются средством обучения новым понятиям и фактам и уточнения старых.

Скорее всего, в будущем ЛРО так и не удастся прийти ни к какой определенной конечной системе аксиом, рассматриваемой как окончательная. Напротив, подобно тому, как это происходит в мире живого, будут появляться (автоматически формироваться) все новые аксиомы обучения, отображающие изменения в окружающем ЛРО мире и в решаемом им классе задач.

Построение системы аксиом в каждой задаче важно не само по себе, а имеет целью выявление оптимальных путей логического вывода. Под *эффективностью системы логического вывода* мы будем понимать меру успешности и поиска процессов распознавания. Для того чтобы выбрать количественный показатель эффективности процесса распознавания, нужно прежде всего спросить себя: чего мы хотим от системы логического вывода, к чему стремимся при формированни процесса распознавания? Выбирая (формируя) процесс распознавания, мы предпочитаем такой, который при его реализации обращает показатель эффективности распознавания в максимум или же в минимум.

Очень часто в качестве показателя эффективности систем логического вывода фигурируют затраты на формирование процесса распознавания (доказательства), которые, естественно, нужно минимизировать. Заметим, что неправильный выбор показателя эффективности очень опасен, так как он может привести к плохим процессам распознавания. Процессы распознавания, выбранные под углом зрения неудачно выбранного показателя эффективности, могут привести к большим неоправданным потерям и затратам.

В рассматриваемом круге задач под *эффективностью* системы логического вывода будем понимать число шагов доказательства

(возможно, усредненное по классу решаемых задач), т. е. число резольвент, формируемых в процессе поиска распознающих признаков. С целью увеличения эффективности системы логического вывода введем некоторые ограничения на процесс образования резольвент, связанные с выбором стратегии формирования процесса распознавания.

Стратегией логического вывода называется способ выбора очередной пары дизъюнктов и литер в них для образования резольвент. Именно стратегия определяет, в каком порядке будут образовываться резольвенты и, следовательно, насколько быстро будет найден (сформирован) процесс распознавания. Стратегия «запускает» процесс доказательства — *начинается дедукция*: с помощью аксиом, резольвент и теорем строится та или иная конструкция доказательства. При этом стратегия решает, какие понятия и факты (аксиомы и литеры в них) несущественны, а какие — необходимы для доказательства. Таким образом, выбор и подстройка стратегии являются основным средством увеличения эффективности системы логического вывода, определяющим быстроту сходимости реализуемого ею метода поиска доказательства. Если правила резольвенции есть правила дедуктивного вывода следствий, то стратегия — это та активная часть, способная к обучению и адаптации, которая имитирует способ «мышления», например, искусственного интеллектуального робота, уровень его познаний в логике, степень его интеллектуальности.

Образно говоря, *стратегия системы логического вывода — это идея формирования процесса распознавания (доказательства), исходя из заданной системы аксиом, в которой заключены вес необходимые для решения задачи знания*. Если идея (стратегия) хороша, то процесс распознавания будет сформирован (найден) быстро. Однако рассчитывать на хороший процесс распознавания (стратегию) ЛРО может лишь тогда, когда в системе аксиом достаточно полно отражены не только необходимые знания о задаче, но и прошлый опыт решения задач в подобных проблемах. Хорошие процессы распознавания (стратегии) имеют своим источником прошлый опыт и ранее приобретенные знания.

Стратегия называется *полной,* если она находит (в конечное число шагов) доказательство любой ППФ, выводимой из аксиом. Примерами полных стратегий являются стратегия опорного множества, стратегия предпочтения единичным элементом, а также тривиальная стратегия полного перебора. Все перечисленные стратегии характеризуются тем, что для них предикат является «неразложимым» понятием, а критерием выбора очередной пары дизъюнктов могут быть количество предикатов в дизъюнкте, порядок их расположения и т. п. Такие

стратегии, не зависящие от внутренней структуры, смысла используемых предикатов, будем называть *синтаксическими*.

Стратегию (процесс распознавания) будем называть *адаптивной,* если она целенаправленно меняется (подстраивается) в процессе логического вывода в зависимости от приобретаемого опыта. Примерами адаптивных стратегий могут служить *семантические* стратегии, в которых критерий выбора очередных дизъюнктов зависит от вхождения в них определенного терма. *Согласно семантической стратегии сначала выбираются термы, соответствующие «интересным» процессам распознавания, затем — предикаты, описывающие их свойства, и, наконец, ППФ, содержащие эти свойства.*

В предыдущем пункте мы отмечали, что процессу поиска доказательства может быть поставлено в соответствие дерево вывода, которое заканчивается пустым дизъюнктом, означающим конец и успех доказательства. Дерево вывода строится и ветвится под каждой резольвентой так же, как и под теоремой (точнее, ее отрицанием). Отсюда ясно, что резольвенты, не приводящие к пустому дизъюнкту, резко увеличивают число шагов доказательства, и их получение крайне нежелательно. В связи с этим задача построения адаптивной стратегии может быть переформулирована как задача отсечения ненужных (лишних) ветвей на дереве вывода. Для решения этой задачи необходимо указать критерий предпочтения ветвей.

Действительно, в процессе доказательства теоремы можно указать, как правило, несколько подходящих аксиом и несколько путей (ветвей) доказательства. Если нет критерия предпочтения одной ветви другой, приходится действовать по методу случайного поиска, что соответствует образованию пучка ветвей на дереве вывода. Введение подходящего критерия предпочтения позволяет исключить лишние тупиковые ветви и благодаря этому существенно увеличить эффективность системы логического вывода. При этом особую роль играют критерии предпочтения, формируемые в процессе решения задач. Примером такого критерия является критерий предпочтения аксиом обучения (хранящихся в памяти наряду с исходной системой аксиом), которые позволяют уменьшить исходную неопределенность относительно условий формирования процессов распознавания.

Введение аксиом обучения, о которых шла речь выше, особенно эффективно в тех случаях, когда в них либо раскрывается неопределенность (т. е. содержится новая необходимая для формирования процессов распознавания информация), либо «запоминается» в компактной форме часто встречающийся в рассматриваемом классе задач «фрагмент» процесса распознавания (доказательства). В

самом деле, если в процессе формирования процесса распознавания потребуется доказать уже доказанную ране теорему, то критерий предпочтения аксиом обучения сократит общее число шагов доказательства по крайне мере на длину доказательства соответствующей аксиом обучения.

Важной особенностью адаптивной системы логического вывода является ее способность логически рассуждать, т. е. сводить сложное заключение к последовательности утверждений, истинность каждого из которые проверяется очень просто и чисто механически. Такая система может также не только автоматически доказывай теоремы, трактуемые как некоторые задания или вопросы, но и обучаться способам их доказательства.

Рассмотрим теперь применение адаптивной системы логического вывода для автомазированного формирования процессов распознавания по решению задач распознавания поведения робота, распознавания и описания сложных изображений трехмерной среды, получаемых с помощью автоматизированной системы распознавания в условиях неопределенности.

5.9.2.3. Логические алгоритмы формирования процессов распознавания поведения робота

Содержание задачи формирования процессов распознавания по распознаванию поведения робота поясним на примере того, как эту задачу решает человек. Первое, с чем мы сталкиваемся ежедневно, — это задача утреннего одевания. Мы должны сформировать для себя рекомендации выработки плана действий, который позволит нам одеться, причем так, чтобы выполнялись естественные общепринятые ограничения (рубашку надевать необходимо, но не поверх пиджака, и т. п.). При этом время — наш основной ресурс, и выбранный план должен быть наилучшим в том смысле, в каком каждый понимает расход своего утреннего времени.

Если отбросить некоторые «несущественные» детали, план одевания должен оперировать такими предметами, как туфли, носки, брюки, рубашка, галстук, пиджак и пальто. *Рекомендация по реализации плана действий представляет собой любой порядок, в котором можно надеть эти предметы.* Всего в этом случае существует 7! = 5040 различных вариантов плана. Многие из них недопустимы, так как либо не удовлетворяют общепринятым ограничениям (рубашка поверх пиджака, носки поверх ботинок), либо непрактичны (галстук под рубашкой) и являются нереализуемыми рекомендациями. Но даже после того, как эти недопустимые рекомендации будут отброшены, все равно придется исследовать некоторое количество допустимых планов (рекомендаций). Как же

выбрать окончательный (желательно, оптимальный) рекомендуемый план? Прежде всего заметим, что в рассматриваемой задаче имеется некоторая мера эффективности, некий критерий, позволяющий нам сравнивать эффективность рекомендуемых допустимых планов. Если мы можем каким-то образом сравнить значение этого критерия для различных планов, то мы сможем тем самым выбрать из них оптимальный. В данной конкретной задаче естественно рекомендовать минимизировать время, необходимое для того, чтобы одеться. Это и есть та мера эффективности рекомендации, с помощью которой можно сравнивать рекомендуемые допустимые планы. Тогда в качестве рекомендованного оптимального плана, позволяющего одеться, не нарушая общепринятых ограничений, можно выбрать следующий план: носки, рубашка, брюки, галстук, туфли, пиджак, пальто. Ясно, что при другом критерии эффективности поведения оптимальным может оказаться иной план.

Характерной особенностью задач планирования является наличие многих допустимых рекомендаций. После того, как эти рекомендации сформированы, возникает следующая задача: выбрать среди них по крайней мере одну оптимальную (в смысле определенного критерия) рекомендацию по реализации плана действий.

В рассмотренном нами простейшем примере рекомендации формировались без специальных обоснований, просто на основе опыта и здравого смысла человека. Оптимизация таких рекомендаций происходит как бы сама собой, в процессе жизненной практики. Если порой выбранная рекомендация окажется не самой удачной, так что же? На ошибках учатся. Нередки, правда, ситуации (соответствующие, например, планированию мероприятий, осуществляемых в первый раз), когда использовать эвристические решения, основанные на опыте и здравом смысле, просто невозможно. В подобного рода ситуациях «опыт» молчит, а «здравый смысл» легко может обмануть, если не будет опираться на математический расчет.

Но бывают рекомендации несравненно более сложные, а главное ответственные — при их реализации от них очень многое зависит. Конечно, при планировании поведения в подобного рода ситуациях можно действовать интуитивно, опираясь опять-таки на опыт и здравый смысл. Но гораздо более разумными могут оказаться рекомендации, подкрепленные количественными, математическими расчетами соответствующего плана действий. Эти предварительные расчеты помогут избежать длительного и накладного поиск рекомендации «на ощупь».

«Семь раз отмерь, один раз отрежь», — говорип известная поговорка. *Формирование рекомендаций по планированию поведения*

как раз и представляет собой своеобразное математическое «отмеривание» к потребному будущему, позволяющее заранее оценить последствия каждой рекомендации, заранее отбросить недопустимые планы и рекомендовать наиболее удачные. Эти последние позволят установить, достаточна ли имеющаяся у нас информация для правильного формирования рекомендации, и если нет— какую информацию нужно дополнительно получать и обрабатывать. Все это позволяет при реализации рекомендованного плана экономить время, энергию и материальные средства.

Необходимость в формировании рекомендаций по планированию поведения возникает у робота при выполнении им сложных заданий в условиях большой априорной неопределенности (например, сборка сложного изделия по чертежу, поиск и транспортировка нужного объекта на неизвестной местности с препятствиями и т. п.). Задача автоматизированого формирования процесса распознавания по планированию поведения решаемая на втором уровне иерархии системы управления робота, может быть переформулирована на языке исчисления предикатов как задача логического вывода (автоматического доказательства теорем). При таком подходе априорные сведения о свойствах и функциональных возможностях робота и окружающей его среды необходимо прежде всего представить в виде правильно построенных формул (ППФ). Совокупность таких ППФ мы будем называть *априорными аксиомами* и разобьем их на четыре класса:

1) сенсорные аксиомы (СА);
2) моторные аксиомы (МА);
3) аксиомы среды (АС);
4) аксиомы начальных условий (АНУ).

Сенсорные аксиомы описывают функциональные возможности информационно-измерительной системы робота, а *моторные аксиомы* — функциональные возможности исполнительных механизмов робота. *Аксиомы среды* определяют состояние и эволюцию среды, а *аксиомы начальных условий* описывают начальные состояния робота и среды. В табл. 1 и 2 приведены типичные для задачи планирования поведения робота на местности с препятствиями: термы, функции, предикаты и априорные аксиомы вместе с их интерпретацией на обычном (русском) языке.

Таблица 1

Функции и предикаты	Интерпретация на естественном языке
$f(a, b, s)$	Ситуация, наступающая после выполнения роботом, находящимся в ситуации s, действия «переехать из a в b по прямой».
$p(s)$	Ситуация, наступающая после выполнения манипулятором, находящимся в ситуации s, действия «погрузить объект на тележку».
$q(s)$	Ситуация, наступающая после выполнения манипулятором, находящимся в ситуации s, действия «сгрузить объект с тележки».
$G(a, b)$	Истинен, если из a в b можно проехать по прямой.
$Pos(s, x, y)$	Истинен, если в ситуации s робот находится в состоянии x, а объект — в состоянии y (предикат позиции).
$Pos(s, x, fin)$	Истинен, если в ситуации s объект находится на складе, указанном в задании, а робот — в состоянии x.
$Pos(s, fin, y)$	Истинен, если в ситуации s робот находится в конечной точке заданного маршрута, а объект — в состоянии y.

Наряду с априорными аксиомами введем *аксиомы обучения* (АО), автоматически формируемые по мере накопления роботом опыта и знаний в процессе выполнения тех или иных заданий. Задания роботу, формулируемые оператором, будем трактовать как заключения теорем, посылками которых служат априорные аксиомы и аксиомы обучения. Заметим, что формирование рекомендаций по выбору аксиом и теорем диктуется окружающим робота миром, который он воспринимает своими органами чувств и на который воздействует своими исполнительными механизмами, а также структурно-функциональными особенностями робота и целями (задачами) его функционирования.

Таблица 2

Тип	Аксиомы: Логическое представление	Смысл на естественном языке
МА	$\forall s \forall x \forall y \forall z\ (\neg Pos\ (s, x, y) \vee \neg G(x, z) \vee Pos\ (f(x, z, s), z, y))$	Если в ситуации s, робот находится в состоянии x, а объект — в состоянии y, и из x можно переехать по прямой в z, то в ситуации $f(x, z, s)$ (т. е. после реализиции функции движения «переехать из точки x в точку z») робот окажется в точке z.
АС	$G\ (a, b)$	Из точки a в точку b можно проехать по прямой.
СА	$K\ (x)$	Объект x принадлежит k-му классу.
АНУ	$Pos\ (s_0, O, C_1)$	В начальной ситуации s_0 робот находится в точке O, а объект — на складе C_1.

Для автоматического доказательства теорем-заданий и извлечения ответа на языке робота целесообразно применить адаптивную систему логического вывода, описанную выше. Такая система в результате доказательства теоремы-задания (или теоремы-вопроса) сформирует процесс распознавания по указанию, какие действия и в какой последовательности нужно роботу совершить для выполнения задания, т. е. выдаст процесса распознавания по реализации искомого план поведения робота.

Продемонстрируем работу адаптивной системы логического вывода в задаче формирования процесса распознавания по планированию поведения робота на примере. Пусть робот находится в цехе с оборудованием (трактуемом как препятствия), где имеются склад заготовок C_1 и склад готовых изделий C_2 (см. рис. 9). Вначале робот находится в точке O и перед ним ставится задача: перевезти определенный объект (который еще нужно распознать) со склада C_1 на склад C_2 (местоположение складов известно) и после этого покинуть цех через выход. Предполагается, что выход задан набором признаков (его координаты роботу неизвестны), а сенсорная система может

измерять значения признаков и координаты видимых ею точек, причем она не может «видеть» сквозь препятствия. Кроме перечисленного и исходной системы аксиом, представленной в табл. 2, роботу ничего неизвестно.

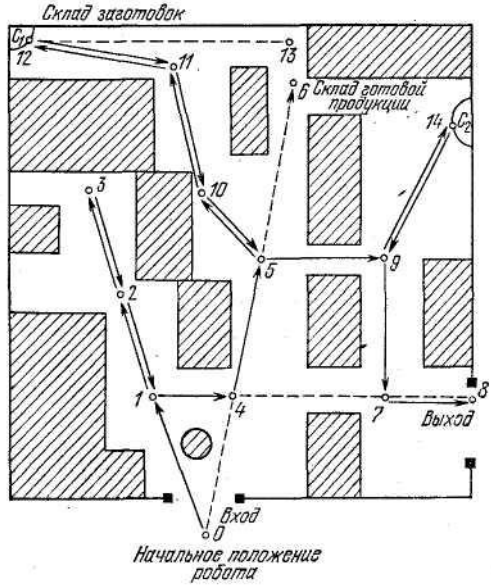


Рис. 9. Планирование поведения робота в незнакомом помещении.

Поскольку решение этой задачи требует дополнительной информации об обстановке в цехе, робот вначале опрашивает сенсорную систему. При этом отыскиваются все видимые границы препятствий и около них фиксируются некоторые точки *1-6*, к которым робот может проехать по прямой. В результате автоматически строятся аксиомы среды: *G(0, 1), G(1, 2), G(2, 3), G(0, 4), G(4, 5), G(5, 6).* По этим данным, а также по аксиоме начальных условий $Pos(s_0, O, C_1)$ робот пытается доказать теорему-задание $\exists s\ Pos(s, fin, fin)$ (где *s*— переменная, описывающая ситуацию). Однако, поскольку знаний о среде, заключенных в построенных АС, явно недостаточно (теорема не выводима из АС), ответ, т. е. процесс распознавания по реализации искомого плана поведения, не будет получен. В процессе логического вывода робот убеждается, что маршруты через точки *О, 1, 2, 3* и *0, 4, 5, 6* к выполнению задания не приводят. Далее, используя критерий близости к складу C_1, робот принимает решение переместиться в точку *3.* Последовательность действий на этом этапе определяется термом

ситуации в резольвенте *f(2, 3, f(1, 2, f(0, 1,* s_0))), который расшифровывается в обратном порядке и в соответствии с определением функции *f* означает: переехать из точки *O* в точку 1, затем в точки *2* и *3*. Передвигаясь согласно выбранному маршруту, робот останавливается в каждой из них и пополняет свои знания о среде посредством опроса сенсорной системы. Так возникают новые аксиомы среды (АС): *G(1, 4), G(4, 7),G(7, 8),* $\neg$ *Pos(s, 8, y)* $\vee$ *Pos(s, fin, y)* (последняя аксиома среды означает, что точка *8* находится у выхода).

Приехав в точку *3*, робот не формирует ни одной новой АС, а поэтому и новой резольвенты. Он в тупике. «Осознав» это, робот вынужден развернуться и исправить те действия, которые привели его в «тупиковую» точку *3*, в обратном порядке, пока не появится первая возможность образования новой резольвенты. В результате он выбирает маршрут через точки 1, *4, 5*, используя ранее построенные аксиомы обучения (АО) (к числу которых относятся АС и СА, формируемые в процессе функционирования робота). Так, формируя и корректируя рекомендации по реализации локальных планов поведения на основе целенаправленной переработки новой информации, робот, в конце концов, решает поставленную задачу: отыскивает (распознает) нужный объект на складе C_1, погружает (с помощью манипулятора) его на тележку, подвозит к складу C_2, сгружает объект и покидает цех через выход. При этом адаптивная система логического вывода строит 47 резольвент. Окончательный маршрут робота, реализующий выработанный план поведения, изображен на рис. 9 сплошными линиями со стрелками.

Рассмотренный пример формирования процесса распознавания по планированию поведения робота в условиях большой априорной (начальной) неопределенности замечателен тем, что он ясно демонстрирует, что обычная (неадаптивная) система логического вывода принципиально не способна решать такого рода задачи без использования элементов обучения и адаптации. Важно отметить, что автоматическое формирование АО и адаптивная подстройка стратегии, использующей АО, не только делает задачу разрешимой, но и существенно сокращает (за счет отсечения многих тупиковых ветвей на дереве вывода) число шагов в процессе поиска плана поведения как при полной, так и при частичной информированности об условиях функционирования робота.

5.9.3.4. Алгоритмы распознавания ситуаций

Задача формирования рекомендаций по описанию, распознаванию и анализу ситуаций, является одной из центральных проблем распознавания с использованием автоматизированных процессов

распознавания. Для формализации и автоматизации решения этой задачи, как мы увидим ниже, опять-таки удобно использовать язык исчисления предикатов и адаптивную систему логического вывода. В качестве примера, на котором мы будем демонстрировать использование языка исчисления предикатов и адаптивной системы логического вывода, рассмотрим функционирование робота. Поскольку для робота наиболее важное значение имеет зрительная информация, мы сосредоточим внимание на методах по описанию, распознаванию и анализу трехмерных сцен по их изображениям. Что же касается задач переработки речевой, тактильной и другой сенсорной информации, то они могут решаться по существу теми же методами. Трудности, возникающие при формировании рекомендаций для решения задачи распознавания отдельных предметов на сложной сцене, связаны с наличием «порочного круга»: для того чтобы распознать некоторый предмет на сцене, нужно прежде всего его «выделить», а для того чтобы «выделить» этот предмет, нужно его распознать. Это приводит к тому, что классические методы распознавания «перцептронного» или статистического типа, в этой задаче практически не применимы. На первый взгляд кажется, что выход из указанного «порочного круга» только один — полный перебор элементов изображения предмета и сцены. Однако более глубокий анализ этой задачи позволяет сформировать рекомендации по ее формулировке и решению как задачи логического вывода.

Идея предлагаемого решения основана, во-первых, на том, что применяется предварительное обучение робота путем показа ему отдельных предметов из различных классов и сообщения ему не только названия предмета, но и, возможно, его описания. Это обучение ведется лператором в диалоговом режиме, позволяющем оперативно выявлять и исправлять ошибки робота. Во-вторых, специфика задачи ярко проявляется в большой вариативности изображений реальных предметов и сцен, которая имеет двоякую природу. С одной стороны, она порождается естественной вариативностью характеристик самих предметов, с другой стороны, — перемещением предметов в пространстве. Вариативность второго рода можно трактовать как результат действия некоторых известных преобразований изображения. Априорное знание этих преобразований позволяет построить алгоритм распознавания, инвариантный по отношению к этим преобразованиям. Благодаря этому удается не только «избавиться» от вариативности второго рода, но и существенно облегчить задачу переработки зрительной информации.

Сама эта задача подразделяется на следующие подзадачи: *описание классов (формирование понятий о классах объектов), распознавание*

изображения данного предмета, анализ изображения сцены (распознавание всех предметов на сцене). Результаты решения перечисленных подзадач могут использоваться для формирования рекомендаций по моделированию внешней среды (на четвертом уровне иерархии) путем преобразования изображений предметов и сцен в их пространственное представление, а также для описания сцен на естественном языке или, наоборот, для синтеза изображения сцены по ее описанию.

Рассмотрим сначала *задачу описания классов* (формирования понятий). Эта задача решается в режиме обучения. Роботу последовательно предъявляют предметы из различных классов (и, возможно, в различных ракурсах) с указанием, к какому классу каждый такой предмет принадлежит. Эти предметы (а также их изображения) называют *эталонными*, а совокупность классифицированных предметов — *обучающей выборкой*. По этим данным робот должен автоматически сформировать описание классов в терминах тех свойств предметов, которые непосредственно измеряются сенсорной системой. Примерами этих свойств, которые мы будем называть первичными признаками, являются следующие: «красное», «ближе», «правее», «выше», «две точки соединены отрезком», «два отрезка параллельны», «зоны одинаковой яркости» и т. п. Таким образом, *изображения предметов и сцен задаются полным набором своих* ***первичных признаков.***

Каждому признаку поставим в соответствие предикат $\xi_i(x_1, \ldots, x_{ni})$, где $x_1, \ldots, x_{ni}$ — элементы изображения ω, определяющие наличие на нем i-го признака. Тогда каждому эталону ω_h (предмету из обучающей выборки) соответствует набор значений предикатов $\xi_i^{(h)}(c_1, \ldots, c_{hi})$, истинных на изображении данного предмета ω_h. Здесь c_i — предметные константы, означающие фиксированные части изображения. Описанием изображения эталона ω_h будем называть конъюнкцию

$$\bigwedge_{i=1}^{P_h} \xi_i^{(h)}(c_1, \ldots, c_{ni}).$$

Поскольку к одному и тому же классу могут принадлежать несколько эталонов (например, предмет из этого класса, показанный в разных ракурсах), то описанием всех эталонных изображений, принадлежащих данному классу Ω_k, является дизъюнкция

$$\bigvee_{\omega_h \in \Omega_k} \bigwedge_{i=1}^{P_h} \xi_i^{(h)}(c_1, \ldots, c_{ni}).$$

Если теперь в этой дизъюнкции все предметные константы c_i заменить на соответствующие предметные переменные x_j, то получим ППФ,

которую естественно назвать описанием класса Ω_k. Введем предикат $\sigma(k)$, означающий принадлежность изображения классу Ω_k. Тогда каждый класс Ω_k описывается аксиомой класса (АК) вида

$$\bigvee_{\omega_h \in \Omega_k} \bigwedge_{i=1}^{P_h} \xi_i^{(h)}(x_1, \ldots, x_{ni}) \to \sigma(k), \tag{1}$$

(которая по существу является логическим определением класса (или соответствующего ему понятия). Из вышеизложенного ясно, что АК вида (1) могут строиться роботом автоматически в режиме обучения по мере последовательного предъявления ему эталонов.

Практически важно, чтобы система АК обладала свойствами полноты, независимости и инвариантности в естественных смыслах. Дадим развернутое определение этих всовйств.

Систему АК будем называть *полной* на множестве изображений {ω}, если для всякого изображения ω из этого множества найдется АК, принимающая на нем значение «истинно». Следует отметить, однако, что полнота системы АК не исключает того, что для некоторого изображения могут найтись две АК, принимающие на нем значение «истинно».

В некоторых случаях требуется, чтобы ни одно исследуемое изображение не было отнесено одновременно к нескольким классам (например, если априори известно, что распознаваемые классы изображений не пересекаются). Это требование должно быть отражено в АК. Систему АК будем называть *непротиворечивой,* если существует только одна АК, истинная на любом данном изображении. Из приведенного определения следует, что непротиворечивая система АК исключает возможность пересечения классов, описываемых этой системой аксиом. Очевидно, что непротиворечивость системы АК всегда можно эффективно проверить.

Вариативность изображений, порождаемая пространственными преобразованиями воспринимаемых предметов, а также действием разного рода помех и искажений, требует, чтобы система обладала определенной инвариантностью и помехозащищенностью. Например, всевозможные изображения стола, отличающиеся от эталонного (по которому строится соответствующая АК) сдвигом, поворотом, масштабом, а также некоторыми незначительными искажениями или помехами (лишние линии, незначительные изменения пропорций и т. п.), должны описываться одной и той же АК «стол», т. е. должны классифицироваться как эквивалентные. Систему АК будем называть *инвариантной* по отношению к заданной группе преобразований, если каждая входящая в нее аксиома принимает одно и то же значение на изображениях, отличающихся преобразованиями из этой группы.

Таким образом, инвариантность системы АК позволяет «снять» охарактеризованную выше вариантность изображений и тем самым облегчить распознавание сцен по их изображениям.

Задачи распознавания и анализа изображений могут быть переформулированы как задачи логического вывода. Эти задачи решаются в режиме распознавания. Роботу предъявляются сцена, изображение $\tilde{\omega}$ которой может содержать одно или несколько изображений предметов. Эти изображения отличаются от эталонов некоторыми преобразованиями и даже могут частично перекрываться, Описание изображения сцены $S(\tilde{\omega})$ представляет собой конъюнкцию всех первичных предикатов, истинных на данном изображении $\tilde{\omega}$. В этих условиях задача отыскания на изображении $\tilde{\omega}$ изображения из определенного класса Ω_k, т. е. задача распознавания, сводится к нахождению доказательства теоремы $S(\tilde{\omega}) \rightarrow \sigma(k)$. Саму эту теорему можно трактовать как вопрос: имеется ли на данном изображении $\tilde{\omega}$ предмет из k -го класса? Ясно, что в процессе ответа на этот вопрос описание $S(\tilde{\omega})$ может быть использовано не полностью. Поэтому измерение тех или иных первичных признаков и предикатов должно производиться по мере необходимости в процессе логического вывода. Задача анализа изображения сцены заключается в распознавании на ней всех изображений предметов из различных классов. Формально эта задача сводится к последовательному доказательству теорем $S(\tilde{\omega}) \rightarrow \exists\, \sigma(k)$, $i = 1, \ldots, N - 1$, где $S_1(\tilde{\omega}) = S(\tilde{\omega})$, а $S_{i+1}(\tilde{\omega})$ получается из $S_i(\tilde{\omega})$ вычеркиванием всех предикатов, участвовавших в выводе i-й теоремы. Содержательно это означает, что, как только выделяется очередное изображение предмета из некоторого класса, оно при дальнейшем анализе не рассматривается. Полный анализ изображения сцены заканчивается распознаванием (и тем самым выделением) всех видимых изображений предметов, составляющих сцену.

В режиме обучения и распознавания на различных изображениях может встречаться один и тот же набор первичных признаков, характеризующих, например, фрагмент изображения. В таких случаях естественно ввести вторичные признаки, каждый из которых представляет собой некоторую совокупность из первичных признаков, а также вторичные предикаты, определяющие соответствующие фрагменты изображения как новые понятия.

Аксиомами обучения (АО) будем называть ППФ вида

$$\bigwedge_{j=1}^{r} \xi_j (x_{j1}, \ldots, x_{jm}) \rightarrow \alpha_i, \qquad (2)$$

где $\xi_j(x_{j1}, ..., x_{jm})$, $j = 1, ..., r$ — первичные предикаты, дающие полное описание вторичного предиката α_i, т. е. определяющие некоторый фрагмент изображения как новое понятие. Использование АО позволяет не только более экономно представить АК, но и повысить эффективность системы логического вывода в процессе распознавания и анализа.

Как мы уже отмечали, универсальным средством логического вывода в исчислении предикатов является метод резолюций. Поэтому любой конкретный алгоритм распознавания или анализа определяется стратегией метода резолюций. Важно отметить, что для распознавания на изображении сцены нужного предмета не обязательно строить доказательство теоремы $S(\tilde{\omega}) \rightarrow \sigma(k)$ полностью, т. е. перебирать все элементы искомого простого изображения на изображении сцены $\tilde{\omega}$. Вместо этого достаточно найти фрагмент искомого изображения, который содержится лишь в изображениях *k*-го класса и не содержится ни в одном изображении предметов из других классов. Именно это обстоятельство позволяет сильно ограничить число шагов логического вывода, а также распознавать частично закрытые изображения предметов.

Качество работы алгоритмов распознавания и анализа естественно характеризовать числом обращений к информационно-измерительной (сенсорной) системе с целью определения нужных признаков. Заметим, что в нашей формализации это в точности совпадает с числом шагов логического вывода, т. е. с числом резольвент, формируемых в процессе распознавания и анализа. Стратегию логического вывода будем называть *оптимальной,* если число шагов доказательства (число резольвент) минимально.

Важно, отметить, что система логического вывода способна совершенствовать алгоритмы распознавания и анализа за счет использования элементов обучения, а именно: АО и адаптивной подстройки стратегии. Адаптивная подстройка стратегии осуществляется путем ее перестройки (например, путем перестройки оптимального распознающего графа) по мере распознавания новых изображений и формирования аксиом обучения (АО). Использование АО сокращает логический вывод на длину ее описания, а процесс построения оптимального распознающего графа — на значение экспоненциальной функции от этой длины.

Проиллюстрируем описанный метод на примере формирования рекомендаций по решению задачи описания, распознавания и анализа обстановки в цехе. Предположим, что в результате предварительной фильтрации исходные изображения предметов и сцен превращаются в контурные изображения, составленные из отрезков. Введем

необходимые для дальнейшего первичные признаки и предикаты, представленные в табл. 3. Вторичные признаки и предикаты (аксиомы обучения) представлены в табл. 4.

Таблица 3

Первичный признак	Предикат	Интерпретация
Вертикальный отрезок	$ВТ\ (x,\ y)$	Истинен, если точки x и y соединены вертикальным отрезком.
Горизонтальный отрезок	$ГР\ (x,\ y)$	Истинен, если точки x и y соединены горизонтальным отрезком.
Параллельность двух отрезков	$ПРЛ\ (x,\ y,\ u,\ v)$	Истинен, если отрезки $(x,\ y)$ и $(u,\ v)$ параллельны.

Таблица 4

Вторичный признак-понятие	Символ понятия	Логическое описание понятия
Параллелограмм	$ПР\ (x_1,\ x_2,\ x_3,\ x_4)$	$ПРЛ\ (x_1,\ x_2,\ x_3,\ x_4) \wedge$ $ПРЛ\ (x_1,\ x_4,\ x_2,\ x_3)$
Горизонтальный параллелограмм	$ГПР\ (x_1,\ x_2,\ x_3,\ x_4)$	$ПР\ (x_1,\ x_2,\ x_3,\ x_4) \wedge$ $ГР\ (x_1,\ x_2) \wedge ГР\ (x_2,\ x_3)$
Вертикальный прямоугольник	$ВПР\ (x_1,\ x_2,\ x_3,\ x_4)$	$ПР\ (x_1,\ x_2,\ x_3,\ x_4) \wedge$ $ВТ\ (x_1, x_2) \wedge ГР\ (x_2,\ x_3)$

В режиме обучения роботу предъявляются эталонные контурные изображения токарного и сверлильного станков, представленные на рис. 10.

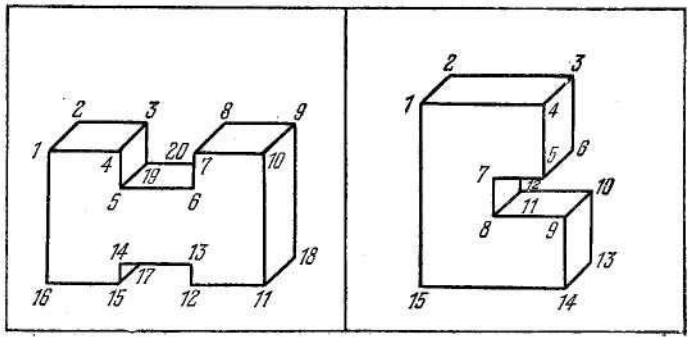


Рис. 3.10. Эталонные изображения (режим обучения).

По этим данным он строит АК, общий вид которых приведен в табл. 5.

Таблица 5

Класс-понятие	Символ класса	Логическое описание класса
Токарный станок	σ (*ТС*)	$ГПР\ (x_1, x_2, x_3, x_4) \wedge ВПР\ (x_5, x_4, x_3, x_{19}) \wedge ГР\ (x_{19}, x_{20}) \wedge ГР\ (x_5, x_6) \wedge ВТ\ (x_6, x_7) \wedge ГПР\ (x_7, x_8, x_9, x_{10}) \wedge ВПР\ (x_{11}, x_{10}, x_9, x_{18}) \wedge ГР\ (x_{12}, x_{11}) \wedge ВТ\ (x_{12}, x_{13}) \wedge ГР\ (x_{14}, x_{13}) \wedge ВТ\ (x_{15}, x_{14}) \wedge ГР\ (x_{15}, x_{17}) \wedge ГР\ (x_{16}, x_{15}) \wedge ВТ\ (x_{16}, x_1)$
Сверлильный станок	σ (*СС*)	$ГПР\ (x_1, x_2, x_3, x_4) \wedge ВПР\ (x_5, x_4, x_3, x_6) \wedge ГР\ (x_7, x_5) \wedge ВТ\ (x_8, x_7) \wedge ГПР\ (x_8, x_{11}, x_{10}, x_9) \wedge ВТ\ (x_{11}, x_{12}) \wedge ВПР\ (x_{14}, x_9, x_{10}, x_{13}) \wedge ГР\ (x_{15}, x_{14}) \wedge ВТ\ (x_{15}, x_1)$

Так в режиме обучения автоматически формируется описание классов

В режиме распознавания роботу предъявляется изображение произвольной сцены — обстановки в цехе, попавшей в «поле зрения» робота. Для определенности пусть это будет изображение сцены, представленное на рис. 11.

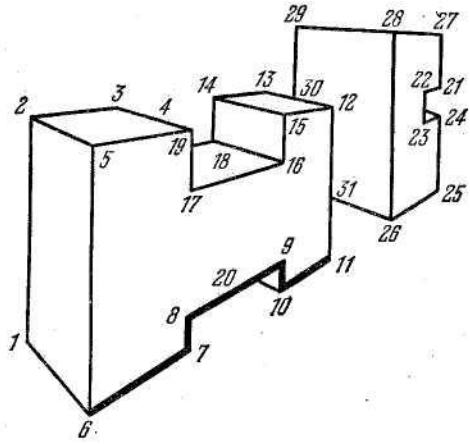


Рис. 11. Изображение сцены (режим распознавания).

Описание этой сцены (на языке первичных и вторичных предикатов) имеет вид

1. *ВПР (6, 5, 2, 1);* 2. *ГПР (5, 2, 3, 4);* 3. *ВТ (17, 4);* 4. *ГР (17, 16);* 5. *ВПР (16, 15, 14, 18);* 6. *ГР (19, 18); 7. ГПР (15, 14, 13, 12);* 8. *ВТ (11, 12);* 9. *ГР (10, 11);* 10. *ВТ (10, 9);* 11. *ГР (10, 20);* 12. *ГР (8, 9,);* 13. *ВТ (7, 8);* 14. ГР *(6, 7);* 15. *ВТ (30, 28);* 16. *ГР (29, 28);* 17. *ГР (29, 27);* 18. *ВТ (21, 27);* 19. *ГР (22, 21);* 20. *ВТ (23, 22);* 21. *ГР (23, 24);* 22. *ВТ (25, 24);* 23. *ГР (26, 25);* 24. *ВТ (26, 29);* 25. *ГР (26, 31).*

Предположим, что мы спрашиваем робота (или он сам задается этим вопросом): «есть ли на изображении сцены токарный станок?». Для ответа на вопрос робот пытается доказать теорему

$$\bigwedge_{j=1}^{25} \xi_j (c_{j1}, \ldots, c_{jk}) \rightarrow \sigma (\text{ТС}),$$

где $\xi_j (c_{j1}, \ldots, c_{jk})$, $j = 1, \ldots, 25$, — предикаты, входящие в приведенное выше описание изображения сцены.

Доказательство этой теоремы с помощью стратегии лозы потребовало в зависимости от порядка записи АК от 8 до 54 резольвент. Однако, как уже отмечалось, для распознавания токарного станка на изображении сцены полное доказательство соответствующей теоремы не обязательно, — возможно распознавание по фрагменту. Использование оптимальной стратегии в этом примере позволило сократить число резольвент до 5. При этом был автоматически выделен и существенно использовался в процессе логического вывода фрагмент изображения, обведенный на рис. 11 жирной линией. Интересно отметить, что решение той же задачи без использования АО приводит

к существенному снижению эффективностисистемы логического вывода. В этом случае описание изображения сцены требует уже не 25, а 47 первичных предикатов. В процессе распознавания токарного станка с помощью стратегии лозы строится по меньшей мере 73 резольвенты.

Резюмируя содержание п. 5.9.2.1—5.9.2.4, отметим, что *язык исчисления предикатов и адаптивная система логического вывода* позволяют эффективно решать не только задачу автоматизированного формирования рекомендаций для планирования поведения робота в условиях априорной неопределенности, но и задачи описания классов, распознавания и анализа ситуаций в окружающей робота среде.

5.9.2.5. Моделирование внешней среды

Любая система, вырабатывающая целесообразное поведение, — будь то человек, животное или АРС, — всегда использует знания о своих собственных функциональных возможностях и о свойствах внешней среды. О таких системах принято говорить, что они обладают «внутренней моделью внешнего мира». Наличие модели среды позволяет быстро и глубоко проникать в механизмы внешних явлений, намного облегчает процессы обучения на опыте и адаптации.

Каковы же особенности и способы представления знаний о внешнем мире в памяти АРС? Прежде всего заметим, что способность к моделированию внешней среды (включая свое собственное «я») присуща лишь интеллектуальным АРС.

Окружающий нас мир настолько сложен, что не только АРС, но и человек бывает доволен, если ему удается уловить и понять хотя бы некоторые самые простые из присущих этому миру закономерностей. Для этого человек строит упрощенные и идеализированные модели, освобожденные от маловажных подробностей и отражающие, как он надеется, наиболее существенные свойства рассматриваемых реальных объектов. По такой же схеме должен действовать и интеллектуальная АРС, желающая создать в себе адекватную модель внешнего мира.

Восприятие внешнего мира осуществляется с помощью искусственных органов чувств АРС. Это значит, что реальные ситуации описываются в памяти АРС с помощью набора показаний сенсорных датчиков. Именно в терминах этих показаний — элементарных высказываний о свойствах среды и самой АРС — и формируется «внутренняя модель внешнего мира».

Однако совокупность показаний сенсорных датчиков — это «сырая» информация, которую мы будем называть первичным описанием реальной ситуации. Анализ и обработка этой информации, как мы видели в предыдущем разделе, позволяют строить обобщенные

описания ситуаций, называемые понятиями. В эти информационные представления о мире могут входить не только показания органов чувств (например, «красное», «горячее», «твердое» и т. п.), но и утверждения о соотношениях между этими показаниями. Одним из наиболее удобных средств для формирования и обработки этих представлений является описанное выше исчисление предикатов; вместе с адаптивной системой логического вывода. На языке исчисления предикатов свойства внешнего мира задаются с помощью элементарных предикатов и функций от показаний сенсорных датчиков, а также с помощью логических средств формирования из этих элементов сложных утверждений (описаний) в зависимости от имеющегося опыта и поступающей информации. В двух предыдущих пунктах мы видели, каким образом осуществляется уточнение (коррекция) и расширение представлений робота о внешнем мире в процессе обучения на опыте и адаптации к существующим (заранее неизвестным) условиям.

Моделирование внешнего мира в памяти АРС не является самоцелью. Оно служит главным образом для «мысленного» планирования поведения и принятия решений о тех или иных целенаправленных процессах АРС. При этом весьма важно, чтобы в модели внешнего мира был отражен сама АРС, ее структурные и функциональные свойства, особенности взаимодействия с окружающей средой. Благодаря этому АРС получает возможность анализировать не только окружающую среду, в которой она функционирует, но и свое поведение в этой среде. Мы можем смело сказать, что интеллектуальная АРС обладает «сознанием» и «самосознанием». Уточним эти интуитивно понятные термины.

Следуя терминологии, предложенной Д. А. Поспеловым, «сознанием» АРС мы будем называть ее способность отображать (моделировать) внешнюю среду в своей памяти и анализировать закономерности этой среды, а также результаты своих воздействий на среду. Под «самосознанием» будем понимать свойство АРС отображать себя в модели среды и анализировать закономерности воздействия среды на свою структуру и функционирование. Именно «сознание» и «самосознание» принципиально отличают интеллектуальных АРС.

5.9.2.6. Алгоритмы построения программных процедур распознавания

Задача построения программных процедур распознавания (ППР) решается на пятом уровне иерархии АРС системы управления. Особенность этой задачи заключается в том, что процедуры

распознавания исполнительных механизмов АРС задаются законом изменения обобщенных координат, а происходит в реальной внешней среде. При этом допустимость тех или иных конфигураций исполнительных механизмов непосредственно наблюдается в реальной же среде и никаким простым способом не выражается «на языке» пространства обобщенных координат. Другой особенностью является наличие процедурной избыточности, присущей АРС с большим числом степеней свободы. Эта избыточность, однако, полезна. Она позволяет среди множества возможных ППР отобрать наиболее «экономные», оптимальные программные процедуры.

Рассмотрим кратко некоторые идеи и алгоритмы построения ППД. Эти алгоритмы служат для того, чтобы АРС моглаг, еще не совершая каких-либо реальных процедур, «мысленно» рассчитать, а если нужно — скорректировать программу своих целенаправленных процедур. В качестве примера рассмотрим задачу управления многозвенным манипулятором. Цель управления в этом случае заключается в распознавании (отслеживании) схватом некоторой траектории, рассчитанной на более высоком уровне планирования поведения. Задача осложняется наличием разного рода препятствий и конструктивных ограничений.

Программной процедурой распознавания называется такой закон изменения обобщенных координат, при котором достигается цель распознавания, удовлетворяются (конструктивные) ограничения и обеспечивается обход препятствий.

Идея «глобального» подхода к построению ППР заключается в следующем. Вначале на основе информации о траектории схвата с учетом конструктивных ограничений и препятствий формируется *план выполнения процедур — последовательность промежуточных процедур (конфигураций), ведущая к цели распознавания*. Затем по «кускам» (например, в классе кусочно-полиномиальных функций) строится само ППР в соответствии с этим планом.

Большой интерес представляет также «локальный» подход, основанный на моделировании так называемого тропизма.

Тропизм — это свойство, присущее простейшим живым организмам; оно заключается в направленных движениях организмов в результате действия односторонних раздражителей. Применительно к уровню построения ППР под таким *«раздражителем» будем понимать информацию о взаимном расположении АРС и препятствий в реальном трехмерном пространстве, а под тропизмом — выполнение процедур распознавания, направленные на удаление от препятствий*. Не приводя (ввиду громоздкости) явные формулы алгоритма построения ППР, имитирующего свойства тропизма, отметим только, что

синтезируемые им целенаправленные выполнение процедур распознавания реализуются за счет избыточности процедур распознавания в АРС, в то время как исполнительный (распознающий) орган движется по заданной траектории.

Мы бегло охарактеризовали различные способы построения ППР АРС. А как строить ППР колесного или гусеничного шасси для АРС, функционирующего на местности с препятствиями? Эта задача естественным образом распадается на две*: прокладка безопасного маршрута и построение закона изменения обобщенных координат, удовлетворяющего конструктивным ограничениям и обеспечивающего движение по этому маршруту.*

Рассмотрим один из подходов к построению *оптимального маршрута* на местности с препятствиями, основанный на методе динамического программирования. Для простоты предположим, что местность представляет собой плоскость, а препятствия задаются ломаными линиями. Пусть заданы координаты *исходной точки* (где находится робот) и *целевой точки* (куда он должен передвинуться). ***Задача заключается в построении (распознании) маршрута (ломаной линии) из исходной точки в целевую, который не пересекает ломаных препятствий и имеет наименьшую длину***. Такой маршрут будем называть оптимальным.

Прежде всего заметим, что если сформулированная задача не имеет тривиального решения (когда маршрутом является просто отрезок, соединяющий исходную и целевую точки), вершинами ломаной наименьшей длины должны быть вершины ломаных — препятствий. Поэтому в дальнейшем будем рассматривать только эти вершины. Введем функцию f_i определяющую минимальную длину пути из некоторой точки i в целевую точку. Сразу же возникает вопрос — как найти эту функцию? Один из классических способов нахождения функции состоит в том, чтобы задать уравнения, определяющие эту функцию.

Приступая к поиску такого уравнения, зададимся вопросом: существует ли такое внутреннее свойство процесса поиска оптимального пути, которое можно использовать для получения уравнения? Ответ, к счастью, положительный. В самом деле, анализируя задачу, мы видим, что отправляясь из точки i, робот на первом шаге попадет в точку j, пока неясно, какую именно. Далее, мы замечаем, что, если робот ищет кратчайший путь от точки i до целевой точки, то в какую бы точку он ни попал, путь из точки j в целевую должен иметь наименьшую длину. Это почти очевидное свойство докажем от противного. Пусть путь из точки i в целевую точку является кратчайшим, тогда часть пути из точки j в целевую точку

также должна иметь минимальную длину, так как, если бы эта часть пути не была кратчайшей, то ее можно было бы заменить более кратким путем и тем самым сократить общую длину пути, а это противоречит тому, что f_i по определению есть минимальная длина пути.

Таким образом, мы установили существенное внутреннее свойство процесса поиска оптимального маршрута: «хвост» (конец) процесса — это всегда кратчайший маршрут. Что все это дает для определения функции f_i? Обозначим через s_{ij} — длину пути между точками i и j (хотя все еще неизвестно, что это за точка j), а через f_j — наименьшую длину пути между точкой j и целевой точкой. Выбирая в качестве точки j такую точку, которая минимизирует сумму $s_{ij} + f_j$, получаем уравнение

$$f_i = \min_{j \neq i} [s_{ij} + f_j]. \tag{3}$$

Это — основное уравнение динамического программирования. Оно обладает тем свойством, что задает две функции f_i и $j(i)$. В самом деле, если мы нашли f_i, то значение j, на котором достигается минимум (3), как раз и указывает ту точку, в которую роботу нужно двигаться дальше. Важно отметить, что функция $j(i)$ представляет собой по существу стратегию поиска оптимального маршрута, т. е. правило, которое позволяет роботу, находящемуся в произвольной точке i, определить, куда ему следует двигаться дальше.

Трудность решения уравнения (3) заключается в том, что неизвестная функция входит в обе части равенства. В такой ситуации приходится прибегать к классическому методу последовательных приближений в функциональном пространстве, используя рекуррентную формулу

$$f_i^{(k+1)} = \min_{j \neq i} [s_{ij} + f_j^{(k)}], \tag{4}$$

где $f_i^{(k)}$ — k-е приближение искомой функции. Можно доказать сходимость алгоритма (4) для произвольной неотрицательной начальной функции f_i^0 с единственным ограничением, что значение функции f в целевой точке равно нулю.

Решение уравнения (3) можно искать и в пространстве стратегий. Этот путь представляется более естественным: «мыслить» в терминах пространства стратегий роботу более удобно, чем в терминах искусственно выбранных функций f_i. К тому же понятие стратегии сохраняет смысл и в тех случаях, когда функция не может быть определена. Какую же приближенную стратегию можно выбрать в задаче прокладки оптимального маршрута? Одна из возможных

стратегий такова: робот решает непосредственно двигаться из точки i в целевую точку. Тогда получается приближение $f_i^{(0)} = s_{i\text{ц}}$, где $s_{i\text{ц}}$ — длина пути между точкой i и целевой точкой. Следующее приближение получится, если робот будет искать решение в классе двухзвенных ломаных. Дальнейшие приближения ищутся в классе трехзвенных, четырехзвенных и т. д. ломаных.

Одно из преимуществ выбора приближений в пространстве стратегий состоит в следующем: выбрав из общих соображений стратегию, мы тут же получаем соответствующую функцию $f_i^{(k)}$. Понятно, что $f_i^{(k)} \le f_i$. В заключение, не входя в детали, отметим только, что описанный метод последовательных приближений в пространстве стратегий действительно сходится к решению уравнения (3) и допускает чрезвычайно простую программную реализацию.

После того, как оптимальный безопасный маршрут построен, можно тем или иным методом (например, с помощью кусочно-полиноминальной аппроксимации) построить программное движение колесного или гусеничного шасси робота. Алгоритмическое решение этой последней задачи обычно не вызывает никаких принципиальных затруднений.

5.8.2.7. Алгоритмы адаптивного управления движением

После того как программа процедур распознавания в АРС построена, возникает следующая задача: *сформировать рекомендации по нахождению закона управления исполнительными приводами АРС, реализующий ППР*. Решение этой задачи существенно осложняется *неопределенностью* условий функционирования АРС. Природа этой неопределенности многообразна: изменение характеристик исполнительных двигателей и механизмов (например, в результате старения и износа), технологические допуски, возможные неисправности, а также изменение условий, внешних по отношению к АРС, но оказывающих на него полностью или частично неконтролируемые воздействия. Если степень неопределенности велика, то для формирования рекомендаций по построению законов управления ППР АРС классические методы теории автоматического управления могут оказаться недостаточными. В подобных условиях рекомендации по широко используемым следящим приводам, параметры которых выбираются из априорных соображений о «средних» условиях функционирования, зачастую не обеспечивают реализацию ППР с требуемой точностью. Поэтому возникает необходимость сформировать рекомендации по использованию адаптивного управления, восполняющем недостающую информацию в процессе функционирования АРС. Синтез *адаптивного управления ППР* осуществляется в два этапа.

Вначале в предположении, что уравнения движения АРС полностью известны, рекомендуется строить закон управления, обеспечивающий близость (с любой наперед заданной точностью) реального и программного движений. Эту «неадаптивнаю» задачу можно рекомендовать решаеть методами классической теории управления. Однако воспользоваться таким «идеальным» законом управления практически нельзя, так как он зависит от ряда *варьируемых параметров* уравнения движения, значения которых неизвестны. Варьируемыми параметрами могут быть, например, масса и моменты инерции распознаваемого объекта, распределение нагрузки на шасси АРС, коэффициенты сцепления с грунтом и т. п. Поэтому на втором этапе на основе «идеального» закона рекомендуется построить адаптивное управление, обеспечивающее близость реального и ППР при любых возможных изменениях варьируемых параметров.

Идея адаптивного управления проста. Она заключается в замене неизвестных параметров «идеального» закона управления их оценками, которые должны целенаправленно «настраиваться» в процессе функционирования АРС. Алгоритмы, по которым осуществляется «настройка» параметров управления, принято называть *алгоритмами адаптации.* Многие известные в настоящее время алгоритмы адаптации с математической точки зрения представляют собой *разностные или дифференциальные уравнения*, определяющие закон целенаправленного изменения параметров управления на основе сенсорной информации.

Ярким примером эффективных алгоритмов адаптации, допускающих простую реализацию, могут служить *конечно-сходящиеся* алгоритмы решения целевых неравенств. Смысл целевых неравенств заключается в том, что если они выполнены, то закон управления, в котором вместо неизвестных параметров используется их оценка, обеспечивает требуемую близость реального и ППР. В этом случае коррекция («настройка») параметров не производится. Если же целевые неравенства в некоторый момент времени нарушаются, то осуществляется адаптивная коррекция параметров по некоторому алгоритму. Этот алгоритм может быть выбран так, что число коррекций будет конечным (и даже минимально возможным).

Описанный конечно-сходящийся процесс адаптации хорошо согласуется с содержательным представлением об адаптации. В самом деле, с интуитивной точки зрения адаптация должна проявляться в том, что в неизменяющихся («стационарных») условиях функционирования АРС (распознаваемого объекта) алгоритм адаптации «работает» не все время, а с течением времени «отключается», осуществляя переход на автоматическое управление

без «настройки» параметров. Лишь значительное изменение условий функционирования АРС (распознаваемого объекта) вызывает необходимость «включения» алгоритма адаптации для коррекции параметров управления.

6. Элементы теории марковских процессов

6.1. Введение в марковские процессы

Рассмотренные ранее динамические системы являются одним из наиболее изученных детерминистических представителей систем без последействия. Однако моделирование сколько-нибудь широкого класса распознаваемых объектов невозможно без рассмотрения вероятностных схем. Наибольшее распространение в теории систем и теории распознавания получили *марковские процессы,* представляющие собой типичную вероятностную модель систем без последействия.

Выдающуюся роль в развитии теории марковских процессов сыграли академик А. Н. Колмогоров и его ученики; теория марковских процессов располагает весьма совершенными аналитическими методами и многочисленными глубокими качественными результатами. С другой стороны, важнейшие классы распознаваемых систем допускают достаточно адекватное описание в рамках марковских процессов.

В данном изложении принят, в основном, элементарный подход к заданию марковских процессов.

Случайный процесс мы будем обозначать $\xi(t)$, где ξ принимают значения из некоторого множества Z, а t —моменты времени, $t \in T$. Сохраняя системную терминологию, конкретное значение $z(t)$, $z \in Z$, принимаемое процессом $\xi(t)$ в момент t, будем называть *состоянием* процесса в момент t, множество Z — *пространством состояний* процесса, функции $z = z(t)$ для всех— $t \in T$ траекториями или *реализациями* случайного процесса $\xi(t)$.

Будем предполагать, что при любых $t_1 < t$ существует вероятность $P(t_1, z_1, t, D)$ того, что, находясь в момент t_1 в состоянии z_1, процесс $\xi(t)$ в момент $t > t_1$ перейдет в одно из состояний, принадлежащих множеству $D \subset Z$.

Еслидля любого конечного набора $\tau_1, \tau_2, \ldots, \tau_n$, где $\tau_i < t_1$,

$i = 1, 2, \ldots, n,$

$$\mathbf{P}\{\xi(t) \in D/\xi(\tau_1) = z^{(1)}, \ldots, \xi(\tau_n) = z^{(n)}, \xi(t_1) = z_1\} =$$
$$= \mathbf{P}\{\xi(t) \in D/\xi(t_1) = z_1\} = \mathbf{P}\{t_1, z_1, t, D\},$$

то случайный процесс $\xi(t)$ называется *марковским.*

Пусть множества T и Z — конечные или счетные. В этом случае процессы $\xi(t)$ называются процессами дискретными во времени и пространстве. Пусть, например, значения случайного процесса $\xi(t)$ в моменты времени $t_i = 0, 1, 2, \ldots$ обозначаются ξ_i, а соответствующие реализации z_i. Рассмотрим условную вероятность того, что $\xi_{n+1} = z_{n+1}$ при условии

$$\xi_n = z_n, \quad \xi_{n-1} = z_{n-1}, \quad \ldots, \quad \xi_0 = z_0.$$

Обозначим ее

$$\mathbf{P}\{\xi_{n+1} = z_{n+1}/\xi_n = z_n; \quad \xi_{n-1} = z_{n-1}; \ldots; \quad \xi_0 = z_0\}.$$

Если структура случайного процесса такова, что условное распределение случайной величины ξ_{n+1} при заданном ξ_n не зависит от всех предыдущих значений ξ_i, т. е.

$$\mathbf{P}\{\xi_{n+1} = z_{n+1}/\xi_n = z_n; \ \xi_{n-1} = z_{n-1}; \ldots; \ \xi_0 = z_0\} =$$
$$= \mathbf{P}\{\xi_{n+1} = z_{n+1}/\xi_n = z_n\}, \qquad (1)$$

будем говорить, что случайный процесс $\xi(t)$ обладает *марковским свойством.*

Дискретные случайные процессы, обладающие марковским свойством, называются *цепями Маркова.*

В дальнейшем вместо состояния процесса z_i будем писать просто j. Это не ограничивает общности, поскольку элементы любого конечного или счетного множества можно закодировать целыми числами.

Обозначим $p_{ij}(n) = \mathbf{P}\{\xi_{n+1} = j/\xi_n = i\}, \quad i, \quad j = 0, 1, 2, \ldots$ При фиксированных i и j вероятность $p_{ij}(n)$ представляет собой условную вероятность того, что в момент $n + 1$ система находится в состоянии j, если известно, что в момент n система была в состоянии i. Когда $p_{ij}(n)$ зависят только от i и j (от состояний) и не зависят от времени n, будем p_{ij} называть *стационарными* вероятностями перехода из состояния i в состояние j. В таком случае цепь Маркова называется *однородной.*

Для того чтобы задать однородную марковскую цепь, кроме множеств T и Z, достаточно определить распределение начальных состояний $p(i) = \mathbf{P}\{\xi_0 = i\}$ и матрицу вероятностей перехода

$$P = \begin{vmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots \\ p_{20} & p_{21} & p_{22} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{vmatrix}. \qquad (2)$$

Наряду с вероятностями перехода p_{ij} за один шаг, представляют интерес вероятности перехода за k шагов $p_{ij}^{(k)}$. Если v — некоторое «промежуточное» состояние, $p_{iv}^{(m)}$ — вероятность перехода из состояния i в состояние v за m шагов, а $p_{vj}^{(k)}$ — вероятность перехода из состояния v в состояние j за k шагов, имеет место следующая формула:

$$p_{ij}^{(m+k)} = \sum_{v} p_{iv}^{(m)} p_{vj}^{(k)}. \qquad (3)$$

Уравнение (3) называется *уравнением Колмогорова—Чепмена.* В матричной форме это соотношение имеет вид

$$P^{m+k} = P^m P^k, \qquad (4)$$

где P^k— матрица вероятностей перехода за k шагов, совпадающая в силу (4) с k-й степенью матрицы P.

Состояние i называется *поглощающим,* если $p_{ii} = 1;\ p_{ij} = 0$ при $i \neq j$.

Пусть вероятность того, что в момент $t = 0$ система находится в состоянии i, равна $p(i)$; тогда *безусловная* вероятность того, что в момент $t = n$ система находится в состоянии j, равна

$$p^{(n)}(j) = \sum_{i=0} p(i)\, p_{ij}^{(n)}. \qquad (5)$$

Исследование поведения безусловных вероятностей $p^{(n)}(j)$ при $n \to \infty$ представляет большой интерес теории распознавания. Для цепей Маркова развит весьма эффективный аналитический аппарат.

Пусть теперь T — некоторый (конечный или бесконечный) интервал числовой прямой, множество Z совпадает с действительной прямой, а

$$F(t,\ z_1,\ \tau,\ z_2) = \mathbf{P}\{\xi(\tau) < z_2 / \xi(t) = z_1\}, \quad \tau > t, \qquad (6)$$

—переходные вероятности процесса $\xi(t)$. Функция $F(t, z_1, \tau, z_2)$ представляет собой условную функцию распределения значения процесса $\xi(\tau)$ при условии, что $\xi(t) = z_2$. Условием, выражающим марковское свойство процесса, является соотношение

$$F(t,\ z_1,\ \tau,\ z_2) = \int_{-\infty}^{+\infty} F(s,\ u,\ \tau,\ z_2)\, d_u F(t,\ z,\ s,\ u), \qquad (7)$$

которое называется *уравнением Колмогорова—Чепмена* для вещественных марковских процессов с непрерывным временем.

Если наложить дальнейшие ограничения на вид переходных функций, которые являются, по существу, ограничениями на характер траекторий марковского процесса, то можно получить различные типы марковских процессов — диффузионные (когда траектории процесса непрерывны с вероятностью 1), разрывные (когда каждая траектория является кусочно-постоянной функцией времени) и др.

Успехи в исследованиях случайных процессов вообще и марковских процессов в частности связаны с аксиоматическими построениями. В последующем нам понадобятся некоторые элементы таких построений. Поэтому целесообразно хотя бы бегло остановиться на некоторых исходных понятиях теории вероятностей и случайных процессов.

В любой задаче теории вероятностей исходят из рассмотрения определенного опыта (статистического испытания), в результате которого наступает некоторое *элементарное событие* ω *из пространства элементарных событий* Ω. Так, пусть опыт состоит в том, что регистрируются космические частицы, попадающие на заданную площадку в течение фиксированного отрезка времени. В этом случае элементарные события можно отождествлять с целыми неотрицательными числами, показывающими, сколько частиц зарегистрировано. Пространство Ω будет представлять собой множество всех целых неотрицательных чисел.

На пространстве Ω задается σ-алгебра S его подмножеств, называемых событиями. Класс S подмножеств пространства Ω называется σ-алгеброй, если выполнены следующие условия. Требуется, чтобы $\Omega \in S$ и, кроме того, чтобы операции объединения, пересечения и взятия дополнения к пространству Ω, совершаемые над элементами класса S, не выводили из класса S. При этом указанные операции можно применять, вообще говоря, счетное число раз. Таким образом, для любых множеств $A_k \in S$, $k > 1$, справедливы соотношения:

$$1)\ \ \Omega \setminus A_k \in S,\ \ k \geqslant 1,\ \ 2)\ \ \bigcup_k A_k \in S,\ \ 3)\ \ \bigcap_k A_k \in S.$$

Отсюда, в частности, следует, что пустое множество $\varnothing \in S$, поскольку Ø можно представить в виде $\varnothing = \Omega \setminus \Omega$, а $\Omega \in S$. Говорят, что событие $A \in S$ наступает, если исход статистического испытания, т. е. элементарное событие ω, принадлежит A.

Пример. Рассмотрим опыт, заключающийся в том, что на лист бумаги бросается дробинка, «не имеющая размера». Элементарным событием является возможный исход данного опыта, т. е. точка, в которую попадает дробинка при бросании. Пусть во множество событий S входит любой правильный многоугольник, лежащий естественно на листе бумаги. Тогда в это множество также должен войти любой круг, поскольку круг является счетным пересечением

множества описанных правильных прямоугольников. Внутренность круга также является событием, поскольку она оказывается счетным объединением вписанных правильных многоугольников за вычетом счетного множества их вершин, которое, в свою очередь, измеримо как счетное объединение счетных пересечений последовательностей квадратов, стягивающихся в точки.

На пространстве элементарных событий Ω вводится вероятностная мера *P(A):* каждому измеримому множеству *A(m.* е. $A \in S$) ставится в соответствие неотрицательное число *P(A)* таким образом, что $P(\Omega) = 1$; если A — объединение последовательности непересекающихся множеств A_n, входящих в S, т. е. $A = \bigcup_n A_n$, то

$$P(A) = \sum_n P(A_n). \tag{8}$$

(Формула сложения вероятностей.)

Тройка (Ω, S, P), т. е. пространство элементарных событий Ω с заданной на нем σ-алгеброй S и вероятностной мерой P, определенной на S, называется вероятностным пространством.

Пусть $\Omega_1, \Omega_2, \ldots, \Omega_n, \ldots$ — некоторая последовательность измеримых пространств, ибозначим соответствующие этим пространствам σ-алгебры измеримых множеств через $S_1, S_2, \ldots, S_n, \ldots$ Тогда можно образовать измеримое пространство Ω, точками которого будут последовательности

$$\omega = (\omega_1, \omega_2, \ldots, \omega_n, \ldots), \tag{9}$$

где $\omega_n \in \Omega_n$, $n = 1, 2, \ldots$

Класс измеримых множеств пространства Ω определяется как минимальная σ-алгебра, содержащая все множества вида

$$\{\omega = (\omega_1, \omega_2, \ldots) : \omega_k \in B_k \in S_k,\ k \leqslant N\}, \quad N = 1, 2, \ldots \tag{10}$$

(Минимальной σ-алгеброй (подмножеств заданного множества), удовлетворяющей заданному свойству, называется σ-алгебра, которая обладает указанным свойством и является подножеством любой σ-алгебры с данным свойством.)

Отметим, что множество (10) называется цилиндрическим множеством; оно определяется лишь конечным (но произвольным) числом N множеств B_k.

Определенное таким образом пространство Ω называется прямым произведением пространств $\Omega_1, \Omega_2, \ldots$

Для нас будет важно определение меры на произведении пространств, отражающее свойство независимости их элементов.

Допустим, что на пространствах Ω_n определены вероятностные меры $P_n(A_n)$, $n \geqslant 1$. Тогда на измеримом пространстве Ω можно также

ввести вероятностную меру *P(A),* обладающую следующим свойством. Если множество *A* имеет вид

$$A = \{(\omega_1, \omega_2, \ldots) : \omega_k \in A_k \in S_k, \quad i = 1, 2, \ldots, N\}, \qquad (11)$$

то

$$P(A) = P_1(A_1) \ldots P_N(A_N), \qquad (12)$$

каковы бы ни были $N \geqslant 1$, $A_i \in S_i$, $i = 1, 2, \ldots, N$. Определенная таким образом вероятностная мера P называется прямым произведением вероятностных мер $P_n(A_n)$, $n \geqslant 1$.

Все, что сказано относительно счетной последовательности пространств $\{\Omega_n\}$, сохраняется в силе и в случае конечного набора пространств.

Если вероятностная мера *P(A)* есть прямое произведение вероятностных мер $P_n(A_n)$, как было определено выше, то события A_i при различных *i,* принадлежащие различным *S,* называются независимыми в совокупности. В этом случае говорят также, что $(\omega_1, \omega_2, \ldots)$ есть последовательность независимых испытаний.

Определим еще два основных понятия теории вероятностей: случайного элемента и случайного процесса. Пусть задано пространство элементарных событий Ω с определенной на нем вероятностной мерой *P(A).* Пусть, кроме того, задано измеримое пространство *X* с σ-алгеброй измеримых множеств *S**. Случайным элементом $\xi = \xi(\omega)$ пространства *X* называется произвольная функция $\xi(\omega)$, $\omega \in \Omega$, элементарного события со значениями из *X,* если выполнено такое свойство: для любого $A^* \in S^*$ множество $\{\omega: \xi(\omega) \in A^*\}$ принадлежит *S,* где *S* — σ-алгебра измеримых множеств пространства Ω. $\{\omega : \xi(\omega) \in A^*\}$ является событием и, следовательно, можно определить его вероятность. В частности если *X* — числовая прямая, *S** — класс борелевских множеств, т. е. минимальная σ-алгебра, включающая все интервалы вида (a, b), $a < b$, $a, b \in X$, то случайный элемент называют *случайной величиной.*

Пусть *I* — конечный или бесконечный интервал прямой, которую мы назовем «осью времени». Точки этой прямой будут называться «моментами времени». Если в качестве *X* выбрать пространство функций времени t, $t \in I$, со значениями из некоторого множества *Z,* случайный элемент называется *случайным процессом.* Следовательно, случайный процесс — это функция $\xi(t, \omega)$ двух аргументов: $t \in I$ и $\omega \in \Omega$ со значениями из некоторого пространства *Z.* При этом должно удовлетворяться следующее условие. Заранее задаются «измеримые» множества A^* функций $f(t)$, $t \in I$, образующие σ-алгебру, и требуется, чтобы для каждого такого A^* множество тех ω,

при которых $\xi(t, \omega) \in A^*$, имело определенную вероятность, т. е. принадлежало S. При фиксированном $\omega \in \Omega$ функция $\xi(t,\omega)$ временного аргумента t называется *реализацией* рассматриваемого случайного процесса.

Опишем теперь, с помощью каких дополнительных предположений выделяется класс марковских случайных процессов.

Пусть Z — некоторое множество, на котором задана σ-алгебра U его подмножеств. Каждое подмножество из σ-алгебры U называется измеримым. Само множество Z назовем *пространством состояний (значений)* рассматриваемого случайного процесса $z(t, \omega)$. Для марковского процесса пару (Z, U) иногда называют фазовым пространством. Пусть (Ω, S, P) — вероятностное пространство для рассматриваемого случайного процесса. Потребуем, чтобы $z(t, \omega)$ при любом фиксированном $t \in I$, рассматриваемая как функция ω, была измерима. Иначе говоря, для любого множества $B \in U$

$$\{\omega : z(t, \omega) \in B \in U\} \in S.$$

Пусть Δ — любое числовое множество. Определим совокупность S_Δ как минимальную σ-алгебру подмножеств пространства Ω, содержащую все подмножества вида $\{\omega : z(t, \omega) \in B \in U\}$ при любых $t \in \Delta \cap I$, $B \in U$. Очевидно, все множества, входящие в S_Δ, входят также в S, поскольку S есть σ-алгебра, содержащая все множества вида

$$\{\omega : z(t, \omega) \in B \in U\}, \quad t \in I.$$

Возьмем любое фиксированное $t \in I$ и любые два множества

$$A_1 \in S_{(-\infty, t)}, \qquad A_2 \in S_{(t, \infty)}.$$

Короче говоря, A_1 — некоторое событие, связанное с поведением процесса $z(\tau)$ до момента t, а A_2 — событие, связанное с его поведением после момента t.

В теории меры показывается, что при принятых допущениях можно определить условные вероятности

$$\mathbf{P}\{A_i/z, t\} = \mathbf{P}\{\omega \in A_i/z(t) = z\}, \quad i = 1, 2,$$
$$\mathbf{P}\{A_1 A_2/z, t\} = \mathbf{P}\{\omega \in A_1 \cap A_2/z(t) = z\}. \tag{13}$$

Если для случайного процесса $z(t, \omega)$ для любых

$$A_1 \in S_{(-\infty, t)}$$

и

$$A_2 \in S_{(t, \infty)}$$

и $t \in I$

$$\mathbf{P}\{A_1, A_2/z, t\} = \mathbf{P}\{A_1/z,t\}\mathbf{P}\{A_2/z, t\}, \quad (14)$$

то случайный процесс $z(t, \omega)$ называется *марковским.*

Именно равенство (14) выражает независимость будущего течения марковского процесса от его предыстории. Марковское свойство можно выразить еще иначе. Для любых $A_1 \in S_{(-\infty, t)}$, $A_2 \in S_{(t, \infty)}$, $t \in I$

$$\mathbf{P}\{A_2/A_1,\ z(t) = z\} = \mathbf{P}\{A_2/z(t) = z\}. \quad (15)$$

В дальнейшем рассматривается применение к распознаванию систем частных случаев марковских процессов — цепей Маркова с дискретным и непрерывным временем, диффузионных процессов и марковских процессов с дискретным вмешательством случая.

Отметим, что, как правило, процессы, описывающие поведение распознаваемых систем, не являются марковскими. Однако возможны приемы, преобразующие процесс к марковскому путем введения дополнительных координат, т. е. путем включения исходного немарковского процесса в более сложный марковский (т. е. сложный марковский процесс содержит исходный немарковский в качестве компоненты).

Ниже будет приведен ряд примеров, общее назначение которых — показать, каким образом немарковский процесс превратить в марковский, расширив множество состояний. Мы надеемся охватить достаточно разнообразные примеры, чтобы читатель мог ими руководствоваться при описании распознаваемых систем.

1. Начнем с простейшего примера. Система может находиться в трех состояниях: исправном *(0),* предотказовом (1) и отказовом (2). Начиная с момента $t = 0$, исправное состояние длится случайное время ζ, распределенное по закону $F(x)$ общего вида. После этого наступает предотказовое состояние, продолжающееся время η с распределением $H(x)$. Когда происходит отказ, система навсегда остается в отказовом состоянии. Предположим, что ζ и η независимы.

Введем случайный процесс $z_0(t)$, определив его как состояние системы в момент t. График траектории процесса $z_0(t)$ изображен на рис. 1.

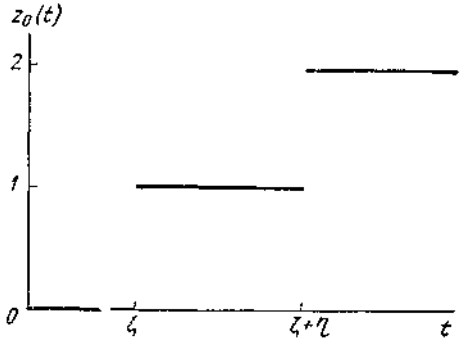


Рис. 1.

Этот процесс в общем случае не является марковским. Пусть $z_0(t) = 0$. Тогда и при всех $\tau < t$ будет $z_0(\tau) = 0$. Следовательно, вероятность любого события при условии $z_0(t) = 0$ совпадает с вероятностью этого же события при условии фиксирования всей траектории процесса до момента t. Далее, при $z_0(t) = 2$ дальнейшая история также вполне определена: $z_0(\tau) = 2$ при всех $\tau > t$. А вот при $z_0(t) = 1$ дополнительная информация о поведении траектории процесса до момента t (именно, о моменте перехода процесса в предотказовое состояние) может существенно повлиять на события, связанные с поведением процесса после момента t.

Покажем, как «марковизировать» процесс $z_0(t)$. Образуем новый процесс

$$(t) = \begin{cases} 0, & \text{если } z_0(t) = 0, \\ 2, & \text{если } z_0(t) = 2, \\ (1,\ \zeta + \eta - t), & \text{если } z_0(t) = 1. \end{cases} \tag{16}$$

Таким образом, множество состояний процесса $z(t)$ состоит из трех изолированных частей: двух точек — 0 и 2 и прямолинейного луча $(0, \infty)$, поскольку каждой точке вида $(1, x)$ можно сопоставить точку x на полупрямой.

Процесс $z(t)$ будет марковским. Действительно, при $z(t) = 0$ либо $z(t) = 2$ дальнейшая траектория процесса после момента t, как и в случае процесса $z_0(t)$, не зависит от $\{z(\tau),\ \tau < t\}$. Если же $z(t) = (1, x)$, где x — любое положительное число, то дальнейшее поведение процесса также вполне детерминировано, а именно:

$$z(t+\tau)=\begin{cases}(1,\ x-\tau) & \text{при } \tau<x,\\ 2 & \text{при } \tau\geqslant x.\end{cases} \qquad (17)$$

Можно сказать, что процесс *z(t)* вбирает в себя всю информацию о предыдущем поведении процесса $z_0(t)$, на основании которой можно делать выводы о его поведении в будущем.

Обратим внимание на то, что процесс *z(t)* не является однородным, т. е. переходные вероятности явным образом зависят от *t.* Так, легко видеть, что

$$\mathbf{P}\{z(t+\tau)=0/z(t)=0\}=\mathbf{P}\{\zeta>t+\tau/\zeta>t\}=\frac{1-F(t+\tau)}{1-F(t)}. \qquad (18)$$

Введя еще одну дополнительную координату, можно построить однородный марковский процесс. С этой целью введем процесс *z*(t)* соотношениями

$$z^*(t)=\begin{cases}(0,\ z_1(t)) & \text{при } z_0(t)=0,\\ (1,\ z_1(t)) & \text{при } z_0(t)=1,\\ 2 & \text{при } z_0(t)=2,\end{cases} \qquad (19)$$

где $z_1(t)$ теперь имеет различный смысл в зависимости от того, чему равно $z_0(t)$ а именно:

$$z_1(t)=\begin{cases}\zeta-t & \text{при } z_0(t)=0,\\ \zeta+\eta-t & \text{при } z_0(t)=1.\end{cases}$$

[Можно эти определения и объединить, сказав, что $z_1t)$ обозначает время до следующего изменения состояния процесса $z_0(t)$.]

В начальный момент времени значение процесса *z(t)* случайно и равно (0, ζ), где, как и выше, ζ - время безотказной работы системы После этого в течение целого интервала (0, ζ) траектория процесса детерминирована (рис. 2)

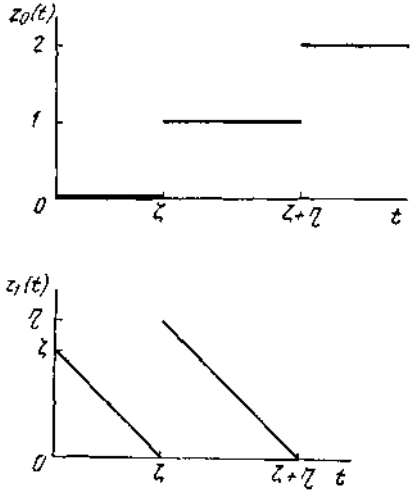


Рис. 2.

В момент $t = \zeta$ вмешивается случай, определяя, сколько времени процессу надлежит быть в предотказовом состоянии. Фиксированием значения процесса в момент ζ, т. е. (1,η), уже вполне определяется все дальнейшее течение процесса.

2. Пусть имеется бесконечная совокупность элементов, обладающих некоторым временем безотказной работы. В момент $t = 0$ включается в работу какой-нибудь из этих элементов, взятый наудачу; после его отказа включается второй и т. д. Обозначим через $N(t)$ число элементов, отказавших до момента t. Типичная траектория процесса $N(t)$ обозначена на рис. 3 жирной линией.

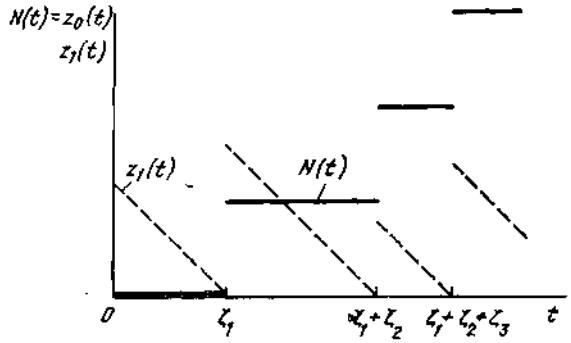


Рис. 3.

Если длительности безотказной работы различных элементов — независимые случайные величины ζ_1, ζ_2, ..., ζ_n...., то поведение процесса $N(t)$ будет описываться однородным марковским процессом

$$z(t) = (z_0(t),\ z_1(t)),$$

где $z_0(t) = N(t)$, $z_1(t)$ — время от момента t до момента следующего отказа. Траектория процесса $z_1(t)$ показана на рис. 3 пунктирной линией. Обратим внимание на то что в интервалах постоянства $z_0(t) = N(t)$ траектория процесса $z_1(t)$ ведет себя детерминированным образом: $z_1(t)$ убывает по линейному закону с угловым коэффициентом —1; при обращении $z_2(t)$ в нуль вмешивается случай сообщая значение следующего интервала постоянства $N(t)$.

Процесс вида $N(t)$ в литературе называется *процессом восстановления*, а множество моментов скачков процесса $N(t)$ - *потоком восстановления,* а также, *рекуррентным потоком* или *потоком с ограниченным последействием* Обычно предполагают, что $\zeta_2, \zeta_3, \ldots, \zeta_n, \ldots$независимы в совокупности и имеют одно и то же распределение $F(x)$, а ζ_1 имеет распределение $F_0(x)$, вообще говоря отличное от $F(x)$; ζ_1 не зависит от $\{\zeta_i,\ i \geqslant 2\}$.

3. Рассмотрим процесс восстановления с положительной длительностью включения элементов. Пусть элементы отказывают, как и в предыдущем случае, а включаются на протяжении случайного времени η, равного η_i для i-го элемента. Предположим, что все случайные величины η_i независимы в совокупности. Тогда $N(t)$ можно следующим (не единственным) образом вложить в марковский процесс

$$z(t) = (z_0(t),\ z_1(t)).$$

Обозначим

$$z_0(t) = (z_{01}(t),\ N(t)) = \begin{cases} (0,\ N(t)), & \text{если в момент } t \text{ в работе находится исправный элемент,} \\ (1,\ N(t)) & \text{если в момент } t \text{ происходит включение элемента.} \end{cases}$$

Обозначим далее

$$z_1(t) = \begin{cases} \text{времени с момента } t \text{ до момента включения нового элемента, если} \\ z_0(t) = (1, N(t)), \\ \text{времени с момента } t \text{ до момента отказа, если} \\ z_0(t) = (0, N(t)). \end{cases}$$

Короче говоря, $z_1(t)$ — время до следующего изменения состояния процесса $z_0(t)$.

Пример траектории процесса $z_1(t)$ показан на рис. 4.

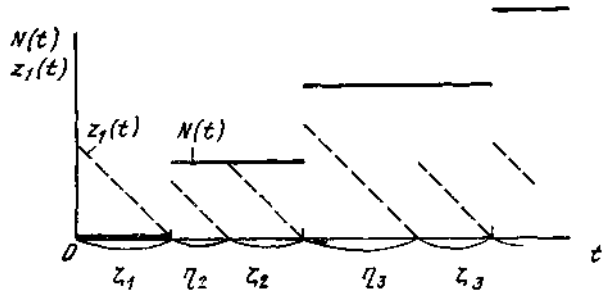


Рис. 4.

Вмешательство случая в данном примере происходит лишь в моменты достижения координатой нулевого значения. Заметим, что термин «включение» можно было бы заменить словом «восстановление»: один восстанавливаемый элемент функционирует точно так же.

6.2. Пространство состояний. Эволюция системы

Пусть имеется некоторый распознаваемый объект, который представлен физической системой *S,* которая с течением времени может менять свое состояние. Как мы знаем, состояние системы в каждый момент времени можно характеризовать набором численных значений ее параметров. Эти параметры будем называть *фазовыми координатами системы*, а состояние системы изображать в виде точки *s* с этими координатами в некотором условном *фазовом пространстве*. Тогда изменению состояния системы в процессе ее эволюции соответствует некоторая *траектория точки s в фазовом пространстве*.

Фазовое пространство может быть различным в зависимости от числа параметров, характеризующих состояние системы, и от мощности множества возможных состояний системы. В зависимости от числа параметров системы фазовое пространство может быть одномерным, двумерным и вообще *n*-мерным, (*n* — произвольное целое положительное число). Множество возможных состояний системы (т. е. множество возможных значений параметров системы и их комбинаций) может быть *конечным, счетным или несчетным*. В соответствии с этим фазовое пространство может быть *дискретным* либо *непрерывным*. Часть фазового пространства, в пределах которого реализуется процесс распознавания, будем называть *пространством, в котором осуществляется процесс распознавания.*

Процесс эволюции системы во времени также может протекать непрерывно либо дискретно. Будем говорить, что процесс в системе протекает дискретно, если состояние системы меняется лишь в определенные моменты времени, которые можно пронумеровать.

Рассмотрим случайный процесс в некоторой физической системе, протекающий в дискретном фазовом пространстве:

$$\mathcal{E} = \{0,1,2,...\},$$

причем множество моментов времени перехода системы из одного состояния в другое также дискретно, т. е.

$$T=\{0, 1, 2, ...\}.$$

Введем случайную величину $\xi_l \in \mathcal{E}$, соответствующую номеру состояния, в котором находится система в момент времени l. Обозначим через E_{li} событие, состоящее в том, что в момент времени l система находится в состоянии i, т. е. $E_{li} \sim (\xi_l = i)$.

Полное теоретико-вероятностное описание эволюции системы состоит в определении вероятности того, что для произвольных наборов $l_1, l_2, ..., l_k$ $(l_1 < l_2 < ... < l_k)$ и $i_1, i_2, ..., i_k$ имеет место событие

$$E_{l_1 l_2 \dots l_k,\, i_1 i_2 \dots i_k} = E_{l_1 i_1} \cap E_{l_2 i_2} \cap \dots \cap E_{l_k i_k} = \bigcap_{t=1}^{k} E_{l_t i_t},$$

т. е. в заданные k моментов времени система находится в заданных состояниях.

Понятно, что вероятность события

$$E_{l_1 l_2 \dots l_k,\, i_1 i_2 \dots i_k}$$

жет быть в принципе вычислена через систему условных вероятностей следующим образом:

$$P(E_{l_1 i_1} \cap E_{l_2 i_2} \cap \dots \cap E_{l_k i_k}) = P(E_{l_1 i_1})\, P(E_{l_2 i_2} \mid E_{l_1 i_1}) \dots$$
$$\dots P(E_{l_k i_k} \mid E_{l_1 i_1} \cap E_{l_2 i_2} \cap \dots \cap E_{l_{k-1} i_{k-1}}). \qquad (1)$$

Однако такое описание поведения системы является достаточно сложным. Вместе с тем существует класс случайных процессов, для которых требуемое описание может быть получено более простым путем. Это класс марковских случайных процессов.

6.3. Маркоский процесс. Цепи Маркова

Процесс, протекающий в физической системе, называется марковским (или процессом без последействия), если для каждого момента времени поведение системы в будущем зависит только от состояния системы в данный момент и не зависит от того, каким

образом система пришла в это состояние. Иначе говорят, что процесс обладает марковским свойством.

Случайный процесс с дискретным временем будем называть случайной последовательностью или случайной цепью.

Случайная цепь, для которой в каждый момент времени дальнейшая последовательность событий зависит только от состояния системы в данный момент, называется марковской цепью.

Иначе, случайная цепь, обладающая марковским свойством, называется простой марковской. Математически это означает следующее:

для любого s= (2, 3, ..., k)

$$P(\xi_{l_s}=i_s \mid \xi_{l_{s-1}}=i_{s-1},\ \xi_{l_{s-2}}=i_{s-2},\ \xi_{l_{s-3}}=i_{s-3},\ \ldots,\ \xi_{l_1}=i_1) =$$
$$= P(E_{l_s i_s} \mid E_{l_{s-1} i_{s-1}} \cap E_{l_{s-2} i_{s-2}} \cap \ldots \cap E_{l_1 i_1}) =$$
$$= P(E_{l_s i_s} \mid E_{l_{s-1} i_{s-1}}). \tag{2}$$

Тогда

$$P(E_{l_1 i_1} \cap E_{l_2 i_2} \cap \ldots \cap E_{l_s i_s}) =$$
$$= P(E_{l_1 i_1}) P(E_{l_2 i_2} \mid E_{l_1 i_1}) P(E_{l_3 i_3} \mid E_{l_2 i_2}) \ldots P(E_{l_s i_s} \mid E_{l_{s-1} i_{s-1}})$$

или

$$P\left(\bigcap_{t=1}^{s} E_{l_t i_t}\right) = P(E_{l_1 i_1}) \prod_{t=2}^{s} P(E_{l_t i_t} \mid E_{l_{t-1} i_{t-1}}).$$

Рассмотрим два смежных момента времени: l-й и (l+1)-й. Введем

$$w_{ij}(l,\ l+1) = P(E_{l+1, j} \mid E_{l, i}) \tag{3}$$

— условную вероятность перехода системы за один шаг из состояния i в момент времени l в состояние j в момент времени (l+1).

Если вероятность перехода $w_{ij}(l,\ l+1)$ не зависит от момента времени, когда осуществляется переход, для всех $(i, j) \in E$, т. е. $w_{ij}(l,\ l+1)=w_{ij}$, то соответствующая марковская цепь называется *однородной.*

Вероятность перехода $w_{ij} > 0$, если переход из состояния i в состояние j возможен за один шаг; в противном случае $w_{ij}=0$.

Совокупность вероятностей перехода w_{ij} для всех $(i, j) \in E$ образует матрицу переходов **W**, часто называемую стохастической.

Элементы матрицы переходов удовлетворяют соотношениям

$$\sum_{j\in\mathcal{E}} w_{ij} = 1, \quad i \in \mathcal{E},$$

$$w_{ij} \geqslant 0, \quad (i, j) \in \mathcal{E}.$$

Рассмотрим совокупность вероятностей $\{\mathbf{P}(\xi_l = i)\}$, $i \in \mathcal{E}$, характеризующую распределение вероятностей пребывания системы в различных состояниях в момент времени l. Введенное распределение будем называть в дальнейшем вектодом-состояний системы в момент времени l и обозначать через $\mathbf{P}(l)$, причем

$$\mathbf{P}(l) = (P_0(l)\, P_1(l)\, P_2(l) \ldots), \qquad l \in T,$$

где

$$P_i(l) = P(\xi_l = i), \quad i \in \mathcal{E}. \tag{4}$$

Вектор состояний $\mathbf{P}(0) = (P_0\ (0)\ P_1\ (0) P_2(0)\ ...)$, соответствующий распределению вероятностей состояний системы в нулевой момент времени, будем называть начальным. Понятно, что при наличии начального вектора системы $\mathbf{P}(0)$ и матрицы переходов $\mathbf{W}$ можно проследить эволюцию системы. Действительно,

$$P_j(1) = \sum_{i\in\mathcal{E}} P_i(0)\, w_{ij}, \quad j \in \mathcal{E}.$$

Это же соотношение в матричных обозначениях имеет вид

$$\mathbf{P}(1) = \mathbf{P}_2\ (0)\mathbf{W}. \tag{5}$$

Аналогично

$$\mathbf{P}(2) = \mathbf{P}(1)\mathbf{W} \tag{6}$$

или, подставляя (5) в (6),

$$\mathbf{P}(2) = \mathbf{P}(0)(\mathbf{W})^2.$$

Точно так же

$$\mathbf{P}(3) = \mathbf{P}(0)(\mathbf{W})^3$$

и, вообще,

$$\mathbf{P}\ (l) = \mathbf{P}(0)(\mathbf{W})^l. \tag{7}$$

С другой стороны, введем совокупность $w_{ij}^{(l)}$- условных вероятностей перехода системны из состояния i в состояние j за l шагов. Переход из состояния i в состояние j за l шагов может быть осуществлен различными путями. Условная вероятность $w_{ij}^{(l)}$ есть сумма вероятностей переходов из i в j для всех возможных путей. При этом, в частности,

$$w_{ij}^{(1)} = w_{ij} \quad \text{и} \quad w_{ij}^{(2)} = \sum_{\nu\in\mathcal{E}} w_{i\nu}^{(1)} w_{\nu j}^{(1)}.$$

По индукции легко показать, что

$$w_{ij}^{(l+m)} = \sum_{v \in \mathcal{E}} w_{iv}^{(l)} w_{vj}^{(m)}, \quad (i, j) \in \mathcal{E}. \tag{8}$$

Система уравнений (8) носит название уравнений Чэпмена—Колмогорова.

В матричной форме эти уравнения записываются следующим образом:

$$\mathbf{W}^{(l+m)} = \mathbf{W}^{(l)} \mathbf{W}^{(m)}.$$

Используя матрицу переходов за l шагов и начальный вектор состояния системы, легко получить вектор состояния в момент l. Тогда

$$P_j(l) = \sum_{i \in \mathcal{E}} P_i(0)\, w_{ij}^{(l)}, \quad j \in \mathcal{E},$$

или

$$\mathbf{P}(l) = \mathbf{P}(0)\mathbf{W}^{(l)} \tag{9}$$

где $(\mathbf{W})^{(l)}$ — матрица вероятностей переходов за l шагов.

Сравнивая (7) и (9), получаем

$$\mathbf{W}^{(l)} = (\mathbf{W})^{l}.$$

Таким образом, матрица переходов за l шагов равна l -й степени матрицы переходов за один шаг.

6.4. Классификация состояний

Введем несколько определений.

Состояние j *достижимо* из состояния i, если существует такое k, что $w_{ij}^{(k)} > 0$.

Подмножество C множества возможных состояний $\mathcal{E}$ называется замкнутым, если за один шаг невозможны никакие переходы из состояния, входящего в C, в состояние, не входящее в C, т. е. $w_{ij} = 0$ для всех $j \notin C$.

Цепь называется *неприводимой*, если соответствующее ей множество всех возможных состояний не содержит никаких замкнутых подмножеств, кроме самого себя.

Состояние i называется *возвратным,* если вероятность того, что система, выйдя из этого состояния, когда-либо вернется в него же, равна единице. Если же эта вероятность меньше единицы, то состояние называется *невозвратным.*

Справедлива следующая теорема.

Теорема 1. Состояние i является возвратным тогда и только тогда, когда ряд

$$\sum_{l=0}^{\infty} w^{(l)}{}_{ii}$$

расходится, т. е.

$$\lim_{r\to\infty} \sum_{l=0}^{r} w_{ii}^{(l)} = \infty.$$

Смысл высказанного в теореме утверждения становится понятным, если принять во внимание, что

$$\lim_{r\to\infty} \sum_{l=0}^{r} w_{i}^{(l)}$$

легко интерпретируется как среднее число возвращений системы в состояние *i*. В самом деле, введем случайную величину

$$\eta_{li} = \begin{cases} 0, & \text{если } \xi_l \neq i, \\ 1, & \text{если } \xi_l = i. \end{cases}$$

Тогда

$$\eta_i = \sum_{l=1}^{\infty} \eta_{ll}$$

есть, очевидно, число пребываний системы в состоянии *i* и

$$M[\eta_i] = \sum_{l=1}^{\infty} M[\eta_{li}] = \sum_{l=1}^{\infty} w_{ii}^{(l)} = \lim_{r\to\infty} \sum_{l=1}^{r} w_{ii}^{(l)}$$

есть среднее число возвращений системы в состояние *i*.

Теперь понятно, что если исходное состояние *i* является возвратным, то система с вероятностью 1 за бесконечное число шагов бесконечно много раз возвратится в *i*. Если же состояние *i* является невозвратным, то за бесконечное число шагов система с вероятностью 1 лишь конечное число раз побывает в состоянии *i*, другими словами, после некоторого конечного числа шагов она никогда больше не возвратится в *i*.

Отсюда следует, что если *i* — возвратное состояние и *j* достижимо из *i*, то и *i*, в свою очередь, достижимо из *j*. Действительно, в противном случае, выйдя из состояния *i*, система с положительной вероятностью $w^{(l)}{}_{ij} = \alpha$ попадала бы в состояние *j*, из которого *i* недостижимо; таким образом, вероятность возвращения в *i* была бы не больше чем 1— α, что противоречит возвратности *i*. Теперь понятно, что если состояние *j* достижимо из возвратного состояния *i*, то и состояние *j* является возвратным.

Наконец, нетрудно видеть, что если цепь Маркова имеет конечное число состояний, причем каждое из них достижимо из любого другого состояния, то все они являются возвратными. Действительно, если имеется лишь конечное число состояний, то за бесконечное число шагов хотя бы в одном из них система побывает бесконечное число раз, т. е. хотя бы одно из состояний системы является возвратным. Поскольку по условию из него можно с положительной вероятностью перейти в любое другое состояние, то все они являются возвратными.

Состояние i называется *поглощающим,* если вероятность ухода системы из этого состояния в любое другое равна нулю, т. е.

$$w_{ij}^{(1)} = 0, \; j \in \mathcal{E}, \; j \neq i.$$

Ясно, что если в системе имеется хотя бы одно поглощающее состояние, то ни одно из состояний системы не является возвратным.

Предположим, что в момент времени 0 система находится в произвольном, но фиксированном состоянии i, т. е. $\xi_0 = i$. Пусть $f_i^{(l)}$ — вероятность того, что первое возвращение в i произойдет в момент l. При этом ясно, что

$$\begin{aligned}
f_i^{(1)} &= w_{ii}, \\
f_i^{(2)} &= w_{ii}^{(2)} - f_i^{(1)} w_{ii}, \\
&\cdots\cdots\cdots\cdots\cdots \\
f_i^{(l)} &= w_{ii}^{(l)} - f_i^{(1)} w_{ii}^{(l-1)} - f_i^{(2)} w_{ii}^{(l-2)} - \ldots - f_i^{(l-1)} w_{ii}.
\end{aligned}$$

Сумма

$$F_i = \sum_{l=1}^{\infty} f_i^{(l)}$$

может быть интерпретирована как вероятность того, что система, выйдя из состояния i, когда-либо вернется в это состояние.

Если $F_i = 1$, то это возвращение обязательно произойдет. Введем параметр

$$\mu_i = \sum_{l=1}^{\infty} l f_i^{(l)},$$

равный математическому ожиданию времени возвращения.

Возвратное состояние i, для которого $\mu_i = \infty$, называется нулевым. Возвратное состояние, для которого возвращение возможно лишь через число шагов, кратное d, называется периодическим (с периодом d). Возвратное состояние, не являющееся ни нулевым, ни периодическим, называется *эргодическим*.

6.5. Предельный вектор

В п. 6.3 была введена вероятность $w^{(l)}_{ij}$ перехода системы из состояния i в состояние j за l шагов. Во многих задачах распознавания требуется изучить асимптотическое поведение $w^{(l)}_{ij}$ при $l \to \infty$.

Введем распределение $\{\pi_k\}$, которое будем называть *стационарным*, если

$$\pi_j = \sum_{i \in \mathcal{E}} \pi_i w_{ij}, \quad j \in \mathcal{E}. \tag{1}$$

Соотношение (1) в векторно-матричной форме имеет вид

$$\mathbf{\Pi} = \mathbf{\Pi W}, \tag{2}$$

где $\mathbf{\Pi}=(\pi_0, \pi_1, ..., \pi_k, ...)$ — стационарное распределение (распределение $\{\pi_k\}$ часто называют вектором предельных (финальных) вероятностей).

Возможность существования стационарного распределения устанавливается следующей теоремой.

Теорема 2. Если все состояния неприводимой цепи являются эргодическими, то для любой пары состояний i и j

$$\lim_{l \to \infty} w^{(l)}_{ij} = \pi_j$$

существует и не зависит от i, причем $\pi_j = 1/\mu_j$, где μ_j — среднее время возвращения в состояние j. Величины π_i удовлетворяют уравнениям

$$\pi_j = \sum_{i \in \mathcal{E}} \pi_i w_{ij}, \quad j \in \mathcal{E},$$

где

$$\sum_{j \in \mathcal{E}} \pi_j = 1.$$

Распределение $\{ \pi_i \}$ является при этом единственным. Заметим, что если множество состояний конечно и $\mathcal{E} = \{0, 1, 2, ... n\}$, то

$$\lim_{l \to \infty} \mathbf{W}^{(l)} = \begin{vmatrix} \pi_0 & \pi_1 & \pi_2 & \dots & \pi_j & \dots & \pi_n \\ \pi_0 & \pi_1 & \pi_2 & \dots & \pi_j & \dots & \pi_n \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \pi_0 & \pi_1 & \pi_2 & \dots & \pi_j & \dots & \pi_n \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \pi_0 & \pi_1 & \pi_2 & \dots & \pi_j & \dots & \pi_n \end{vmatrix}, \tag{3}$$

поэтому

$$\mathbf{\Pi} = \mathbf{P}(0) \lim_{l \to \infty} \mathbf{W}^{(l)} = (\pi_0, \pi_1, \dots, \pi_j, \dots, \pi_n)$$

и не зависит от начального распределения.

Таким образом, предельный вектор существует в том и только в том случае, если все состояния неприводимой цепи являются эргодическими.

Понятно, что если цепь имеет конечное число возможных состояний, каждое из них достижимо из любого другого состояния и не является периодическим, то предельный вектор для такой цепи существует. Если же состояния цепи являются периодическими, то $\lim_{l\to\infty} \mathbf{W}^{(l)}$ не существует, хотя стационарное распределение, удовлетворяющее (1), может существовать.

Так, например, для цепи с двумя состояниями и матрицей переходов

$$\mathbf{W} = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}$$

предельной матрицы нет, так как

$$\mathbf{W}^{(2)} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}, \quad \mathbf{W}^{(3)} = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}, \ \ldots, \ \mathbf{W}^{(2k)} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix},$$

$$\mathbf{W}^{(2k+1)} = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}, \ \ldots,$$

в то время как предельное распределение существует и

$$\Pi = \left| {}^1/_2, \quad {}^1/_2 \right|.$$

Совокупность линейных алгебраических уравнений (1), дополненная условием нормировки

$$\sum_{j\in\mathcal{E}} \pi_j = 1,$$

позволяет рассчитать компоненты предельного вектора. Важно заметить, что существование предельного распределения и численные значения его компонент не зависят, как уже было сказано, от начального распределения вероятностей состояний системы $\mathbf{P}(0)$, а целиком определяются лишь стохастической матрицей $\mathbf{W}$. Систему, для которой предельный вектор существует, будем называть эргодической.

6.6. Отображение марковской цепи в виде графа

Рассмотрим стохастическую матрицу $\mathbf{W}$ марковской цепи. Если некоторый элемент w_{ij} этой матрицы отличен от нуля, переход из состояния i в состояние j возможен за один шаг.

Множество всех состояний системы и выполнимых переходов очень удобно отображать в виде графа. Введем следующее определение. *Граф* есть совокупность вершин, соединенных направленными дугами. Соответствие между множеством состояний и возможных переходов системы, с одной стороны, и множеством вершин и дуг графа, с другой, установим следующим образом.

Множество состояний системы отображается совокупностью вершин графа, а возможные переходы системы— в виде дуг графа, причем направление дуги указывает, из какого состояния и в какое переходит система. Каждой дуге графа припишем число, равное соответствующей вероятности перехода, системы за один шаг. Таким образом, матрица переходов системы **W** и ее граф взаимно однозначно соответствуют друг другу.

Пусть, например, матрица переходов системы имеет вид

$$\mathbf{W} = \begin{vmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 1/4 & 0 & 1/4 & 1/2 \\ 1 & 0 & 0 & 0 \end{vmatrix}.$$

Соответствующий этой системе граф изображен на рис. 1.

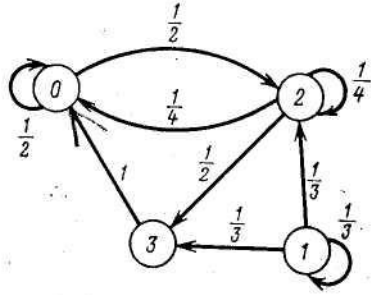


Рис. 1.

6.7. Примеры применения теории цепей Маркова

Пример 1 (случайное блуждание). Рассмотрим случайное блуждание распознаваемого элемента, при котором из каждой целочисленной точки i своего первоначального распознавания на следующем шаге с вероятностью p переходит в соседнюю справа точку $j=i+1$ своего последующего использования в распознавании и с вероятностью q — в соседнюю слева точку $j = i-1$, $(p+ q =1)$. Ясно, что в этом случае

$$w_{ij}^{(1)} = \begin{cases} p, & j = i+1, \\ q, & j = i-1, \\ 0, & (j = i) \vee (|j-i| > 1). \end{cases}$$

Матрица перехода **W** системы имеет вид

$$\mathbf{W} = \begin{vmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & q & 0 & p & 0 & 0 & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & q & 0 & p & 0 & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & 0 & q & 0 & p & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & 0 & 0 & q & 0 & p & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & q & 0 & p & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{vmatrix}.$$

Понятно, что каждое состояние этой цепи достижимо из любого другого состояния. Непосредственно можно убедиться в том, что

$$w_{ii}^{(l)} = \begin{cases} 0, & l = 2k+1, \\ C_{2k}^{k} p^k q^k, & l = 2k, \ k > 0. \end{cases}$$

Используя формулу Стерлинга, при $k \to \infty$ получаем

$$C_{2k}^{k} p^k q^k = \frac{(2k)!}{(k!)^2} p^k q^k \approx \frac{\sqrt{4\pi k}\left(\frac{2k}{e}\right)^{2k} p^k q^k}{\left(\sqrt{2\pi k}\left(\frac{k}{e}\right)^{k}\right)^2} = \frac{(4pq)^k}{\sqrt{\pi k}},$$

где $4pq = (p+q)^2 - (p-q)^2 = 1 - (p-q)^2 \leq 1$, причем знак равенства имеет место лишь при $p = q = 1/2$.
Таким образом, при $k \to \infty$

$$w_{ii}^{(2k)} \approx \frac{1}{\pi k} (4pq)^k,$$

откуда следует, что ряды

$$\sum_{k=1}^{\infty} w_{ii}^{(2k)} \quad \text{и} \quad \sum_{k=1}^{\infty} \frac{(4pq)^k}{\sqrt{\pi k}}$$

сходятся или расходятся одновременно.
Если $p \neq q$, то $4pq < 1$ и ряд

$$\sum_{k=1}^{\infty} w_{ii}^{(2k)}$$

сходится в соответствии с признаком Даламбера, так как

$$\lim_{k\to\infty}\frac{a_{k+1}}{a_k}=\lim_{k\to\infty}\frac{(4pq)^{k+1}\sqrt{\pi k}}{(4pq)^k\sqrt{\pi(k+1)}}=4pq\lim_{k\to\infty}\sqrt{\frac{k}{k+1}}=4pq<1.$$

В этом случае каждое состояние i распознаваемаго элемента является невозвратным. Интуитивно ясно, что при этом распознаваемый элемент будет неограниченно удаляться от его использования в положительном направлении (при $p>q$) или в отрицательном (при $p<q$), рано или поздно навсегда покидая любое фиксированное состояние i. Если же p—q, то $4pq=1$ и ряд

$$\sum_{k=1}^{\infty}p_{ii}^{(2k)}=\frac{1}{\sqrt{\pi}}\sum_{k=1}^{\infty}\frac{1}{\sqrt{k}}$$

расходится, так как члены этого ряда, начиная с некоторого, становятся больше соответствующих членов гармонического ряда

$$\Gamma=1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\ldots,$$

который, как известно, расходится. В этом случае все состояния распознаваемого элемента являются возвратными и распознаваемый элемент при неограниченно продолжающемся симметричном случайном использовании бесконечное число раз возвращается в каждое из состояний распознавания. Для распознаваемого элемента со случайным блужданием предельный вектор не существует, так как ни при $p=q$, ни при $p\neq q$ состояния цепи не являются эргодическими.

Пример 2. Система передачи информации о распознаваемом объекте работает в режиме, называемом нормальным, до появления сбоев в n сообщениях о распознаваемом объекте подряд. В этом случае система переходит в режим аварии, в котором остается до тех пор, пока очередное сообщение о распознаваемом объекте не будет принято правильно. После этого система возвращается в нормальный режим. Пусть вероятность появления сбоя в очередном сообщении о распознаваемом объекте равна p, а вероятность безошибочного приема сообщении о распознаваемом объекте равна $q=1$—p.

Введем множество возможных состояний системы:

E_0 — сбоев нет;

E_1 — имеется сбой в одном сообщении;

E_2 — имеется сбой в двух сообщениях подряд;

……

E_n — имеется сбой в n сообщениях подряд.

Понятно, что процесс, протекающий в такой системе, является марковским. Множество возможных состояний системы конечно и содержит $n+1$ элементов. Матрица переходов системы имеет вид

$$\mathbf{W} = \begin{vmatrix} q & p & 0 & \dots & 0 & 0 \\ q & 0 & p & \dots & 0 & 0 \\ q & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ q & 0 & 0 & \dots & p & 0 \\ q & 0 & 0 & \dots & 0 & p \\ q & 0 & 0 & \dots & 0 & p \end{vmatrix}.$$

Граф состояний и переходов изображен на рис. 1.

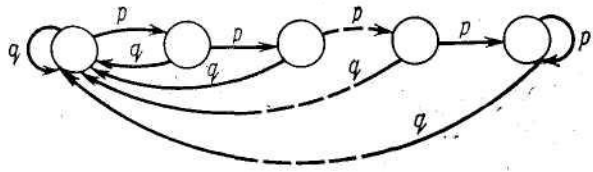


Рис. 1.

Очевидно, что каждое состояние системы достижимо из любого другого состояния. Поэтому, учитывая, что множество возможных состояний системы конечно, можно утверждать, что в этой системе существует предельный вектор.

Отыщем компоненты этого вектора, используя соотношение

$$\mathbf{\Pi} = \mathbf{\Pi W}$$

или

$$\begin{aligned} \pi_0 &= \sum_{i=0}^{n} \pi_i w_{i0}, \\ \pi_1 &= \sum_{i=0}^{n} \pi_i w_{i1}, \\ &\dots\dots\dots \\ \pi_n &= \sum_{i=0}^{n} \pi_i w_{in}. \end{aligned} \tag{1}$$

В рассматриваемом случае система уравнений (1) приобретает вид

$$\pi_0 = q\sum_{i=0}^{n}\pi_i = q,$$

$$\pi_1 = \pi_0 p = pq,$$

$$\pi_2 = \pi_1 p = p^2 q,$$

$$\cdots\cdots\cdots$$

$$\pi_{n-1} = \pi_{n-2} p = p^{n-1} q,$$

$$\pi_n = \pi_{n-1} p + \pi_n p. \qquad (2)$$

Из последнего уравнения системы (2) имеем

$$\pi_n = (p/q)\,\pi_{n-1} = p^n.$$

Таким образом, предельный вектор системы **П** имеет вид

$$\Pi = (q,\ pq,\ p^2 q,\ \ldots,\ p^{n-1}\ q,\ p^n).$$

Пример 3. На выходе приемного устройства радиолокатора в результате локации цели появляется отраженный сигнал. Из-за наложения на этот сигнал внутренних шумов приемника и изрезанности диаграммы направленности вторичного излучения цели принимаемый сигнал флуктуирует. Пусть вероятность превышения сигналом некоторого заранее выбранного порога равна p (вероятность непревышения, естественно, равна $q=1—p$).

Примем следующую логику работы устройства распознавания целей: а) цель следует считаь распознаванной, если сигнал превышает порог при двух последовательных локациях; б) цель следует считаь нераспознаванной, если сигнал не превышает порог дважды подряд.

Проанализируем работу устройства. Введем множество возможных состояний системы:

E_0 — исходное состояние. После очередной локации система должна оставаться в этом состоянии с вероятностью q и переходить в состояние E_1 с вероятностью p;

E_1 — состояние, соответствующее однократному превышению порога. Если при очередной локации сигнал будет превышать порог (вероятность этого события равна p), то система должна переходить в состояние E_2, в противном случае — должна возвращаться в исходное состояние (вероятность этого события равна q).

E_2 — состояние, соответствующее двухкратному превышению порога подряд (распознавание цели). Система должна оставаться в этом состоянии с вероятностью p и переходить и состояние E_3 с вероятностью q;

E_3 — состояние, соответствующее однократному непревышению порога после обнаружения цели. Если после очередной локации сигнал превышает порог, система должна возвратиться в состояние E_2, в

противном случае — перейти в исходное состояние E_0. Граф переходов системы приведен на рис. 2.

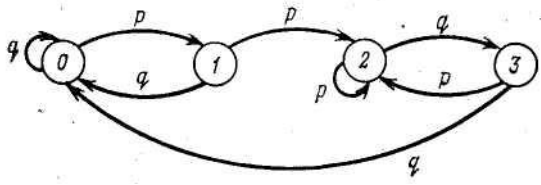


Рис. 2.

Выпишем матрицу переходов системы

$$\mathbf{W} = \begin{vmatrix} q & p & 0 & 0 \\ q & 0 & p & 0 \\ 0 & 0 & p & q \\ q & 0 & p & 0 \end{vmatrix}.$$

Поскольку множество состояний конечно и все они достижимы из j любого другого, система обладает эргодическим свойством и предельный вектор существует.

Для отыскания компонент этого вектора используем соотношение

$$\mathbf{\Pi} = \mathbf{\Pi W}.$$

Так как в рассматриваемом примере $\mathbf{\Pi}=(\pi_0, \pi_1, \pi_2, \pi_3)$, то

$$\pi_0 = \sum_{i=0}^{3} \pi_i w_{i0}, \qquad \pi_1 = \sum_{i=0}^{3} \pi_i w_{i1},$$

$$\pi_2 = \sum_{i=0}^{3} \pi_i w_{i2}, \qquad \pi_3 = \sum_{i=0}^{3} \pi_i w_{i3},$$

откуда

$$\pi_0 = q(\pi_0 + \pi_1 + \pi_3), \qquad \pi_1 = p\pi_0,$$

$$\pi_2 = p(\pi_1 + \pi_2 + \pi_3), \qquad \pi_3 = q\pi_2.$$

Решим эту систему, дополнив ее условием нормировки

$$\pi_0 + \pi_1 + \pi_2 + \pi_3 = 1,$$

решение которой дает нам следующий результат использования процесса распознавания цели

$$\pi_0 = \frac{q^2}{1-pq}; \qquad \pi_1 = \frac{pq^2}{1-pq}; \qquad \pi_2 = \frac{p^2}{1-pq}; \qquad \pi_3 = \frac{qp^2}{1-pq}.$$

6.8. Асимптотическое поведение неэргодических систем

Практический интерес в теории распознавания имеет изучение асимптотического поведения системы, не все состояния которой являются эргодическими. Рассмотрим, например, марковскую цепь, граф переходов которой изображен на рис. 1.

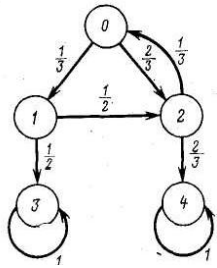


Рис. 1.

Эта цепь содержит два поглощающих состояния ($i = 3$ и $i = 4$). Понятно, что предельный вектор такой системы в том смысле, как он был введен ранее, не существует. В самом деле, состояние, в котором окажется эта система через достаточно большое число шагов, зависит от исходного. Если исходным состоянием было одно из состояний множества {0, 1, 2}, то система придет, в конце концов, в одно из двух поглощающих состояний. Если же исходным было одно из поглощающих состояний, то система навсегда в нем и останется.

Таким образом, для характеристики асимптотического поведения такой системы уже недостаточно введения предельного вектора **П**, так как он будет различным в зависимости от исходного состояния.

В связи с этим введем матрицу $\mathbf{B}=(b_{ij})$, строки которой определяли бы предельные распределения вероятностей различных состояний системы, причем номер строки указывал бы номер исходного состояния, т. е.

$$\mathbf{B}=\begin{vmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{i1} & b_{i2} & \dots & b_{in} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{vmatrix} = \begin{vmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \dots \\ \mathbf{B}_i \\ \dots \\ \mathbf{B}_n \end{vmatrix},$$

где $\mathbf{B}_i(b_{i1}b_{i2} \dots b_{in})$ — вектор распределения предельных вероятностей системы, если исходным было состояние с номером i.

Метод расчета элементов матрицы **B** состоит в следующем. Множество всех поглощающих состояний обозначим через *T,* а множество всех остальных состояний системы (они все являются невозвратными) обозначим через $\overline{T}$. Пусть множество возможных состояний системы содержит *k* поглощающих. Перенумеруем все состояния системы таким образом, чтобы поглощающим состояниям соответствовали бы последние *k* номеров, т, е.

$$\overline{T} = \{i : i \in \mathcal{E},\quad 0 \leqslant i \leqslant n-k\},$$

$$T = \{i : i \in \mathcal{E},\quad n-k+1 \leqslant i \leqslant n\}.$$

Тогда матрица переходов **W** системы будет иметь следующую структуру:

$$\mathbf{W} = \begin{vmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{O} & \mathbf{I} \end{vmatrix}.$$

Подматрица **Q** содержит вероятности переходов из невозвратных состояний в невозвратные, а подматрица **R** — вероятности переходов из невозвратных состояний в поглощающие. По аналогии с (3 п.6.5) запишем

$$\mathbf{B} = \lim_{l \to \infty} \mathbf{W}^{(l)}. \tag{1}$$

Возведя (путем непосредственного перемножения) **W** в *l*-ю степень, имеем

$$\mathbf{W}^{(l)} = \begin{vmatrix} \mathbf{Q}^{(l)} & \mathbf{B}^*_l \\ \mathbf{O} & \mathbf{I} \end{vmatrix}, \tag{2}$$

где

$$\mathbf{B}^*_l = (\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots + \mathbf{Q}^{l-1})\,\mathbf{R} = (\mathbf{I} - \mathbf{Q})^{-1}(\mathbf{I} - \mathbf{Q}^l)\,\mathbf{R}.$$

Выполнив предельный переход по *l*, в результате получим

$$\lim_{l \to \infty} \mathbf{Q}^l = \mathbf{O},$$

$$\lim_{l \to \infty} \mathbf{B}^*_l = \lim_{l \to \infty} (\mathbf{I} - \mathbf{Q})^{-1}(\mathbf{I} - \mathbf{Q}^l)\,\mathbf{R} = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R} = \mathbf{B}^*. \tag{3}$$

Таким образом, объединяя (1) — (3), можно записать, что

$$\mathbf{B} = \begin{vmatrix} \mathbf{O} & \mathbf{B} \\ \mathbf{O} & \mathbf{I}^* \end{vmatrix}.$$

Заметим, что для подматрицы **B*** выполняется следующее соотношение:

$$\mathbf{B}^* = \mathbf{R} + \mathbf{Q}\mathbf{B}^{\#}, \tag{4}$$

справедливость которого становится ясной из следующих соображений.

Пусть $i \in \overline{T}$ является начальным состоянием процесса и $j \in T$ есть некоторое поглощающее состояние. Выйдя из i, процесс может поглотиться в j на первом шаге или на одном из последующих шагов. Вероятность захвата j-м поглощающим состоянием на первом шаге равна w_{ij}. Другими исходами первого шага могут быть захват каким-либо другим поглощающим состоянием (тогда достигнуть j-го состояния будет невозможно) или переход в некоторое другое, например k-е ($k \in \overline{T}$), невозвратное состояние. В последнем случае процесс поглотится в состоянии j с вероятностью b_{kj}. Следовательно,

$$b_{is} = w_{ij} + \sum_{k \in T} w_{ik} b_{kj}.$$

Матричным аналогом этого соотношения и является (4).

Покажем теперь, что

$$\mathbf{WB}=\mathbf{B}. \qquad (5)$$

Действительно,

$$\mathbf{WB} = \begin{vmatrix} \mathbf{Q} & \mathbf{R} \\ 0 & \mathbf{I} \end{vmatrix} \begin{vmatrix} 0 & \mathbf{B}^* \\ 0 & \mathbf{I} \end{vmatrix} = \begin{vmatrix} 0 & \mathbf{QB}^* + \mathbf{R} \\ 0 & \mathbf{I} \end{vmatrix},$$

и с учетом (4) получаем требуемое.

Матричное уравнение (5) трансформируется в совокупность уравнений вида

$$\mathbf{WP}_j = \mathbf{P}_j, \quad j = 0, 1, ..., n, \qquad (6)$$

где

$$\mathbf{P}_j = \begin{vmatrix} b_{0j} \\ b_{1j} \\ \cdots \\ b_{ij} \\ \cdots \\ b_{nj} \end{vmatrix}$$

— вектор-столбец, содержащий вероятности поглощения в j-м состоянии для различных исходных состояний.

Векторно-матричные уравнения (6) решаются уже описанным выше способом, путем преобразования к системе алгебраических уравнений

$$\begin{aligned} &w_{00}b_{0j} + w_{01}b_{1j} + \ldots + w_{0n}b_{nj} = b_{0j}, \\ &w_{10}b_{0j} + w_{11}b_{1j} + \ldots + w_{1n}b_{nj} = b_{1j}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &w_{n0}b_{0j} + w_{n1}b_{1j} + \ldots + w_{nn}b_{nj} = b_{nj}. \end{aligned} \qquad (7)$$

Система (7) может быть упрощена, если учесть структуру матриц **W** и **B**. Действительно, так как

$$w_{ij} = \begin{cases} 0, & i \in T, \quad j \in T, \quad i \neq j, \\ 1, & i \in T, \quad j \in T, \quad i = j, \end{cases}$$

$$b_{ij} = \begin{cases} 0, & (j \in \overline{T}) \vee [(j \in T) \wedge (i \in T) \wedge (i \neq j)], \\ 1, & (j \in T) \wedge (i = j), \end{cases}$$

то уравнения (7) трансформируются к виду

$$\begin{aligned} & w_{00} b_{0j} + w_{01} b_{1j} + \ldots + w_{0,\,n-k} b_{n-k,\,j} + w_{0j} = b_{0j}, \\ & w_{10} b_{0j} + w_{11} b_{1j} + \ldots + w_{1,\,n-k} b_{n-k,\,j} + w_{1j} = b_{1j}, \\ & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ & w_{n-k,\,0} b_{0j} + w_{n-k,\,1} b_{1j} + \ldots + w_{n-k,\,n-k} b_{n-k,\,j} + \\ & + w_{n-k,\,j} = b_{n-k,\,j}, \end{aligned}$$

$$j = n - k + 1, \ldots, n. \qquad (8)$$

Итак, решая систему уравнений (8) для различных j ($j = n—k+1.....n$), последовательно заполняем матрицу **B**.

Как уже указывалось, строки этой матрицы задают распределения вероятностей поглощения для различных исходных состояний процесса.

Проиллюстрируем изложенную методику на примере марковской цепи, граф переходов которой изображен на рис. 1.

Пример.

$$\mathbf{W} = \begin{vmatrix} 0 & 1/3 & 2/3 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 1/3 & 0 & 0 & 0 & 2/3 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} \quad \overline{T} = \{0;\ 1;\ 2\}, \quad T = \{3;\ 4\}.$$

$$w_{00}b_{0j} + w_{01}b_{1j} + w_{02}b_{2j} + w_{0j} = b_{0j},$$
$$w_{10}b_{0j} + w_{11}b_{1j} + w_{12}b_{2j} + w_{1j} = b_{1j},$$
$$w_{20}b_{0j} + w_{21}b_{1j} + w_{22}b_{2j} + w_{2j} = b_{2j} \quad (j = 3;\ 4).$$

$j = 3$

$$\frac{1}{3}b_{13} + \frac{2}{3}b_{23} = b_{03},$$
$$\frac{1}{2}b_{23} + \frac{1}{2} = b_{13},$$
$$\frac{1}{3}b_{03} = b_{23}.$$

Решая эту систему уравнений, имеем

$$b_{03} = {}^{3}/_{13}, \quad b_{13} = {}^{7}/_{13}, \quad b_{23} = {}^{1}/_{13}.$$

$j = 4$

$$\frac{1}{3}b_{14} + \frac{2}{3}b_{24} = b_{04}$$
$$\frac{1}{2}b_{24} = b_{14}$$
$$\frac{1}{3}b_{04} + \frac{2}{3} = b_{24}.$$

В результате решения этой системы уравнений, имеем

$$b_{04} = {}^{10}/_{13}, \quad b_{14} = {}^{6}/_{13}, \quad b_{24} = {}^{12}/_{13}.$$

Таким образом, матрица **B** имеет вид

$$\mathbf{B} = \begin{vmatrix} 0 & 0 & 0 & {}^{3}/_{13} & {}^{10}/_{13} \\ 0 & 0 & 0 & {}^{7}/_{13} & {}^{6}/_{13} \\ 0 & 0 & 0 & {}^{1}/_{13} & {}^{12}/_{13} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}.$$

Хорошей проверкой правильности решения задачи является выполнение условия нормировки

$$\sum_{j=0}^{n} b_{ij} = 1$$

для строк матрицы **B**.

Заметим, что вычислительная процедура отыскания элементов матрицы **B** по объему эквивалентна k -кратному решению системы из $n—k+1$ линейных алгебраических уравнений с таким же числом неизвестных. В то же время, если иметь в виду, что исходное

состояние системы обычно фиксируется, интерес представляет лишь одна строка матрицы **B**, номер которой равен номеру исходного состояния. В связи с этим представляется целесообразной разработка метода расчета предельного распределения, не использующего громоздкую процедуру вычисления элементов **B**. С этой целью преобразуем исходную марковскую цепь в псевдоэргодическую путем введения фиктивных переходов из поглощающих состояний в начальное, приписав им некоторую вероятность α. При этом для каждого из поглощающих состояний вероятность перехода в себя должна быть уменьшена на α. С учетом этого граф переходов, например, для цепи, анализ которой проведен в рассмотренном выше примере, будет иметь вид, показанный на рис. 2.

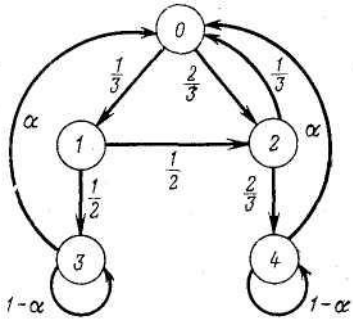


Рис. 2.

Здесь для определенности в качестве исходного выбрано нулевое состояние цепи.

Предельный вектор такой цепи уже может быть рассчитан в результате решения векторно-матричного уравнения

$$\Pi(\alpha) = \Pi(\alpha)\,\mathbf{W}(\alpha),$$

$$\mathbf{W}(\alpha) = \left|\begin{array}{c|c} \mathbf{Q} & \mathbf{R} \\ \hline \begin{matrix}\alpha\\ \alpha\\ \cdot\\ \cdot\\ \cdot\\ \alpha\end{matrix} & \begin{matrix}\\ \\ \mathbf{O} \\ \\ \\ \\ \end{matrix} \quad \mathbf{I}(1-\alpha) \end{array}\right|. \tag{9}$$

С учетом перенумерации состояний искомый вектор решения может быть записан следующим образом:

$$\Pi(\alpha) = |\,\mathbf{X}_{\text{н}}(\alpha)\,|\,\mathbf{X}_{\text{п}}(\alpha)\,|, \tag{10}$$

где $\mathbf{X}_{\text{н}}(\alpha)$ — вектор-строка, содержащий $n-k+1$ компонент, соответствующих непоглощающим состояниям цепи;

$\mathbf{X}_{\text{п}}(\alpha)$ — вектор-строка, содержащий k компонент, соответствующих поглощающим состояниям.

Покажем теперь, что компоненты предельного вектора $\mathbf{\Pi}(\alpha)$, получаемого в результате решения уравнения (9), после предельного перехода по α в точности соответствуют компонентам первой строки матрицы $\mathbf{B}$. Введем матрицы

$$\alpha = \begin{vmatrix} \alpha \\ \alpha \\ \cdot \\ \cdot \\ \cdot \\ \alpha \end{vmatrix}; \quad \mathbf{Q}_1 = \begin{vmatrix} q_{00} \\ q_{10} \\ \cdot \\ \cdot \\ \cdot \\ q_{n-k,\,0} \end{vmatrix}; \quad \tilde{\mathbf{Q}} = \begin{vmatrix} q_{01}q_{02} & \cdots & q_{0,\,n-k} \\ q_{11}q_{12} & \cdots & q_{1,\,n-k} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ q_{n-k,\,1}q_{n-k,\,2} & \cdots & q_{n-k,\,n-k} \end{vmatrix} \tag{11}$$

с размерностями, соответственно равными $k \times 1$, $(n-k+1) \times 1$, $(n-k+1) \times (n-k)$. С учетом (10) и (11) перепишем (9) в виде

$$|\mathbf{X}_{\text{н}}(\alpha)\,|\,\mathbf{X}_{\text{п}}(\alpha)| = |\mathbf{X}_{\text{н}}(\alpha)\,|\,\mathbf{X}_{\text{п}}(\alpha)| \left| \begin{array}{c|c|c} \mathbf{Q}_1 & \tilde{\mathbf{Q}} & \mathbf{R} \\ \hline \alpha & \mathbf{O} & \mathbf{I}(1-\alpha) \end{array} \right|$$

или

$$|\mathbf{X}_{\text{н}}(\alpha)\,|\,\mathbf{X}_{\text{п}}(\alpha)| = |\mathbf{X}_{\text{н}}(\alpha)\mathbf{Q}_1 + \mathbf{X}_{\text{п}}(\alpha)\alpha\,|\,\mathbf{X}_{\text{н}}(\alpha)\tilde{\mathbf{Q}}\,|\,\mathbf{X}_{\text{н}}(\alpha)\mathbf{R} + \mathbf{X}_{\text{п}}(\alpha)\mathbf{I}(1-\alpha)| =$$

$$= \left| \mathbf{X}_{\text{н}}(\alpha)\mathbf{Q}_1 + \alpha \sum_{j=n-k+1}^{n} x_j \,|\, \mathbf{X}_{\text{н}}(\alpha)\tilde{\mathbf{Q}} \,|\, \mathbf{X}_{\text{н}}(\alpha)\mathbf{R} + \mathbf{X}_{\text{п}}(\alpha)(1-\alpha) \right|,$$

откуда

$$\begin{aligned} \mathbf{X}_{\text{н}}(\alpha) &= \mathbf{X}_{\text{н}}(\alpha)\mathbf{Q} + \mathbf{A}^{T}, \\ \mathbf{X}_{\text{п}}(\alpha) &= \mathbf{X}_{\text{н}}(\alpha)\mathbf{R} + \mathbf{X}_{\text{п}}(\alpha)(1-\alpha), \end{aligned} \tag{12}$$

где

$$\mathbf{A} = \begin{vmatrix} \alpha \sum\limits_{j=n-k+1}^{n} x_j(\alpha) \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{vmatrix}$$

— вектор размерности $(n-k+1)\times 1$; T — знак транспонирования. Из первого уравнения системы (12) имеем

$$\mathbf{X}_{\text{н}}(\alpha) = \mathbf{A}^T (\mathbf{I} - \mathbf{Q})^{-1}. \tag{13}$$

Второе уравнение системы упрощается к виду

$$\mathbf{X}_{\text{п}}(\alpha) = \frac{1}{\alpha} \mathbf{X}_{\text{н}}(\alpha) \mathbf{R}. \tag{14}$$

Подставляя (13) в (14), получаем

$$\mathbf{X}_{\text{п}}(\alpha) = \frac{1}{\alpha} \mathbf{A}^T (\mathbf{I} - \mathbf{Q})^{-1} \mathbf{R} = \left| \begin{matrix} \sum\limits_{j=n-k+1}^{n} x_j(\alpha) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{matrix} \right| (\mathbf{I} - \mathbf{Q})^{-1} \mathbf{R}. \tag{15}$$

С учетом условия нормировки

$$\sum_{j=0}^{n} x_j(\alpha) = 1,$$

а также имея в виду (3), перепишем (15) следующим образом:

$$\mathbf{X}_{\text{п}}(\alpha) = \left| \begin{matrix} 1 - \sum\limits_{j=0}^{n-k} x_j(\alpha) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{matrix} \right| (\mathbf{I} - \mathbf{Q})^{-1} \mathbf{R} = \left| \begin{matrix} 1 - \sum\limits_{j=0}^{n-k} x_j(\alpha) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{matrix} \right| \mathbf{B}^* = \mathbf{B}^*_0 - \mathbf{B}^*_0 \sum_{j=0}^{n-k} x_j(\alpha), \tag{16}$$

где $\mathbf{B}^*_0$ — нулевая строка матрицы $\mathbf{B}^*$.

Если теперь принять во внимание, что

$$\lim_{\alpha \to 0} x_j(\alpha) = x_j(0) = 0, \quad j = 0, 1, 2, \ldots, n-k,$$

то окончательно имеем

$$\mathbf{X}_{\text{п}} = \lim_{\alpha \to 0} \mathbf{X}_{\text{п}}(\alpha) = \mathbf{B}^*_0; \qquad \mathbf{X}_{\text{н}} = \lim_{\alpha \to 0} \mathbf{X}_{\text{н}}(\alpha) = \mathbf{O},$$

что и требовалось,

Таким образом, предельное распределение вероятностей системы с произвольным числом поглощающих состояний может быть найдено в результате решения векторно-матричного уравнения (9) с последующим предельным переходом по α.

Заметим, что методика расчета предельного вектора не меняется, если изменить начальное состояние. Однако фиктивные переходы из поглощающих состояний в начальное необходимо ввести соответствующим этому изменению образом. Проиллюстрируем изложенную методику на примере цепи, граф которой изображен на рис. 1. Граф соответствующей псевдоэргодической цепи изображен на рис. 2.

Матрица переходов для этого графа имеет вид:

$$\mathbf{W}(\alpha)=\begin{vmatrix} 0 & 1/3 & 2/3 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 1/3 & 0 & 0 & 0 & 2/3 \\ \alpha & 0 & 0 & 1-\alpha & 0 \\ \alpha & 0 & 0 & 0 & 1-\alpha \end{vmatrix}.$$

Рассчитаем компоненты предельного вектора $\mathbf{\Pi}(\alpha)$. С этой целью решим следующую систему уравнений:

$$\begin{aligned} \pi_0(\alpha) &= 1/3\,\pi_2(\alpha)+\alpha[\pi_3(\alpha)+\pi_4(\alpha)], \\ \pi_1(\alpha) &= 1/3\,\pi_0(\alpha), \\ \pi_2(\alpha) &= 2/3\,\pi_0(\alpha)+1/2\,\pi_1(\alpha), \end{aligned} \tag{17}$$

$$\pi_3(\alpha)=\frac{1}{2}\pi_1(\alpha)+(1-\alpha)\pi_3(\alpha),$$

$$\pi_4(\alpha)=\frac{2}{3}\pi_2(\alpha)+(1-\alpha)\pi_4(\alpha). \tag{18}$$

Решая подсистему (17), имеем

$$\begin{aligned} \pi_0(\alpha) &= \frac{18}{13}\alpha[\pi_3(\alpha)+\pi_4(\alpha)], \\ \pi_1(\alpha) &= \frac{6}{13}\alpha[\pi_3(\alpha)+\pi_4(\alpha)], \\ \pi_2(\alpha) &= \frac{15}{13}\alpha[\pi_3(\alpha)+\pi_4(\alpha)]. \end{aligned} \tag{19}$$

После упрощения (18) и использования (19) получим

$$\pi_3(\alpha)=\frac{3}{13}[\pi_3(\alpha)+\pi_4(\alpha)],$$

$$\pi_4(\alpha)=\frac{10}{13}[\pi_3(\alpha)+\pi_4(\alpha)].$$

Если учесть теперь, что

$$\lim_{\alpha \to 0} [\pi_3(\alpha) + \pi_4(\alpha)] = \lim_{\alpha \to 0} \{1 - [\pi_0(\alpha) + \pi_1(\alpha) + \pi_2(\alpha)]\} = 1,$$

то искомый предельный вектор имеет вид

$$\Pi = |0 \ 0 \ 0 \ {}^3/_{13} \ {}^{10}/_{13}|.$$

Как и следовало ожидать, вычисленные компоненты предельного вектора совпадают с элементами нулевой строки матрицы **B**, полученной ранее.

6.9. Применение теории марковских цепей для оценки эффективности распознавания объекта

Процесс распознавания функционирования распознаваемого объекта определяется алгоритмом его работы, который обеспечивает приспосабливающееся к изменениям внешней среды поведение распознаваемого объекта в соответствии с логикой его алгоритма. Реакция алгоритма распознавания на ту или иную комбинацию внешних воздействий определяет эффективность распознавания объекта в этой конкретной ситуации. Оценка эффективности распознавания объекта на всем множестве возможных входных воздействий может быть получена, если будет найдено соответствующее распределение вероятностей реализации различных реакций на выходе алгоритма распознавания объекта.

Пусть множество ситуаций, каждая из которых соответствует фиксированной комбинации входных воздействий на систему, пронумеровать и образовать алфавит **A**. Аналогичным образом может быть сформировать алфавит возможных реакций системы **Ф**. Таким образом, *алгоритм — это алфавитный оператор, отображающий элементы множества **A** на элементы множества **Ф**.*

Поставим задачу отыскания распределения вероятностей реализации различных реакций на выходе алгоритма распознавания при фиксированных статистических характеристиках и структуре входных воздействий на распознаваемый объект.

Введем предварительно несколько определений. Обозначим через $E(s)$ множество состояний, в которых может оказаться распознаваемый объект в процессе функционирования, если исходным является состояние s. Тогда будем говорить, что в алгоритме распознавания имеются циклы, если существует хотя бы одно состояние s_i такое, что $s_i \in E(s_i)$. Кроме того, будем считать, что в алгоритме распознавания

имеются пересечения, если существует хотя бы одна пара s_i и s_k таких, что

$$E(s_i) \cap E(s_k) \neq \varnothing.$$

Если алгоритм распознавания имеет простую ветвящуюся структуру без циклов и пересечений, искомое распределение может быть легко получено. В самом деле, ветвящийся алгоритм распознавания без циклов и пересечений схематически может быть представлен в виде дерева (рис. 1), состоящего из узлов и соединяющих их направленных дуг.

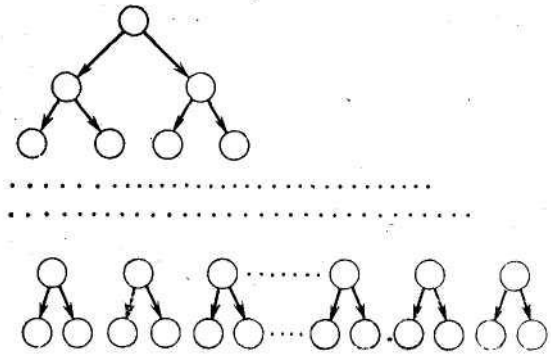


Рис. 1.

Узлам этого дерева соответствуют логические операторы алгоритма распознавания, статистика работы которых по различным ветвям соответствует содержанию входной информации о состоянии внешней среды и распознаваемого объекта и определяется априорным распределением вероятностей реализации различных вариантов входных воздействий, т. е. элементов алфавита **А**. Каждой дуге, соединяющей два каких-либо смежных узла, в соответствии с содержанием входной информации может быть приписана вероятность выполнения алгоритма распознавания именно по этой дуге. Понятно, что эта вероятность может в случае необходимости учитывать надежность элементов системы, реализующих выполнение алгоритма распознавания по выбранной дуге. Множество оконечных узлов соответствует множеству элементов алфавита **Ф**.

Если алгоритм распознавания не имеет циклов и пересечений, каждому оконечному узлу дерева, очевидно, соответствует одна и только одна ветвь, соединяющая этот узел с начальным узлом дерева. Вероятность попадания в этот конечный узел поэтому может быть определена как произведение вероятностей прохождения дуг,

образующих выбранную ветвь. Однако описанная процедура не может быть реализована, если дерево, соответствующее алгоритму распознавания, имеет циклы и пересечения. Так, например, прямой подсчет распределения вероятностей попадания в оконечные узлы затруднен для алгоритма распознавания, дерево которого изображено на рис. 2.

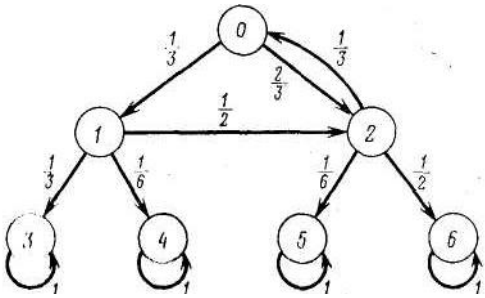


Рис. 2.

Распределение вероятностей реализации различных исходов работы алгоритма распознавания для любого распознаваемого объекта в принципе может быть получено непосредственным статистическим моделированием алгоритма распознавания функционирования распознаваемого объекта. Однако вычислительные трудности ограничивают возможности использования такого подхода для оценки эффективности распознавания объекта.

Рассмотрим в связи с этим аналитический метод отыскания закона распределения вероятностей попадания в оконечные узлы для алгоритмов распознавания, дерево которых имеет произвольную (в смысле наличия циклов и пересечений и их количества) структуру.

В общем случае (для получения распределения различных реакций алгоритма распознавания на входные воздействия) представим алгоритм распознавания как реализацию некоторого дискретного марковского процесса, т. е. в виде простой марковской цепи. При этом множеству состояний цепи поставим в соответствие множество узлов дерева алгоритма распознавания, а множеству вероятностей перехода из одного состояния в другое — множество вероятностей прохождения соответствующих дуг. Поскольку оконечные узлы дерева алгоритма распознавания соответствуют поглощающим состояниям цепи, каждому из этих состояний необходимо приписать вероятность перехода в самих себя, равную единице.

Таким образом, алгоритму распознавания функционирования объекта может быть поставлена в соответствии некоторая

неэргодическая марковская цепь, изучение поведения которой может быть проведено методами п. 6.8.

Поскольку начальное состояние алгоритма распознавания известно заранее, интерес представляет лишь одна строка матрицы **B**, номер которой равен номеру исходного состояния. Приведем без пояснений результат решения задачи по отысканию матрицы **B** для алгоритма распознавания, граф которой изображен на рис. 2:

$$\mathbf{B} = \begin{vmatrix} 0 & 0 & 0 & 2/13 & 1/13 & 5/26 & 15/26 \\ 0 & 0 & 0 & 14/39 & 7/39 & 3/26 & 9/26 \\ 0 & 0 & 0 & 8/39 & 4/39 & 3/13 & 9/13 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}.$$

Знание распределения вероятностей реализации различных реакций алгоритма распознавания может быть использовано для оценки эффективности распознавания объекта. Итак, пусть вектор **P** представляет собой закон распределения вероятностей различных реакций распознаваемого объекта на внешние воздействии. При наличии модели распознаваемого объекта нетрудно оценить эффективность распознавания объекта для каждой из реакций алгоритма распознавания. Обозначим соответствующее множество оценок через

$$R = \{r_1, r_2, \ldots, r_m\}.$$

Тогда легко рассчитать $M[r]$ — математическое ожидание эффективности распознавания объекта на всем множестве входных воздействий

$$M[r] = \sum_{l=1}^{m} r_l P_l. \tag{1}$$

Знание компонент вектора **P** позволяет определить и другие оценки эффективности распознавания объекта, например вероятность того, что эффективность распознавания объекта находится в заданных пределах:

$$\text{Вер}\{r_{min} \leqslant r \leqslant r_{max}\} = \sum_{i \in I_0} P_i,$$
$$I_0 = \{i : i \in [1, m], \quad r_{min} \leqslant r \leqslant r_{max}\}, \tag{2}$$

или вероятность того, что эффективность распознавания объекта не ниже заданной:

$$\text{Вер}\{r_{\text{зад}} \leqslant r\} = \sum_{i \in I_1} P_i,$$
$$I_1 = \{i : i \in [1, m],\ r_{\text{зад}} \leqslant r_i\}, \tag{3}$$

или дисперсию случайного значения эффективности распознавания объекта

$$Д[r] = \sum_{i=1}^{m} (r_i - M[r])^2 P_i = \sum_{i=1}^{n} r_i^2 P_i - \left(\sum_{i=1}^{n} r_i P_i\right)^2.$$

7. Конечные автоматы

7.1. Общие понятия теории конечных автоматов

1. Вводные замечания. Из сказанного выше следует, что для описания динамических детерминистических распознаваемых объектов могут быть использованы булевы и высказывательные функции. Однако существует класс систем (называемых алфавитными преобразователями информации), для описания которых оказывается весьма полезной специальная математическая схема (структура) – конечные автоматы. Теория конечных автоматов — частный случай общей математической теории систем. Она не изучает реальных физических устройств и явлений, а рассматривает абстрактные математические модели и их общие свойства. Однако на конечные автоматы можно смотреть как на идеализированные модели многих распознаваемых устройств и явлений. Идеи и методы, развитые в теории конечных автоматов, широко используются в теории распознавания.

Под алфавитом будем понимать конечное множество объектов, называемых буквами. В качестве букв можно рассматривать объекты любой природы: буквы алфавита русского или латинского языка, цифры, знаки, рисунки, слова, фразы и т. д.

Словом в данном алфавите называется конечная упорядоченная совокупность букв. Например, в алфавите *(x, y)* словами будут *xxy, xyx, yxxyx* и т. д. Длина слова измеряется числом содержащихся в нем букв. Употребляется также и *пустое* слово, не содержащее ни одной буквы; оно обозначается 0.

В дальнейшем существенную роль играют соответствия между словами в одном и том же или различных алфавитах. *Алфавитным оператором* или *алфавитным отображением* называется

соответствие, сопоставляющее словам в данном алфавите слова в том же самом или в некотором другом алфавите. В последнем случае различают *входной* и *выходной* алфавиты оператора и соответственно входные и выходные слова.

Алфавитный оператор является однозначным, если он каждому входному слову ставит в соответствие не более одного выходного. Если данному входному слову алфавитный оператор не сопоставляет никакого выходного слова, то он не определен на этом слове. Совокупность всех слов, на которых алфавитный оператор определен, называется его областью определения.

Два алфавитных оператора считаются *равными,* если они имеют одну и ту же область определения и относят любому слову из этой области одинаковые выходные слова. Алфавитный оператор может быть задан при помощи таблицы соответствия, которая для каждого входного слова содержит определенное выходное слово или правило, следуя которому для каждого входного слова может быть установлено выходное слово (если оно существует).

Алфавитные операторы, задаваемые с помощью конечных систем правил, позволяющих за конечное число шагов для любого входного слова, входящего в область определения, найти соответствующее выходное слово, называются *алгоритмами.* Два алгоритма считаются равными, если равны реализуемые ими алфавитные операторы, а также совпадают системы правил, задающие действие этих операторов на входные слова. Алгоритмы, у которых совпадают только реализуемые ими алфавитные операторы, но, вообще говоря, не совпадают способы задания, будем называть *эквивалентными.*

Важность для практики алфавитных операторов связана в первую очередь с тем, что любой реальный преобразователь информации может рассматриваться (при некоторых весьма общих условиях) как прибор, реализующий тот или другой алфавитный оператор. Обратим внимание на алфавитные операторы, осуществляющие побуквенное отображение входных слов в выходные слова; они имеют широкое практическое распространение наряду с другими алфавитными операторами.

Перейдем к определению конечного автомата. Вначале задается множество моментов времени $t_0 < t_1 < t_2 < ...$, в каждый из которых конечный автомат находится в некотором состоянии. Состояние автомата в момент t_n обозначается $z\ (t_n)$ или более кратко, $z(n)$. Множество состояний автомата будем обозначать Z. Предполагается, что это множество конечно, откуда и название «конечный автомат», в каждый момент t_i автоматного времени, начиная с t_1, на вход автомата поступает в качестве входного сигнала одна из букв x входного

алфавита *X*, т. е. $x \in X$. Конечные упорядоченные совокупности входных сигналов *x* (1), *x* (2), ..., *x (k)* называются входными словами. Автомат следующим образом реагирует на поступление входных сигналов. Во-первых, внутреннее состояние автомата изменяется в соответствии с *функцией переходов* (в новое состояние):

$$z(t) = \varphi[z(t\text{-}1), x(t)].$$

Во-вторых, в каждый момент автоматного времени на выходе автомата появляется выходной сигнал *y,* определяемый *функцией выходов:*

$$y(t)=\psi[z(t\text{-}1), x(t)].$$

Любое допустимое входное слово (из фиксированного множества допустимых слов) вызывает появление на выходе автомата выходного слова такой же длины. Получаемое таким образом соответствие между допустимыми входными и выходными словами называется отображением, индуцируемым данным автоматом.

Укажем ограничения, налагаемые на класс конечных автоматов. Предполагают, что конечный автомат имеет конечное число входов и выходов. В момент времени *t* на каждый вход конечного автомата поступают сигналы и на его выходах появляются сигналы, являющиеся его реакцией на эти входные сигналы. Считают, что смена сигналов происходит в дискретные моменты времени $t_0, t_1, \ldots$ и отождествляют их с целыми неотрицательными числами. Тогда переменная времени *t* принимает значения 0, 1, 2, ... Этот класс моделей не рассматривает физическую природу сигналов. Он предполагает, что в момент времени *t* на каждый вход конечного автомата поступает только один сигнал, и на выходе вырабатывается тоже только один сигнал. Число входных и выходных сигналов конечно. Будем обозначать входы конечного автомата переменными $x_1, x_2, \ldots, x_n$, а его выходы $y_1, y_2, \ldots, y_m$. На рис. 1 схематически изображен конечный автомат $\mathfrak{M}$.

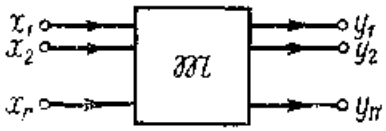


Рис. 1.

Входные сигналы входа x_i зададим буквами входного алфавита X_i $(i=1, \ldots, n)$. Выходные сигналы выхода y_j зададим буквами выходного алфавита Y_j $(j=1, \ldots, m)$. Из требования конечности множеств входных и выходных сигналов вытекает, что все алфавиты $X_1, \ldots, X_n, Y_1, \ldots, Y_m$ — конечны. Если конечный

автомат $\mathfrak{M}$ имеет n входов $x_1, \ldots, x_n$, на которые подаются входные сигналы из алфавитов $X_1, \ldots, X_n$, то без потери общности его можно рассматривать как имеющий один вход x, на который поступают сигналы алфавита $X_1 \times \ldots \times X_n$. Аналогично обстоит дело с выходом.

В зависимости от потребностей можно рассматривать конечный автомат, имеющий как один вход или выход, так и несколько.

2. Поведение конечного автомата. Состояния. Под поведением автомата понимают его способность преобразовывать буквы входного алфавита в буквы выходного алфавита. Предположим, что автомат имеет один вход и один выход. Будем считать, что поведение автомата описано, если указана функциональная зависимость выходного сигнала от входного в момент времени t. Выход в момент t определяется не только входом, но и всей предыдущей работой автомата или памятью о предыдущей работе. Можно сказать, что существуют некоторые внутренние переменные, от которых зависит выход в момент t. Будем учитывать суммарное влияние этих переменных, вводя понятие - некоторой величины — состояния, которая позволяет по входному сигналу и значению этой величины определить однозначно выходной сигнал. Еще одним ограничением, налагаемым на класс рассматриваемых моделей, является конечность числа состояний автомата. Теперь дадим точное математическое определение конечного автомата.

Конечным автоматом $\mathfrak{M}$ называется совокупность пяти объектов:

1) конечного непустого множества $X = \{a_1, \ldots, a_p\}$, называемого *входным алфавитом* автомата $\mathfrak{M}$, его элементы называются входными символами-буквами;

2) конечного непустого множества $Y = \{b_1, \ldots, b_t\}$, называемого *выходным алфавитом* автомата $\mathfrak{M}$, его элементы называются выходными символами-буквами;

3) конечного непустого множества $Z = \{q_1, \ldots, q_s\}$, называемого *множеством состояний* автомата $\mathfrak{M}$, его элементы называются состояниями;

4) *переходной функции состояний f,* отображающей множество всех упорядоченных пар (a_j, q_i) в множество Z;

5) *выходной функции g,* отображающей множество всех упорядоченных пар (a_j, q_i) в множество У.

Таким образом, конечный автомат $\mathfrak{M}$ — пятерка $<X, Y, Z, f, g>$. В каждый момент t на вход автомата $\mathfrak{M}$ поступает одна из букв $x(t)$ алфавита X, при этом в тот же самый момент времени t на выходе появляется одна из букв $y(t)$ алфавита Y, а состояние автомата $z(t)$ меняется.

Исходя из 4-го и 5-го пунктов определения конечного автомата имеем соотношения

$$z(t+1)=f\left(x(t),\ z(t)\right),$$
$$y(t)=g\left(x(t),\ z(t)\right), \tag{1}$$

где $z(t)$ и $z(t+1)\in Z$, $x(t)\in X$, $y(t)\in Y$. Если дано $z(0)$ —начальное состояние конечного автомата $\mathfrak{M}$, то последовательность l входных букв (или, как говорят, входное слово длины l) однозначно определит в силу (1) последовательность состояний и выходное слово той же самой длины. Конечный автомат $\mathfrak{M}$, для которого указано $z(0)$, называется *инициальным автоматом.*

Пример 1. Дан последовательный двоичный сумматор с двумя входами, т. е. устройство, на входы которого поступают две последовательности двоичных цифр, причем каждая последовательность представляет собой число в двоичной записи, последовательность на выходе есть сумма двух чисел, подаваемых на входы. Показать, что это устройство можно рассматривать как конечный автомат.

В момент времени t на входы устройства, их два, поступают сигналы, соответствующие символам 0 или 1, следовательно, множество $X=\{00,\ 01,\ 10,\ 11\}$ является входным алфавитом. На выходе устройства появляется сигнал, соответствующий 0 или 1, поэтому множество $Y=\{0,1\}$ — его выходной алфавит. Выходной сигнал определяется входным сигналом и переносом, поэтому множество $Z=\{q_0$ — нет переноса, q_1 — есть перенос$\}$ — множество состояний и функции выхода $g(x(t),\ z(t))$..... и перехода $f(x(t),\ z(t))$ определяются следующим образом:

$$f(00,\ q_0)=q_0;\ f(01,\ q_0)=q_0;\ f(10,\ q_0)=q_0;\ f(11,\ q_0)=q_1;$$
$$f(00,\ q_1)=q_0;\ f(01,\ q_1)=q_1;\ f(10,\ q_1)=q_1;\ f(11,\ q_1)=q_1;$$
$$g(00,\ q_0)=0;\ g(01,\ q_0)=1;\ \ g(10,\ q_0)=1;\ \ g(11,\ q_0)=0;$$
$$g(00,\ q_1)=1;\ \ g(01,\ q_1)=0;\ \ g(10,\ q_1)=0;\ \ g(11,\ q_1)=1.$$

Изоморфизм конечных автоматов

Автоматы $\mathfrak{M}_1=\langle X_1,\ Y_1,\ Z_1,\ f_1,\ g_1\rangle$ и $\mathfrak{M}_2=\langle X_2,\ Y_2,\ Z_2,\ f_2,\ g_2\rangle$ называют *изоморфными,* если:

1) $X_1=X_2=X$;

2) можно установить взаимно однозначное соответствие между Z_1 и Z_2 так, что если $q_1\in Z_1$ соответствует $q_2\in Z_2$, то $\forall a_i\,(a_i\in X)\ f_1(a_i,\ q_1)$ соответствует $f_2(a_i,\ q_2)$ и $g_1(a_i,\ q_1)=g_2(a_i,\ q_2)$.

Таким образом, автоматы $\mathfrak{M}_1$ и $\mathfrak{M}_2$ отличаются только метками состояний. В дальнейшем конечные автоматы будут интересовать нас с точностью до изоморфизма.

3. Классификация конечных автоматов. В зависимости от мощности множеств X, Y, Z и вида функций f и g различаются следующие виды автоматов.

1. Автомат без памяти (комбинационная схема). В этом случае множество Z состоит из одного элемента, т. е. автомат есть тройка $<X, Y, g>$. Рекуррентные соотношения (1) вырождаются в соотношение

$$y(t) = g(x(t)). \tag{2}$$

2. Автономный автомат. Множество X состоит из одного элемента, т. е. автомат $\mathfrak{M}$ есть четверка $<Y, Z, f, g>$. Рекуррентные соотношения (1) вырождаются в соотношения

$$\begin{gathered} z(t+1) = f(z(t)), \\ y(t) = g(z(t)). \end{gathered} \tag{3}$$

3. Автомат без выхода. Множество Y состоит из одного элемента, т. е. автомат $\mathfrak{M}$ есть тройка $<X, Z, f>$. Рекуррентные соотношения (1) вырождаются в соотношение

$$z(t+1) = f(x(t),\ z(t)). \tag{4}$$

4. Автомат с задержкой. Функция g зависит только от состояния $z(t)$, соотношения (1) имеют вид

$$\begin{gathered} z(t+1) = f(x(t),\ z(t)), \\ y(t) = g(z(t)). \end{gathered} \tag{5}$$

5. Автомат Мура. Функция g зависит от состояния $z(t+1)$, соотношения (1) имеют вид

$$\begin{gathered} z(t+1) = f(x(t),\ z(t)), \\ y(t) = g(z(t+1)). \end{gathered} \tag{6}$$

4. Задание конечных автоматов таблицами и графами.

Функции перехода f и выхода g, поскольку области определения и изменения их конечны, могут быть заданы таблицами, которые называются *таблицей переходов* и *таблицей выходов* соответственно. Построение таблиц переходов и выходов показано в табл. 1 и 2 соответственно

Таблица 1

$x(t)$ / $z(t)$	$z(t+1)$			
	a_1	a_2	...	a_p
q_1				
q_2				
⋮				
q_s				

Таблица 2

$x(t)$ / $z(t)$	$y(t)$			
	a_1	a_2	...	a_p
q_1				
q_2				
⋮				
q_s				

Строки таблиц занумерованы состояниями, а столбцы входными символами. В таблице переходов на пересечении q_i строки и a_j столбца стоит $f(a_j, q_i)$, а в таблице выходов на пересечении q_i строки и a_j- столбца стоит $g(a_j, q_i)$.

Пример 2. Построить таблицы переходов и выходов для последовательного двоичного сумматора, рассмотренного в примере 1. Таблицы переходов и выходов имеют вид (табл. 3 и 4).

Таблица 3

$x(t)$ / $z(t)$	$z(t+1)$			
	00	01	10	11
q_0	q_0	q_0	q_0	q_1
q_1	q_0	q_1	q_1	q_1

Таблица 4

$x(t)$ / $z(t)$	$y(t)$			
	00	01	10	11
q_0	0	1	1	0
q_1	1	0	0	1

Табличное задание конечных автоматов удобно использовать при мощностных оценках. Если фиксировать алфавиты *X, Y* и множество Z, то таблица переходов может быть заполнена s^{ps}способами, а таблица выходов t^{ps}. Следовательно, общее число автоматов с заданными алфавитами будет равно $(st)^{rs}$.

Другим способом задания конечного автомата является задание с помощью ориентированного графа, называемого *графом переходов.* Граф переходов автомата $\mathfrak{M}$ имеет *s* вершин, которым приписаны *s* состояний автомата $\mathfrak{M}$. Если $\{a_{i_1}, \ldots, a_{i_r}\}$—множество входных символов таких, что $f(a_{i_\nu}, q_l) = q_j$ и $g(a_{i_\nu}, q_l) = b_{i_\nu}$ $(\nu = 1, \ldots, r)$, то существует ребро, идущее из вершины q_i в вершину q_j и этому ребру поставлено в соответствие выражение $a_{i_1}/b_{i_1} \vee \ldots \vee a_{i_r}/b_{i_r}$ (если

множество $\{a_{i_1}, \ldots, a_{i_r}\}$ пусто, то таких ребер нет). Задание конечного автомата графом является более наглядным.

Пример 3. Построить граф перехода для последовательного двоичного сумматора, рассмотренного в примере 1.

Из описанного построения графа переходов следует, что граф последовательного двоичного сумматора имеет вид, представленный на рис. 2.

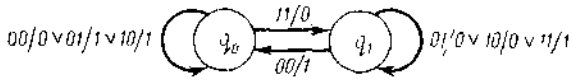


Рис. 2.

7.2. Эквивалентные состояния. Минимальная форма конечного автомата

1. Эквивалентные состояния. Состояние q_i называется *достижимым из состояния* q_j, если существует входное слово, под действием которого автомат, находящийся в состоянии q_j, перейдет в состояние q_i.

Теорема 1. *Пусть автомат* $\mathfrak{M}$ *имеет* s *состояний и состояние* q_i *достиюимо из* q_j. *Тогда состояние* q_i *достижимо из* q_j *входным словом длины* s—1 *или меньше.*

Доказательство. Если состояние q_i достижимо из состояния q_j, то в графе переходов автомата $\mathfrak{M}$ существует путь, ведущий из вершины q_j в вершину q_i. Опуская в нем все циклы, получаем путь из q_j в q_i, который проходит через любое состояние не более чем один раз. Так как число состояний в автомате s, то ребер в пути не больше чем s—1.

Состояние q_1 автомата $\mathfrak{M}_1 = <X, Y, Z_1, f_1, g_1>$ называется *k-эквивалентным состоянию* q_2 автомата $\mathfrak{M}_2 = <X, Y, Z_2, f_2, g_2>$, если при действии всякого входного слова длины k или меньше на автомат $\mathfrak{M}_1$ в состоянии q_1 и на автомат $\mathfrak{M}_2$ в состоянии q_2 получаются одинаковые выходные слова. В этом случае пишут $q_1 \approx_k q_2$.

В противном случае состояния называют k различимыми.

Пример 1. Дан автомат $\mathfrak{M}$ с входным алфавитом $X = \{\alpha, \beta\}$, выходным алфавитом $Y = \{0, 1\}$ и множеством состояний $Z = \{1,2, 3,4\}$, его граф перехода дан на рис. 1. Показать, что его состояния 1 и 2 являются 1-эквивалентными и 2-различимыми.

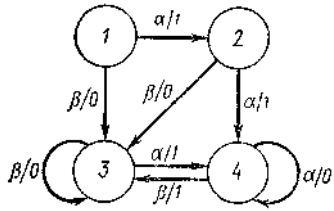


Рис. 1.

Действительно, если на автомат $\mathfrak{M}$, находящийся в состоянии 1 или в состоянии 2, подать любое слово длины 1, то получим одни и те же выходные слова. Но если подать слово αα длины 2, то получим разные выходные слова 11 и 10.

Предположим, что $Z_1 \cap Z_2 = \varnothing$ (этого всегда можно добиться переименованием элементов множеств Z_1 или Z_2). Тогда множество состояний $Z = Z_1 \cup Z_2$ можно разбить на классы k-эквивалентных состояний так, что два состояния принадлежат одному классу тогда и только тогда, когда они k-эквивалентны. Будем обозначать такое разбиение через π_k.

Состояния q_1 и q_2 называются *эквивалентными,* если они k-эквивалентны для любого конечного k. В этом случае пишут $q_1 \approx q_2$. В противном случае их называют различимыми.

Пример 2. Показать, что состояния 2 и 3 автомата $\mathfrak{M}$, данного в примере 1, эквивалентны.

По графу переходов автомата $\mathfrak{M}$ (рис 1) видно, что слово α переводит состояния 2 и 3 в одно и то же состояние 4, при этом на выходе появляется буква 1; слово β — в состояние 3, при этом на выходе появляется 0 Так как после подачи слов длины 1 состояния 2 и 3 переходят в одно и то же состояние, то любое слово длины больше 1 не сможет различить состояния 2 и 3 автомата $\mathfrak{M}$. Следовательно, состояния 2 и 3 эквивалентны.

Множество состояний Z можно разбить на классы так, что два состояния принадлежат одному классу тогда и только тогда, когда они эквивалентны. Это разбиение есть отношение эквивалентности в Z, обозначим его через π.

Свойства разбиения π_k

1. Разбиение π_k единственно.

Доказательство этого свойства очевидно.

2. Если $\pi_k \neq \pi$, то число классов в π_{k+1} больше, чем в π_k.

Доказательство. Если $\pi_k \neq \pi$, то существуют состояния q_1 и q_2 k-эквивалентные и различимые. Пусть входное слово

$a_{i_1}\ldots a_{i_l}$ $(l > k)$ .....— слово наименьшей длины, различающее состояния q_1 и q_2 автоматов $\mathfrak{M}_1$ и $\mathfrak{M}_2$. Возьмем слово $a_{i_1}\ldots a_{i_{l-k-1}}$, являющееся началом слова $a_{i_1}\ldots a_{i_l}$, и подадим на вход автомата $\mathfrak{M}_1$, находящегося в состоянии q_1, и на вход автомата $\mathfrak{M}_2$, находящегося в состоянии q_2, тогда автомат $\mathfrak{M}_1$ перейдет в состояние q'_1, а автомат $\mathfrak{M}_2$—в состояние q'_2. Состояния q'_1 и q'_2 будут k -эквивалентны, но k+1-различимы. Следовательно, состояния q'_1 и q'_2 будут принадлежать одному классу разбиения π_k, но разным классам разбиения π_{k+1}. С другой стороны, любые два состояния, которые k -различимы, должны быть k+1-различимы. Получили, что если $\pi_k \neq \pi$, то π_{k+1} есть собственное подразбиение π_k т. е. число классов π_{k+1} больше, чем π_k.

Число классов в любом разбиении не может превзойти числа состояний, поэтому получаем следующее свойство π_k.

3. Если Z содержит s состояний, то $\pi_{s-1} = \pi$.

В силу полученного выше можно найти π следующей процедурой. Строим π_1. В один класс попадают два состояния, если они 1-эквивалентны или если им в таблице выходов соответствуют одинаковые строки. Теперь укажем, как по разбиению π_k построить π_{k+1}. Если состояния q_1 и q_2 принадлежат разным классам π_k или если состояния q_1 и q_2 принадлежат одному классу π_k, но существует входная буква a_i такая, что состояния $f_1(a_i, q_1)$ и $f_2(a_i, q_2)$ принадлежат разным классам π_k, то состояния q_1 и q_2 принадлежат разным классам π_{k+1}. Полученное разбиение π_{k+1} является разбиением π, если либо $\pi_k = \pi_{k+1}$, либо $k+1 = s-1$.

Пример 3. Построить π разбиение состояний автомата $\mathfrak{M}$, данного табл. 1.

Таблица 1

$z(t)$ \ $x(t)$	$z(t+1)$		$y(t)$	
	α	β	α	β
1	1	4	1	1
2	1	3	0	0
3	7	7	0	0
4	4	1	1	1
5	4	6	0	0
6	7	3	0	0
7	7	7	1	1

Процесс построения π начнем с построения π_1. Разбиение π_1 состоит из двух классов:

$$\Sigma_{11} = \{1, 4, 7\} \text{ и } \Sigma_{12} = \{2, 3, 5, 6\}.$$

Это видно из таблицы выходов автомата $\mathfrak{M}$.

Разбиение π_{i+1} $(i=1, 2, 3, 4)$ строим по разбиению π_i и таблице перехода автомата:

$$\pi_2\ \Sigma_{21} = \{1, 4, 7\};\ \Sigma_{22} = \{2, 5, 6\};\ \Sigma_{23} = \{3\},$$
$$\pi_3\ \Sigma_{31} = \{1, 4, 7\};\ \Sigma_{32} = \{2, 6\};\ \Sigma_{33} = \{3\};\ \Sigma_{34} = \{5\},$$
$$\pi_4 = \pi_3.$$

Следовательно, $\pi = \pi_3$.

2. Минимальная форма автомата. Автоматы $\mathfrak{M}_1$ и $\mathfrak{M}_2$ называются *эквивалентными* $(\mathfrak{M}_1 \approx \mathfrak{M}_2)$, если каждому состоянию q_i автомата $\mathfrak{M}_1$ найдется по крайней мере одно состояние q_j автомата $\mathfrak{M}_2$ такое, что $q_i \approx q_j$, и наоборот. В противном случае автоматы $\mathfrak{M}_1$ и $\mathfrak{M}_2$ различимы.

Пример 4. Показать, что автоматы $\mathfrak{M}_1$, и $\mathfrak{M}_2$, данные на рис. 2 и 3, эквивалентны.

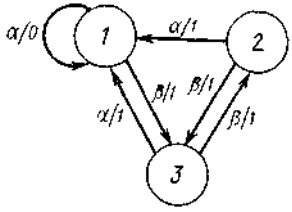


Рис. 2

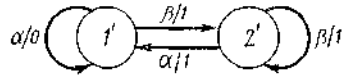


Рис. 3

Легко видеть, что состояние 1 эквивалентно состоянию 1', состояние 2 — состоянию 2' и состояние 3 — состоянию 2'. Следовательно, автоматы $\mathfrak{M}_1$ и $\mathfrak{M}_2$ эквивалентны.

Чтобы установить, являются ли автоматы $\mathfrak{M}_1$ и $\mathfrak{M}_2$ эквивалентными, надо построить разбиение π множества $Z = Z_1 \cup Z_2 (Z_1 \cap Z_2 = \varnothing)$. Если каждый класс разбиения π содержит элементы из множеств Z_1 и Z_2, тогда $\mathfrak{M}_1 \approx \mathfrak{M}_2$, в противном случае автоматы различимы. Рассмотрим автомат $\mathfrak{M} = \langle X, Y, Z, f, g \rangle$. Пусть разбиение π множества Z состоит из классов $\Sigma_1, \Sigma_2, \ldots, \Sigma_t$ и q_i — произвольный элемент из Σ_t. Определим автомат $\tilde{\mathfrak{M}} = \langle X, Y, \tilde{Z}, \tilde{f}, \tilde{g} \rangle$ следующим

образом. Множество состояний $\tilde{Z}$ состоит из l состояний $\tilde{\Sigma}_1, \ldots, \tilde{\Sigma}_l$, и функциии $\tilde{f}$ и $\tilde{g}$ таковы, что

$$\tilde{f}(a_j, \tilde{\Sigma}_i) = \tilde{\Sigma}_k, \text{ если } f(a_j, q_i) \in \Sigma_k,$$

$$\tilde{g}(a_j, \tilde{\Sigma}_i) = b_k, \text{ если } g(a_j, q_i) = b_k.$$

Из определения эквивалентных состояний следует, что построение автомата $\mathfrak{M}$ не зависит от выбора состояния q_i из класса Σ_i. Поэтому автомат $\tilde{\mathfrak{M}}$ единственный с точностью до изоморфизма для автомата $\mathfrak{M}$.

Теорема 2. *Автомат $\tilde{\mathfrak{M}}$ обладает следующими свойствами:*

1) никакие два состояния автомата $\tilde{\mathfrak{M}}$ не эквивалентны;

2) $\tilde{\mathfrak{M}} \approx \mathfrak{M}$;

3) не существует автомата $\tilde{\mathfrak{M}}$ такого, что $\mathfrak{M}' \approx \mathfrak{M}$ и содержит меньше состояний, чем автомат $\tilde{\mathfrak{M}}$.

Доказательство. Рассмотрим разбиение π для автоматов $\mathfrak{M}$ и $\tilde{\mathfrak{M}}$. Каждый класс разбиения π состоит из состояний класса Σ_i и состояния $\tilde{\Sigma}_i$. Следовательно, $\tilde{\mathfrak{M}} \approx \mathfrak{M}$ и никакие два состояния автомата $\tilde{\mathfrak{M}}$ не эквивалентны. Для доказательства свойства 3) предположим, что существует автомат $\mathfrak{M}'$ такой, что $\mathfrak{M}' \approx \mathfrak{M}$ и содержащий состояний меньше, чем $\tilde{\mathfrak{M}}$. Так как $\mathfrak{M}' \approx \mathfrak{M}$ и $\mathfrak{M} \approx \tilde{\mathfrak{M}}$, то $\mathfrak{M}' \approx \tilde{\mathfrak{M}}$. Последнее противоречит предположению, что $\mathfrak{M}'$ содержит меньше состояний, чем $\tilde{\mathfrak{M}}$.

Автомат $\tilde{\mathfrak{M}}$ называется *минимальной формой* автомата $\mathfrak{M}$.

Пример 5. Найти минимальную форму для автомата, рассмотренного в примере 3.

Из построенного в примере 3 разбиения π следует, что минимальная форма имеет граф переходов, представленный на рис. 4.

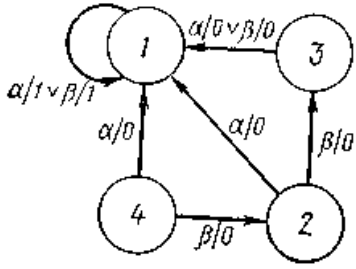


Рис. 4

7.3. Эксперименты с автоматами

1. Понятие эксперимента и классификация. Процесс применения входных слов к автоматам, наблюдения соответствующих выходных слов и получение заключений относительно внутренней структуры автомата называется *экспериментом.*

Эксперимент может быть безусловным, когда входное слово определяют заранее, или условным порядка *k,* когда входное слово состоит из *k* слов и каждое слово (исключая первое) определяют на основе предыдущих наблюдений. Эксперимент может быть простым, когда используют только один автомат, или кратным кратности *l*, когда используют *l* тождественных автоматов с одним и тем же начальным состоянием. Длина эксперимента есть число букв во входном слове. Эксперимент реализуется, если его длина конечна. Длину, порядок и кратность эксперимента можно рассматривать как грубую меру их стоимости и использовать как критерии для сравнения различных экспериментов.

В дальнейшем допустимым множеством A автомата $\mathfrak{M}$ будем называть множество состояний, про которые известно, что они могут быть начальными состояниями автомата $\mathfrak{M}$.

2. Эксперименты распознавания. Сформулируем задачу распознавания. Даны автомат $\mathfrak{M}$ своей минимальной формой и множество допустимых состояний $A=\{q_{i_1}, \ldots, q_{i_m}\}$. Требуется распознать начальное состояние автомата $\mathfrak{M}$. Решить эту задачу — значит выполнить такой эксперимент, что об истинном начальном состоянии можно сделать вывод по наблюдению входного и выходного слов. Эксперимент, который решает такую задачу, будем называть *экспериментом распознавания.* Задача распознавания, очевидно, всегда решается для допустимого множества, состоящего из одного элемента.

Теорема 1. *Задача распознавания для автомата с s состояниями и допустимым множеством из двух состояний всегда решается простым безусловным экспериментом длины, меньшей или равной* s—1.

Доказательство. Так как автомат $\mathfrak{M}$ дан минимальной формой, то любые два состояния его различимы. Входное слово, различающее состояния, принадлежащие допустимому множеству, будет решать задачу распознавания. Длина такого слова не больше чем s—1.

Входное слово, решающее задачу распознавания, будем называть *последовательностью распознавания.* Последовательность распознавания минимальной длины будем называть *минимальной*

последовательностью распознавания. Найти минимальную последовательность распознавания $\varepsilon(q_i, q_j)$ для автомата $\mathfrak{M}$ допустимого множества $A = \{q_i, q_j\}$ можно следующей процедурой. Пусть число l таково, что состояния q_i, q_j находятся в разных классах π_l и в одном классе π_{l-1}. Тогда существует входная буква a_{v_1}, такая, что состояния q_{i_1} и q_{j_1}, получаемые из состояний q_i и q_j подачей буквы a_{v_2}, являются (l—1)-различимыми и (l — 2)-эквивалентными. По состояниям q_{i_1} и q_{j_1} можно найти входную букву a_{v_2} и состояния q_{i_2} и q_{j_2}, которые будут (l — 2)-различимы и (l—3)-эквивалентны. Продолжив этот процесс, получим входное слово $a_{v_1} \ldots a_{v_l}$. Оно и есть искомая минимальная последовательность.

Пример 1. Построить минимальную последовательность распознавания для автомата $\mathfrak{M}$, заданного табл. 1, и допустимого множества $A = \{1, 2\}$.

Таблица 1

$z(t)$ \ $x(t)$	$z(t+1)$		$y(t)$	
	α	β	α	β
1	4	1	1	0
2	5	1	1	0
3	1	5	1	0
4	4	3	1	1
5	5	2	1	1

Построим π_l разбиения:

$\pi_1 : \Sigma_{11} = \{1, 2, 3\}; \Sigma_{12} = \{4, 5\},$

$\pi_2 : \Sigma_{21} = \{1, 2\}; \Sigma_{22} = \{3\}; \Sigma_{23} = \{4, 5\},$

$\pi_3 : \Sigma_{31} = \{1, 2\}; \Sigma_{32} = \{3\}; \Sigma_{33} = \{4\}; \Sigma_{34} = \{5\},$

$\pi_4 : \Sigma_{41} = \{1\}; \Sigma_{42} = \{2\}; \Sigma_{43} = \{3\}; \Sigma_{44} = \{4\}; \Sigma_{45} = \{5\}.$

При l=4 состояния 1 и 2 попадают в разные классы π_4 и находятся в одном классе π_3. При подаче буквы α они переходят в состояния 4 и 5, которые находятся в разных классах π_3, но в одном классе π_2 Эти два состояния при подаче символа β переходят в состояния 3 и 2, которые находятся в одном классе π_1 и в разных π_2. Наконец, при подаче буквы $\alpha(\beta)$ они переходят в состояния 1, 5 (5, 1), которые находятся в разных классах π_1, т. е. различимы подачей буквы β. Последовательность будет αβαβ или $\alpha\beta\beta\beta$.

Рассмотрим эксперимент распознавания в случае, когда допустимое множество A содержит m элементов (m>2). При построении простого

безусловного эксперимента распознавания для автомата $\mathfrak{M}$ с допустимым множеством $A=\{q_{i_1},\ldots,q_{i_m}\}$ надо найти входное слово, при подаче которого на вход автомата $\mathfrak{M}$ для каждого из m состояний множества A получаются на выходе разные слова. Возьмем входное слово $\tilde{\alpha}$. Пусть $A^{(v)}$— множество состояний, которое получается из состояний A при подаче v букв слова $\tilde{\alpha}$. Различных множеств $A^{(v)}(|\tilde{Z}|=s)$ может быть не больше чем $\sum_{i=1}^{m} C_s^i$. Следовательно, после подачи слова длины $\sum_{i=1}^{m} C_s^i$ множество A перейдет в множество $A_m^{(v)}$, которое появлялось раньше. Таким образом, если первые $\sum_{i=1}^{m} C_s^i$ букв слова $\tilde{\alpha}$ не составляют последовательность распознавания, то слово $\tilde{\alpha}$ не будет ею. Последовательность распознавания можно найти перебором всех входных слов длины $\sum_{i=1}^{m} C_s^i$; их не более чем $p^{\sum_{i=1}^{m} C_s^i}$, где p — число букв в алфавите X.

Процесс нахождения последовательности распознавания можно упорядочить. Для этого надо построить дерево эксперимента распознавания. Рассмотрим дерево с корнем, обладающее свойствами:

1) оно ориентированное,

2) из каждой вершины исходит p ребер, если $|X|=p$,

3) в каждую вершину, исключая корень, входит одно ребро;

4) каждому исходному ребру приписана буква входного алфавита, разным исходящим ребрам одной вершины — разные буквы входного алфавита,

5) корню дерева приписано допустимое множество A;

6) остальным вершинам приписаны множества $A^{(v)}$, которые получаются из A подачей входного слова длины v, приписанного пути, ведущему из корня в эту вершину;

7) множество $A^{(v)}$ есть объединение множеств $A_1^{(v)},\ldots,A_{l_v}^{(v)}$, где множество

$$A_j^{(v)}\ (j=1,\ldots,\ l_v)$$

получается из тех состояний множества, которые неразличимы подачей входного слова.

Полученное дерево является бесконечным; из него, обрезая ветви дерева, можно получить дерево эксперимента распознавания. Будем обрезать ветвь дерева в тех случаях, если:

1) множество встречалось ранее,

2) $A_j^{(\nu)}$ содержит два одинаковых элемента,

3) все $A_j^{(\nu)}(j=1,\ldots,l_\nu)$ состоят из одного элемента.

В результате мы получим дерево эксперимента распознавания. Последовательностью распознавания будут являться входные слова, приписанные ветвям дерева, обрезанным в силу условия 3).

Пример 2. Построить эксперимент распознавания для автомата $\mathfrak{M}$,заданного графом переходов (рис. 1), и допустимого множества $A = \{1, 2, 3, 4\}$.

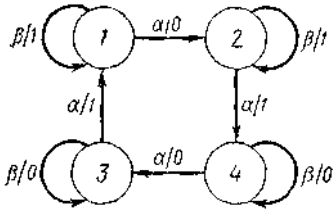


Рис. 1

По дереву эксперимента распознавания, представленного на рис. 2, видно, что в качестве последовательности распознавания можно взять либо слово $\alpha\beta$, либо слово $\beta\alpha$.

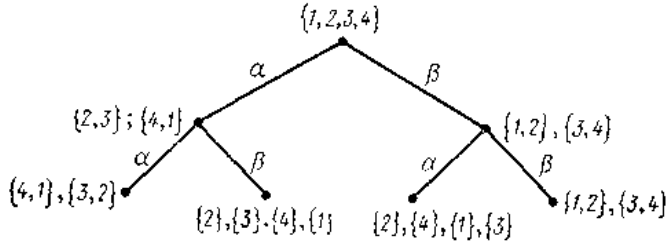


Рис. 2

Задача распознавания не всегда решается простым безусловным экспериментом. Например, для автомата $\mathfrak{M}$, заданного графом переходов (рис. 3), и множества допустимых состояний

$A = \{1, 2, 3, 4\}$ задачу распознавания нельзя решить простым безусловным экспериментом.

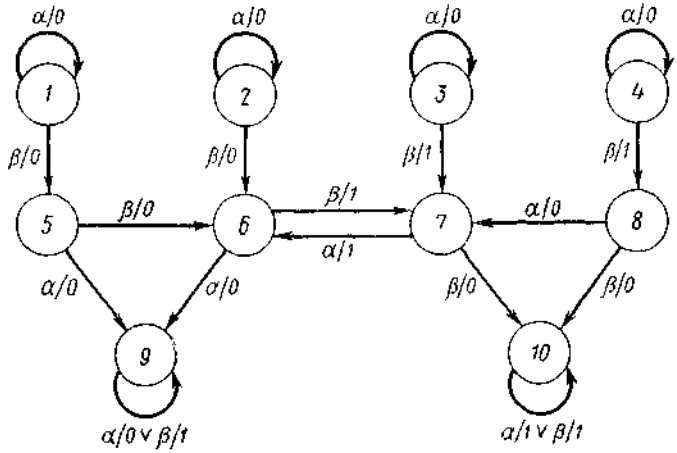


Рис. 3

Ясно, что последовательность распознавания должна начинаться с буквы β, второй буквой не может быть ни буква α, в этом случае состояния 1 и 2 неразличимы, ни буква β, в этом случае состояния 3 и 4 неразличимы. Однако задача распознавания для этого множества допустимых состояний может быть решена простым условным экспериментом. Надо сначала подать слово β и затем в зависимости от выходного слова подать либо слово α, либо слово β. Задача распознавания для множества допустимых состояний {5, 6, 7, 8} не решается простым экспериментом.

Теорема 2. *Задача распознавания всегда решается кратным экспериментом.*

Доказательство. В множестве A допустимых начальных состояний любые два состояния различимы.

На первую копию автомата подадим входное слово $\tilde{\alpha}$, различающее некоторые два состояния множества A. Множество A можно представить как объединение множеств $A_1, \ldots, A_t$ $(A_i \cap A_j = \varnothing)$, где A_i состоит из состояний, неразличимых словом $\tilde{\alpha}$. В каждом множестве A_i любые два состояния различимы. На следующую копию автомата подадим входное слово $\tilde{\beta}$, различающее некоторые два состояния множества A_i. Этот процесс надо продолжать до тех пор, пока не удастся различить все состояния множества A Очевидно, что эта процедура конечна в силу конечности множества A.

3. Установочные эксперименты. Сформулируем установочную задачу. Даны конечный автомат $\mathfrak{M}$ своей минимальной формой и допустимое множество $A=\{q_{i_1}, \ldots, q_{i_m}\}$. Требуется перевести автомат $\mathfrak{M}$ в известное состояние (провести такой эксперимент, что о состоянии, достигнутом в конце эксперимента, можно судить, наблюдая входное и выходное слово). Эксперимент, который решает эту задачу, называется *установочным экспериментом* для автомата $\mathfrak{M}$ и допустимого множества *A*.

Теорема 3. *Установочная задача всегда решается простым безусловным экспериментом.*

Доказательство. Построим серию множеств $A_1, \ldots, A_l$, где каждое множество

$$A_j = A_{j_1} \cup \ldots \cup A_{j_{r_j}},$$

а A_{j_k} — подмножество состояний автоматам, может быть с повторяющимися элементами. Множество A_1 совпадает с допустимым множеством $A=\{q_{i_1}, \ldots, q_{i_m}\}$. Если каждое множество $A_{j_k}(k=1, \ldots, r_j)$ состоит из одного, может быть повторяющегося, элемента множества Z, то множество A_i совпадает с A_l, т. е. с последним множеством в серии. Если множество A_i не совпадает с множеством A_l то из него можно получить множество A_{i+1} следующим образом. Пусть множество A_{j_k} — произвольное из A_j, которое содержит по крайней мере два различных состояния q и q'. Они различимы входным словом $\varepsilon_j(q, q')=\varepsilon_j$. Пусть $A'_{j_1}, \ldots, A'_{j_{r_j}}$ означают множества состояний, в которые входное слово ε_j переводит множества

$$A_{j_1}, \ldots, A_{j_{r_j}}.$$

По слову ε_j определим разбиение множества A_{j_k}. Два состояния множества A_{j_k} принадлежат одному классу, если они не различимы словом ε_j. По разбиению множества A_{j_k} определим разбиение множества A'_{j_k}. Если два состояния множества A_{j_k} принадлежат разным классам, то состояния, в которые они переходят при подаче слова ε_j, тоже принадлежат разным классам A'_{j_k}. Множество A_{j+1} получается как множество всех классов, получающихся из разбиений всех $A^*_{j_k}$. Множество A_{j_k} разбивается по крайней мере на два класса, поэтому число множеств в A_{j+1} всегда превосходит число их в

A_j. После самое большое m применений описанной процедуры получается множество A_l, где каждое A_{l_k} есть множество, содержащее только один может быть повторяющийся элемент. Входное слово $\varepsilon_1\varepsilon_2 \ldots \varepsilon_l$ переводит множество A в множество A_l и по построению обладает тем свойством, что каждое состояние из A_l достигается с помощью его из A с разными выходными словами.

Пример 3. Построить установочный эксперимент для автомата $\mathfrak{M}$ (рис. 4) и допустимого множества $A = \{1, 2, 3, 4\}$.

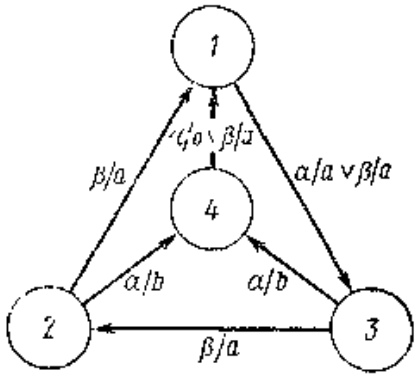


Рис. 4

Состояния 1 и 2 различимы входным словом α. Подадим это слово на автомат $\mathfrak{M}$. Получим, что если автомат $\mathfrak{M}$ был 1) в состоянии 1, он перейдет в состояние 3 и на выходе его появится слово *a,* 2) в состоянии 2 или 3 — перейдет в состояние 4 и на его выходе появится слово *b,* 3) в состоянии 4, то он перейдет в состояние 1 и на его выходе будет слово *b.* Теперь осталось различить состояния 1 и 4, они различаются входным словом α. Усгановочной последовательностью будет входное слово αα.

7.4. Абстрактный синтез конечных автоматов

1. Вводные замечания. Существующие языки, применяемые для описания конечных автоматов, можно разбить на две группы: языки, в которых используется переменная, определяющая внутреннее состояние, и языки, оперирующие только с понятиями вход — выход. К первой группе относятся описания автомата с помощью таблиц и графа переходов. Ко второй группе относятся разные варианты

языка регулярных формул, предикатный язык и т. п. Последние являются более удобными для разговора между заказчиком и исполнителем. Поэтому желательно уметь переходить от описания автоматов с помощью языков второй группы к описанию автоматов на языках первой. Такой переход называют *абстрактным синтезом* конечного автомата.

2. Регулярные выражения. Пусть дан алфавит $X = \{a_1, \ldots, a_p\}$, символы Λ, $\varnothing$, $+$, $\cdot$, $*$ и скобки. *Регулярным выражением* являются:

1) всякая буква алфавита X и символы Λ, $\varnothing$,

2) выражения $(E_1 \cdot E_2)$, $(E_1 + E_2)$, (E_1^*), если E_1 и E_2— регулярные выражения.

Других регулярных выражений не существует.

Регулярные выражения применяют для описания множеств слов в алфавите X, при этом символы Λ, $\varnothing$, $+$, $\cdot$ и $*$ интерпретируют следующим образом:

1) символ Λ — пустое слово (слово нулевой длины),

2) символ $\varnothing$ — пустое множество слов,

3) множество $(E_1 + E_2)$— теоретико-множественное объединение множеств слов E_1 и E_2,

4) множество $(E_1 \cdot E_2)$ есть множество всех слов вида $\tilde{\alpha}\tilde{\beta}$, где $\tilde{\alpha}$— слово из E_1 и $\tilde{\beta}$ — слово из E_2. Символ $\cdot$ называют *умножением,*

5) множество (E_1^*) есть множество

$$\Lambda + E_1 + (E_1 \cdot E_1) + ((E_1 \cdot E_1) \cdot E_1) + \ldots.$$

Символ * называют *итерацией.*

Например, выражение $(a_1 \cdot a_2 + a_3) \cdot a_3$ следует понимать как совокупность слов $a_1a_2a_3$ и a_3a_3, а выражение $(a_1 + a_2)^*$ — как совокупность всех слов, состоящих из букв a_1 и a_2, т. е. Λ, a_1, a_2, a_1a_1, a_1a_2, a_2a_1, a_2a_2, a_1a_1 $a_1, \ldots$.

Некоторые свойства операций

1) $E_1+E_2=E_2+E_1$,
2) $E_1+(E_2+E_3)=(E_1+E_2)+E_3$,
3) $E_1\cdot(E_2\cdot E_3)=(E_1\cdot E_2)\cdot E_3$,
4) $E_1\cdot(E_2+E_3)=(E_1\cdot E_2)+(E_1\cdot E_3)$,
5) $(E_1+E_2)\cdot E_3=(E_1\cdot E_3)+(E_2\cdot E_3)$,
6) $E_1+\varnothing=\varnothing+E_1=E_1$,
7) $E_1\cdot\varnothing=\varnothing\cdot E_1=\varnothing$,
8) $E_1\cdot\Lambda=\Lambda\cdot E_1=E_1$;
9) $E_1+E_1=E_1$,
10) $E_1^*=\Lambda+E_1\cdot(E_1^*)$,
11) $E_1\cdot(E_1^*)=(E_1^*)\cdot E_1$,
12) $(E_1^*)\cdot(E_1^*)=E_1^*$,
13) $(E_1+E_2)^*=\left((E_1^*)+(E_2^*)\right)^*$,
14) $\Lambda^*=\Lambda$,
15) $\varnothing^*=\Lambda$.

Пример 1. Показать, что множество всех последовательностей из 0 и 1, которые содержат две последовательные единицы, может быть представлено регулярным выражением.

Регулярное выражение $(0+1)^*\,11\,(0+1)^*$ представляет все последовательности, содержащие две последовательные единицы.

3. События. Регулярные события. Множество входных слов называется *событием.* Событие представимо в инициальном автомате $\mathfrak{M}$ выходной буквой b_i, если выходные слова, которые получаются при подаче на автомат $\mathfrak{M}$ события, кончаются символом b_i. Событие представимо в инициальном автомате $\mathfrak{M}$ множеством выходных букв $Y'\subseteq Y$, если событие есть объединение событий, представимых всеми элементами множества Y'.

Рассмотрим конечный автомат $\mathfrak{M}$ в начальном состоянии q_1. Поведение этого автомата полностью можно задать совокупностью событий $E_{b_1},\ldots,E_{b_t}$, где E_{b_i} — множество всех входных слов, которые переходят в выходные слова, кончающиеся на букву b_i ($b_1,\ldots,b_t$ — выходной алфавит). Так, если дано входное слово $a_{i_1}\ldots a_{i_l}$, то можно определить, в какое выходное слово оно перейдет, если известны события $E_{b_1},\ldots,E_{b_t}$ для автомата $\mathfrak{M}$. С. К. Клини показал, что всякое событие, представимое в автомате, может быть представлено в виде регулярного выражения, и наоборот,

— всякое событие, представимое в виде регулярного выражения, представимо в автомате. Докажем эту теорему для некоторого частного случая, а именно будем полагать, что $XY=\{0,1\}$, и автомат задан уравнениями

$$y(t)=g\left(z(t+1)\right),$$
$$z(t+1)=f\left(x(t),\ z(t)\right). \tag{1}$$

Теорема *1* **(Клини)** (начало теоремы). *Всякое событие, представимое в автомате, регулярно, т. е. может быть записано в виде регулярного выражения.*

Доказательство. Пусть дан автомат $\mathfrak{M}$ с $Z=\{q_1, \ldots, q_s\}$, где $q_1=z(0)$ — начальное состояние автомата. В случае когда автомат $\mathfrak{M}$ задан системой (1) и $X=Y=\{0, 1\}$, считают, что событие представимо в автомате, если слова из этого события переводят начальное состояние q_1 в состояние, которое дает на выходе букву 1.

Докажем индукцией по p, что множество e_{ij}^{p} всех входных слов, переводящих состояние q_i в состояние q_j без прохождения через состояния q_k, с номерами k, большими номера p, регулярно. Множество e_{ij}^{0} всех входных слов, переводящих состояние q_i в q_j непосредственно без прохождения других состояний, есть $\{0\}$, $\{1\}$, $\{0+1\}$, Λ или $\varnothing$.

Пусть e_{ij}^{p-1} — регулярное событие, покажем, что e_{ij}^{p} — регулярное событие. Событие e_{ij}^{p} можно представить в виде

$$e_{ij}^{p-1}+e_{ip}^{p+1}\cdot(e_{pp}^{p-1})^{*}\cdot e_{pj}^{p-1}.$$

Выписанное выражение является регулярным, так как построено из регулярных событий e_{ij}^{p-1}, e_{ip}^{p-1}, e_{pp}^{p-1} и e_{pj}^{p-1} с помощью операций $+$, $\cdot$, *.

Теперь рассмотрим автомат $\mathfrak{M}$ с начальным состоянием q_1. Пусть $q_{i_1}, \ldots, q_{i_k}$ — состояния, которые ассоциируются с выходом 1, тогда событие, реализуемое автоматом, имеет вид

$$e_{1i_1}^{s}+\ldots+e_{1i_k}^{s}.$$

Таким образом, показано, что событие, реализуемое в автомате, есть регулярное.

Осталось показать обратное, для этого введем вспомогательные понятия и докажем некоторые их свойства.

4. Производная события. Свойства производной. Если дано событие E, то под производной этого события по входному слову $\tilde{\alpha}$ понимается множество

$$\partial_{\tilde{\alpha}} E = \{\tilde{\beta} / \tilde{\alpha}\tilde{\beta} \in E\}.$$

Дадим другое определение производной для случая, когда событие представимо в виде регулярного выражения. Это определение будет иметь индуктивный характер в силу индуктивного характера определения регулярного выражения.

Даны алфавит $X = \{a_1, \ldots, a_p\}$, событие E и слово $\tilde{\alpha}$. Производная от события E по слову $\tilde{\alpha}$ определяется следующим образом:

1) если $E = a_i$ и $\tilde{\alpha} = a_i$, то $\partial_{a_i} E = \Lambda$,

2) если $E = a_i$ и $\tilde{\alpha} = a_j (i \neq j)$, то $\partial_{a_j} E = \varnothing$,

3) если $E = \varnothing$ и $\tilde{\alpha} = a_i$, то $\partial_{a_i} E = \varnothing$,

4) если $E = \Lambda$ и $\tilde{\alpha} = a_i$, то $\partial_{a_i} E = \varnothing$,

5) если $E = E_1 + E_2$ и $\tilde{\alpha} = a_i$, то $\partial_{a_i} E = \partial_{a_i} E_1 + \partial_{a_i} E_2$,

6) если $E = E_1 \cdot E_2$ и $\tilde{\alpha} = a_i$, то

$$\partial_{a_i} E = \begin{cases} (\partial_a E_1) \cdot E_2, & \text{если } \Lambda E \notin E_1, \\ (\partial_{a_i} E_1) \cdot E_2 + \partial_{a_i} E_2, & \text{если } \Lambda \in E_1, \end{cases}$$

7) если $E = E_1^*$ и $\tilde{\alpha} = a_i$, то

$$\partial_{a_i} E = (\partial_{a_i} E_1) \cdot E_1^*,$$

8) если E — произвольное событие и $\tilde{\alpha} = \Lambda$, $\partial_\Lambda E = E$,

9) если E — произвольное событие и $\tilde{\alpha} = a_{i_1} \ldots a_{i_{k-1}} a_{i_k}$,

то

$$\partial_{\tilde{\alpha}} E = \partial_{a_{i_k}} (\partial_{a_{i_1} \ldots a_{ik-1}} E).$$

Пример 2. Найти производную события $(0 + 1)^* 0 + 0$ по слову 01.

Из определения производной имеем:

$$\partial_{0\tilde{\alpha}}((0+1)^*\cdot 0+0)=\partial_1\partial_0((0+1)^*0+0),$$
$$\partial_0((0+1)^*0+0)=\partial_0((0+1)^*\cdot 0)+\partial_0 0=(\partial_0(0+1)^*)0+$$
$$+\partial_0 0+\Lambda=(\partial_0(0+1))\cdot(0+1)^*\cdot 0+\Lambda=$$
$$=(\partial_0 0+\partial_0 1)(0+1)^*0+\Lambda=(\Lambda+\varnothing)(0+1)^*\cdot 0+\Lambda=$$
$$=(0+1)^*\cdot 0+\Lambda,$$
$$\partial_1((0+1)^*0+\Lambda)=\partial_1((0+1)^*\cdot 0)+\partial_1\Lambda=$$
$$=(\partial_1(0+1))(0+1)^*\cdot 0+\varnothing=(\partial_1 0+\partial_1 1)(0+1)^*\cdot 0=$$
$$=(0+1)^*0.$$

Теорема 2. *Любое регулярное выражение E имеет конечное число t различных производных, каждая из которых равна некоторой производной регулярного выражения E по слову длины* $\leqslant t-1$.

Доказательство. Докажем конечность числа производных регулярного выражения E индукцией по числу операций $+$, $\cdot$, * в E. Пусть число операций равно нулю, тогда регулярное выражение E есть Λ, $\varnothing$ или a_i, где $a_i\in X$, а производные по произвольному слову имеют вид

$$\partial_{\tilde{\alpha}}\Lambda=\begin{cases}\Lambda, & \text{если } \tilde{\alpha}=\Lambda,\\ \varnothing, & \text{если } \tilde{\alpha}\neq\Lambda,\end{cases}\qquad \partial_{\tilde{\alpha}}a_i=\begin{cases}\tilde{a}_i, & \text{если } \alpha=\Lambda,\\ \Lambda, & \text{если } \tilde{\alpha}=a_i,\\ \varnothing, & \text{в остальных}\end{cases}$$

случаях, $\partial_{\tilde{\alpha}}\varnothing=\varnothing$.

В этом случае число t производных регулярного выражения конечно и меньше четырех.

Предположим, что всякое регулярное выражение E, содержащее n или меньше операций, имеет конечное число производных t_E. Рассмотрим регулярное выражение E, содержащее $n+1$ операцию. Выделим последнюю операцию, тогда регулярное выражение E может быть представлено как E_1+E_2, $E_1\cdot E_2$ или E_1^*, где E_1 и E_2 — регулярные выражения, содержащие n или меньше операций. Следовательно, в силу индукционного предположения они имеют конечное число производных t_{E_1} и t_{E_2}. В первом случае, когда $E=E_1+E_2$, производная от регулярного выражения E по слову $\tilde{\alpha}$ равна $\partial_{\tilde{\alpha}}E_1+\partial_{\tilde{\alpha}}E_2$. Поэтому число производных t_E конечно и $\leqslant t_{E_1}\cdot t_{E_2}$. Во втором случае, когда $E=E_1\cdot E_2$, производная от E по $\tilde{\alpha}=a_1a_2\ldots a_r$ равна

$$\partial_{\tilde{\alpha}}E=(\partial_{a_1\ldots a_r}E_1)E_2+\partial_{a_r}E_2+$$
$$+\partial_{a_{r-1}a_r}E_2+\ldots+\partial_{a_1\ldots a_r}E_2. \qquad (2)$$

Некоторые производные $\partial_{a_r}E_2$, $\partial_{a_{r-1}a_r}E_2, \ldots,$ $\partial_{a_1 \ldots a_r}E_2$ могут отсутствовать в (2) в зависимости от того, содержат ли регулярные выражения $E_1, \partial_{a_1}E_1, \ldots \ldots, \partial_{a_1 \ldots a_{r-1}}E_1$ слово Λ. Число производных регулярного выражения E конечно и меньше, либо равно $t_{E_1} \cdot 2^{t_{E_2}}$. В третьем случае, когда $E = E_1^*$, производная от E по $\tilde{\alpha} = a_1 a_2 \ldots a_r$ равна

$$\partial_{\tilde{\alpha}}E = (\partial_{a_1 \ldots a_r}E_1)\,E_1^* + (\partial_{a_2 \ldots a_r}E_1)\,E_1^* + \ldots + (\partial_{a_r}E_1)\,E_1^*. \tag{3}$$

Некоторые производные $\partial_{a_1 \ldots a_r}E_1$, $\partial_{a_2 \ldots a_r}E_1, \ldots,$ $\partial_{a_r}E_1$ в (3) могут отсутствовать. Число производных регулярного выражения E конечно и меньше или равно $2^{t_{E_1}}$. Итак, число t_E производных регулярного выражения E конечно.

Осталось показать, что каждая производная совпадает с производной по слову длины, меньшей или равной t—1. Для этого покажем, что если для всех слов длины l нет новых производных регулярного выражения E, то не появится новых производных ни для какого слова длины, большей l. Рассмотрим произвольное слово $\tilde{\gamma}$ длины l+1, оно представимо в виде αa_i, где $\tilde{\alpha}$—слово длины l, а a_i — буква алфавита X. Тогда $\partial_{\tilde{\gamma}}E = \partial_{\tilde{\alpha}a_i} \times E = \partial_{a_i}\partial_{\tilde{\alpha}}E$. Но $\partial_{\tilde{\alpha}}E$ совпадает с производной регулярного выражения E по некоторому слову $\tilde{\beta}$ длины, меньшей чем l. Поэтому

$$\partial_{a_i}(\partial_{\tilde{\alpha}}E) = \partial_{a_i}(\partial_{\tilde{\beta}}E) = \partial_{\tilde{\beta}a_i}E,$$

где слово $\tilde{\beta}a_i$ длины, меньшей или равной l. Предположим, что наше утверждение верно для всякого слова $\tilde{\gamma}$ длины $l+k$. Покажем, что оно верно для слов длины $l+k+1$. Последнее показывается аналогично базису индукции. Таким образом, процесс получения производных оборвется, если для слов данной длины не будет получено новых производных. Последнее вместе с утверждением о конечном числе производных регулярного выражения доказывает теорему.

Теорема Клини (продолжение). *Всякое регулярное событие представимо в конечном автомате.*

Доказательство. Пусть дано регулярное событие E. Построим конечный автомат $\mathfrak{M}_E$, который будет представлять событие E. Если событие E имеет s различных производных, то у автомата $\mathfrak{M}_E$ будет s состояний. Состояние, соответствующее производной $\partial_{\tilde{\alpha}}E$, обозначим через $q_{\partial_{\tilde{\alpha}}E}$. Если на вход автомата $\mathfrak{M}_E$ в состоянии $q_{\partial_{\tilde{\alpha}}E}$

подают 0, то он перейдет в состояние $q_{\partial_0 \partial_{\tilde{\alpha}} E}$, а если 1, то — в состояние $q_{\partial_1 \partial_{\tilde{\alpha}} E}$. Если производная содержит слово Λ, то состояние $q_{\partial_{\tilde{\alpha}} E}$ определяет выход 1, все остальные состояния — выход 0. Начальным состоянием автомата $\mathfrak{M}_E$ является состояние $q_{\partial_\Lambda E}$. По построению входные слова, которые переводят автомат $\mathfrak{M}_E$ из состояния $q_{\partial_\Lambda E}$ в состояние $q_{\partial_{\tilde{\alpha}} E}$, начинаются со слова $\tilde{\alpha}$. С другой стороны, слово $\tilde{\alpha}$ принадлежит событию E тогда и только тогда, когда слово Λ принадлежит $\partial_{\tilde{\alpha}} E$. Следовательно, множество всех входных слов, переводящих автомат $\mathfrak{M}_E$ из состояния $q_{\partial_\Lambda E}$ в состояния, дающие выход 1, есть событие E.

Пример 3. Построить автомат, реализующий событие 1+0*. Найдем производные события 1+0*:

$$\partial_\Lambda (1+0^*) = 1+0^*;\ \partial_0 (1+0^*) = 0^*;\ \partial_1 (1+0^*) = \Lambda,$$

$$\partial_{00} (1+0^*) = 0^*,\ \partial_{01} (1+0^*) = \varnothing,\ \partial_{10} (1+0^*) = \varnothing,$$

$$\partial_{11} (1+0^*) = \varnothing,$$

$$\partial_{000} (1+0^*) = 0^*,\ \partial_{001} (1+0^*) = \varnothing,\ \partial_{010} (1+0^*) = \varnothing,$$

$$\partial_{011} (1+0^*) = \varnothing,$$

$$\partial_{100} (1+0^*) = \varnothing,\ \partial_{101} (1+0^*) = \varnothing,\ \partial_{110} (1+0^*) = \varnothing,$$

$$\partial_{111} (1+0^*) = \varnothing.$$

Автомат, реализующий событие 1 + 0*, представлен на рис. 5.

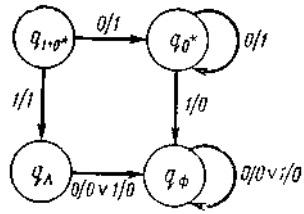


Рис. 5

7.5. структурный синтез

1. Операции над автоматами. До сих пор автомат рассматривался с функциональной точки зрения, т. е. как нечто неделимое. Теперь будем

предполагать, что он состоит из элементов — «элементарных» автоматов, соединенных между собой,— рассматривать структурную модель автомата. Будем интересоваться связью между функциональной и структурной моделями автоматов, а именно нахождением структурной модели по функциональной модели, и наоборот. Первая задача называется задачей *структурного синтеза*, вторая — *структурного анализа.*

Сформулируем задачу синтеза. Даны инициальный конечный автомат (его таблица, система уравнений, граф), базис синтеза — список «элементарных» автоматов, известны правила соединения элементарных автоматов. Найти структурную схему автомата, заданного функциональной моделью. Последнее означает: соединить «элементарные» автоматы по указанным правилам так, чтобы функциональная модель полученного автомата совпадала с функциональной моделью заданного автомата. Под «элементарными» автоматами понимают любые автоматы, которые в задаче синтеза рассматриваются как неделимые блоки, из которых строится структурная модель.

Остановимся на способах соединения элементарных автоматов. В этом параграфе будем полагать, что каждый автомат может иметь произвольное конечное число входов и выходов. Соединение автоматов есть отождествление входов и выходов (или входных и выходных полюсов). Произвольное отождествление полюсов может быть не корректно, так как, в силу определения конечного автомата, должны выполняться условия наличия в любой момент времени сигнала на всех входах и выходах и однозначности сигнала на каждом входе и выходе. Некорректность возникает, например, при присоединении к одному входу нескольких выходов или в случае петли в структурной схеме. Можно избежать некорректности, вводя ограничения на способы соединения элементарных автоматов.

Объединение автоматов

Пусть даны два автомата $\mathfrak{M}_1$ и $\mathfrak{M}_2$, имеющие n_1 и n_2 входов и m_1 и m_2 выходов соответственно. Автомат $\mathfrak{M}_1$ задан системой

$$\begin{aligned}
&y_{11}(t)=g_{11}\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&y_{1m_1}(t)=g_{1m_1}\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&z_1(t+1)=f_1\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&z_1(0)=q_{11},
\end{aligned} \qquad (1)$$

а автомат $\mathfrak{M}_2$ — системой

$$y_{21}(t) = g_{21}\left(x_{21}(t), \ldots, x_{2n_2}(t), z_2(t)\right),$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$y_{2m_2}(t) = g_{2m_2}\left(x_{21}(t), \ldots, x_{2n_2}(t), z_2(t)\right),$$

$$z_2(t+1) = f_2\left(x_{21}(t), \ldots, x_{2n_2}(t), z_2(t)\right), \qquad (2)$$

$$z_2(0) = q_{21}.$$

Объединением автоматов $\mathfrak{M}_1$ и $\mathfrak{M}_2$ называется автомат $\mathfrak{M}$ с n_1+n_2 входами и m_1+m_2 выходами и заданный системой уравнений, являющейся объединением уравнений (1), (2), объединение автоматов $\mathfrak{M}_1$ и $\mathfrak{M}_2$ схематически изображено на рис. 1.

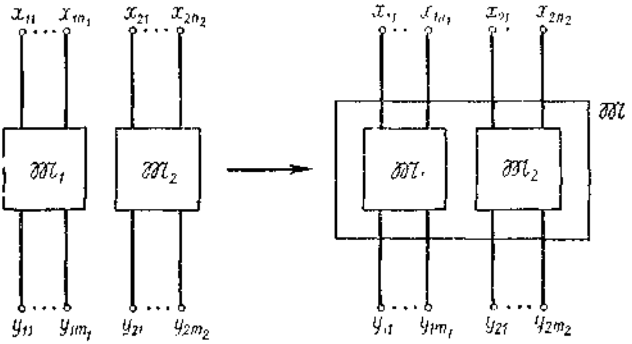


Рис. 1

Композиция автоматов

Пусть даны два автомата $\mathfrak{M}_1$ и $\mathfrak{M}_2$, заданные системами (1) и (2) соответственно. *Композицией автоматов* $\mathfrak{M}_1$ и $\mathfrak{M}_2$ 'называется автомат $\mathfrak{M}$, который получается отождествлением выходов $y_{1l_1}, \ldots, y_{1l_k}$ автомата $\mathfrak{M}_1$ и входов $x_{2j_1}, \ldots, x_{2j_k}$ автомат а $\mathfrak{M}_2$. Этот автомат задан системой

$$
\begin{aligned}
&y_{11}(t)=g_{11}\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&y_{1(i_1-1)}(t)=g_{1(i_1-1)}\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&y_{1(i_1+1)}(t)=g_{1(i_1+1)}\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&y_{1(i_k-1)}(t)=g_{1(i_k-1)}\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&y_{1(i_k+1)}(t)=g_{1(i_k+1)}\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&y_{1m_1}(t)=g_{1m_1}\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&z_1(t+1)=f_1\left(x_{11}(t),\ldots,\ x_{1n_1}(t),\ z_1(t)\right),\\
&z_1(0)=q_{11},\\
&y_{21}(t)=g_{21}\left(\ldots,\ x_{2(j_1-1)}(t),\ y_{1i_1}(t),\ldots,\ x_{2(j_k-1)}(t),\right.\\
&\qquad\qquad\qquad\qquad\qquad\qquad\left.y_{1i_k}(t),\ldots\right),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\\
&y_{2m_2}(t)=g_{2m_2}\left(\ldots,\ x_{2(j_1-1)}(t),\ y_{1i_1}(t),\ldots,\ x_{2(j_k-1)}(t),\right.\\
&\qquad\qquad\qquad\qquad\qquad\qquad\left.y_{1i_k}(t),\ldots\right),\\
&z_2(t+1)=f_2\left(\ldots,\ x_{2(j_1-1)}(t),\ y_{1i_1}(t),\ldots,\ x_{2(j_k-1)}(t),\right.\\
&\qquad\qquad\qquad\qquad\qquad\qquad\left.y_{1i_k}(t),\ldots\right),\\
&z_2(0)=q_{21}.
\end{aligned}
\tag{3}
$$

Графическое представление этой операции изображено на рис. 2. В силу данного определения один входной полюс можно отождествить с одним выходным.

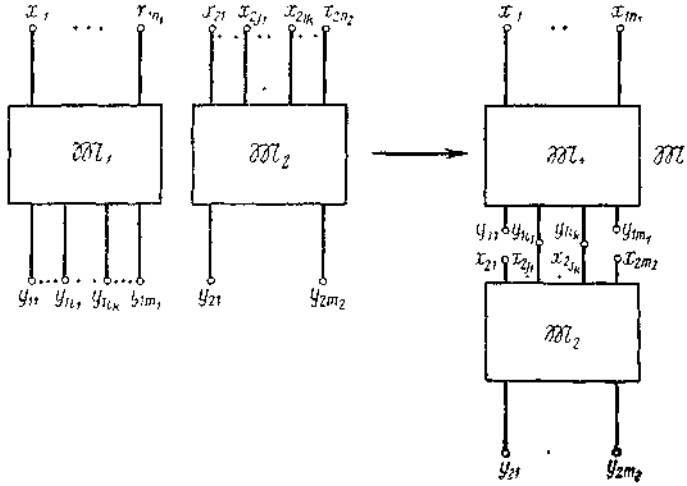


Рис. 2

Отождествление входов

Пусть автомат $\mathfrak{M}_1$ дан системой уравнений (1). Автомат $\mathfrak{M}$, полученный отождествлением x_{1i} и x_{1j} входов, задается системой уравнений

$$\begin{cases} y_{11}(t) = g_{11}\big(x_{11}(t), \ldots, x_{1i}(t), \ldots, x_{1(j-1)}(t), \\ \qquad\qquad x_{1i}(t), \ldots, x_{1n_1}(t), z_1(t)\big), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_{1m_1}(t) = g_{1m_1}\big(x_{11}(t), \ldots, x_{1i}(t), \ldots, x_{1(j-1)}(t), \\ \qquad\qquad x_{1i}(t), \ldots, x_{1n_1}(t), z_1(t)\big), \\ z_1(t+1) = f_1\big(x_{11}(t), \ldots, x_{1i}(t), \ldots, x_{1(j-1)}(t), \\ \qquad\qquad x_{1i}(t), \ldots, x_{1n_1}(t), z_1(t)\big), \end{cases} \tag{4}$$

$$z_1(0) = q_{11}.$$

Графическое, изображение этой операции представлено на рис. 3

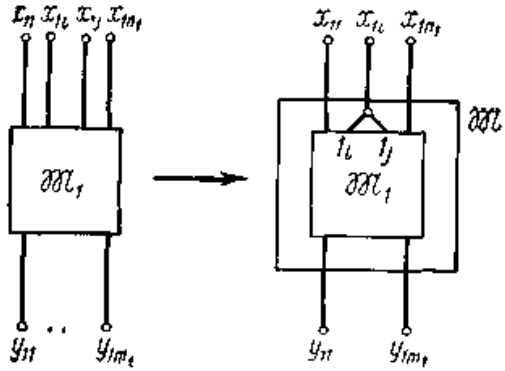


Рис. 3

Операция обратной связи

Пусть дан автомат $\mathfrak{M}_1$, заданный системой уравнений (1), и функция g_{1j} не зависит существенно от входной переменной x_{1i}. Тогда автомат $\mathfrak{M}$, полученный отождествлением y_{ij} выхода с x_{1i} входом автомата $\mathfrak{M}_1$, есть автомат, полученный операцией обратной связи. Он задается системой

$$
\begin{aligned}
y_{11}(t) &= g_{11}\big(x_{11}(t), \ldots, x_{1(i-1)}(t), y_{1j}(t), \\
&\qquad x_{1(i+1)}(t), \ldots, x_{1n_1}(t), z_1(t)\big), \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
y_{1j}(t) &= g_{1j}\big(x_{11}(t), \ldots, x_{1(i-1)}(t), \\
&\qquad x_{1(i+1)}(t), \ldots, x_{1n_1}(t), z_1(t)\big), \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
y_{1m_1}(t) &= g_{1m_1}\big(x_{11}(t), \ldots, x_{1(i-1)}(t), y_{1j}(t), \\
&\qquad x_{1(i+1)}(t), \ldots, x_{1n_1}(t), z_1(t)\big), \\
z_1(t+1) &= f_1\big(x_{11}(t), \ldots, x_{1(i-1)}(t), y_{1j}(t), \\
&\qquad x_{1(i+1)}(t), \ldots, x_{1n_1}(t), z_1(t)\big), \\
z_1(0) &= q_{11}.
\end{aligned}
\tag{5}
$$

Графическое изображение этой операции представлено на рис. 4.

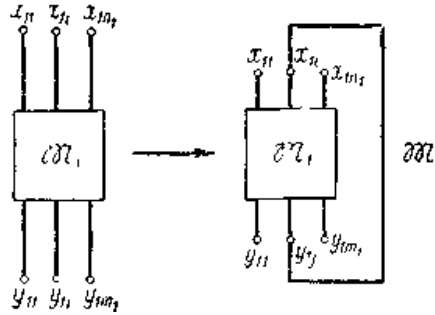


Рис. 4

Выделение выхода

Пусть дан автомат $\mathfrak{M}_1$, заданный системой (1). Тогда автомат $\mathfrak{M}$ с n_1 входами и $k\,(k \leqslant m_1)$ выходами, полученный из $\mathfrak{M}_1$ выделением выходов $y_{1i_1}, \ldots, y_{1i_k}$ есть автомат, полученный операцией выделения выходов; он дан системой

$$\begin{aligned} y_{1i_1} &= g_{1i_1}\left(x_{11}(t), \ldots, x_{1n_1}(t), z_1(t)\right), \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y_{1i_k} &= g_{1i_k}\left(x_{11}(t), \ldots, x_{1n_1}(t), z_1(t)\right), \\ z_1(t+1) &= f_1\left(x_{11}(t), \ldots, x_{1n_1}(t), z_1(t)\right), \\ z_1(0) &= q_{11}. \end{aligned} \tag{6}$$

Графическое изображение этой операции представлено на рис. 5.

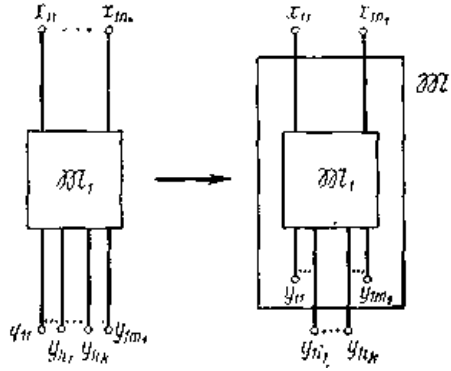


Рис. 5

Прежде чем перейти к решению задачи синтеза, остановимся на вопросе выбора базиса. Не всякий набор элементарных автоматов позволяет решить задачу структурного синтеза. Поэтому сначала приходится решать задачу о возможности построения автомата с заданным функциональным поведением из данного набора элементарных автоматов, т. е. решить, удовлетворяет ли базис условию полноты.

Базис называется *полным* относительно некоторого класса автоматов, если в нем может быть синтезирован любой автомат из данного класса.

2. Структурный синтез автоматов без памяти. Рассмотрим задачу структурного синтеза автоматов без памяти, т. е. конечных автоматов, у кото рых $|Z| = 1$ и система имеет вид

$$\begin{aligned} y_1(t) &= g_1(x_1(t), \ldots, x_n(t)), \\ &\cdots\cdots\cdots\cdots\cdots \\ y_m(t) &= g_m(x_1(t), \ldots, x_n(t)). \end{aligned} \qquad (7)$$

Введем еще одно ограничение — будем решать задачу синтеза для важного как с теоретической, так и практической точки зрения случая, когда входной и выходной алфавиты автомата совпадают и состоят из 0 и 1. Тогда функции выхода автомата есть булевы, или переключательные функции. Формулируя задачу синтеза в случае автоматов без памяти, следует отказаться от операции обратной связи, которая ведет к некорректности. Вопрос о выборе полного базиса в этом случае сводится к исследованию полноты системы булевых функций.

Задача синтеза для автомата без памяти может быть сформулирована так: дана система уравнений (7) и полная система булевых функций; найти представление функций $g_1, \ldots, g_m$ через функции полной системы.

Естественно попытаться найти какой-либо универсальный метод синтеза, которым можно было бы пользоваться при любом базисе. Одним из таких методов является метод перебора логических схем, составленных сначала из одного элемента базиса, затем из двух, из трех и т. д. При этом проверяется реализуемость функций $g_1, \ldots, g_m$ перебираемыми схемами. Очевидно, что в конце концов решение будет получено и можно говорить о приципиальной возможности решения задачи синтеза. Однако эффективность этого метода настолько низка, что не приходится говорить о его практическом применении. Поэтому рассматривают задачу синтеза для конкретного базиса. Отдается предпочтение следующим полным базисам: конъюнкция, дизъюнкция и отрицание, штрих Шеффера, стрелка Пирса.

Рассмотрим реализацию автоматов без памяти в базисе конъюнкция, дизъюнкция, отрицание. Известно, что каждую булеву функцию можно представить в виде дизъюнктивной или конъюнктивной нормальной формы. Эти формы можно рассматривать как соответствующую суперпозицию элементарных автоматов с функциями выхода: конъюнкцией, дизъюнкцией и отрицанием.

Пример 1. Пусть дан конечный автомат $\mathfrak{M}$, его функции выхода представлены в табл 1.

Таблица 1

x_1	x_2	y_1	y_2
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	1

Построить схему, его реализующую.

Выпишем совершенные к. н. ф. функций выходов y_1 и y_2 автомата $\mathfrak{M}$. Они имеют вид $y_1 = x \vee x_2$ и $y_2 = (x_1 \vee x_2)(\bar{x} \vee x_2)$ в силу известного соотношения. Эти формулы указывают суперпозицию автоматов базиса, в результате которой получается схема с функциями выхода y_1 и y_2. Графически схема представлена на рис. 6.

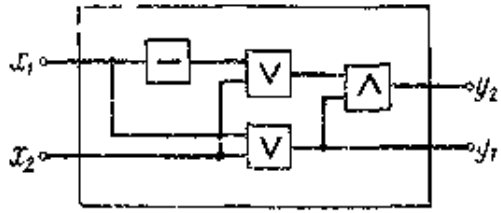


Рис. 6

Используя методы получения минимальных д. н. ф. и к. н. ф., можно упростить схему, уменьшив число элементарных автоматов.

Пример 2. Дан конечный автомат $\mathfrak{M}$, функции выхода представлены в табл. 2

Таблица 2

x_1	x_2	x_3	y_1	y_2
0	0	0	1	0
0	0	1	0	0
1	0	0	1	1
1	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	1	0	0	1
1	1	1	1	1

Построить схему, его реализующую.

Минимально д. н. ф. функций y_1 и y_2 будут $\overline{x_2}x_3 \vee x_1x_3$ и x_1 соответственно. Схема, реализующая автомат $\mathfrak{M}$, представлена на рис. 7.

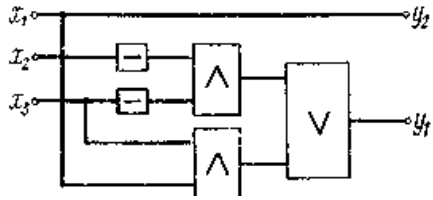


Рис. 7

Ясно, что задача синтеза часто решается неоднозначно, поэтому естественно ввести приятие сложности схемы $\Sigma\mathfrak{M}$, реализующей автомат $\mathfrak{M}$,—величины $L(\Sigma\mathfrak{M})$, являющейся функционалом, и требовать такого решения задачи синтеза, при котором функционал оптимален.

В разобранных примерах функционал определен как число вхождений символов переменных в д. н. ф., представляющую функцию выхода автомата без памяти. В других случаях он может характеризовать надежность схемы или время работы и т. п. Поэтому задачу синтеза можно уточнить так: для любой системы булевых функ-

ций $g_1, \ldots, g_m$ найти схему Σ, реализующую ее, для которой сложность $L(\Sigma)$ экстремальна.

Другой подход к задаче синтеза состоит в отказе от поиска экстремальной схемы для каждой функции и перехода к поиску алгоритмов синтеза экстремальных в некотором классе функций.

Остановимся на этом подходе. В качестве класса функций будем рассматривать булевы функции, зависящие от n переменных. Определим функцию $L(n)$:

$$L(n) = \max_{\text{по всем функциям от } n \text{ переменных}} \min_{\text{по всем схемам, реализующим функцию } f(x_1, \ldots, x_n)} L(\Sigma_f).$$

Здесь $L(\Sigma_f)$ — сложность схемы Σ_f, реализующей функцию f. Функционал $L(n)$ называется *функцией Шеннона.* Очевидно, что любую функцию от n переменных можно реализовать схемой сложности не большей, чем $L(n)$. Рассмотрим случай, когда в качестве базиса используются автоматы, реализующие конъюнкцию, дизъюнкцию и отрицание. Определим $L(\Sigma_f)$ в этом случае как

$$L(\Sigma_f) = n_1 L_\wedge + n_2 L_\vee + n_3 L_-,$$

где n_1 — число элементов конъюнкции, n_2 — число элементов дизъюнкции, n_3 — число элементов отрицания, а $L_\wedge$, $L_\vee$, L_- — сложности элементов конъюнкции, дизъюнкции и отрицания соответственно. Тогда можно указать метод синтеза, при котором

$$L(n) \leqslant \frac{2^n}{n}[1 + 0(1)] L_\vee.$$

Этот метод является наилучшим, так как можно показать, что

$$L(n) \sim \frac{2^n}{n} L_\vee.$$

3. Структурный синтез автоматов с памятью. Будем рассматривать так же, как в случае автоматов без памяти, автоматы, входной и выходной алфавиты которых состоят из символов 0 и 1. Тогда функция выхода такого автомата будет булевой. Чтобы полностью перейти в системе уравнений к булевым функциям, закодируем состояния автомата $q_1, \ldots, q_s$ последовательностями из 0 и 1 длины $l = [\log_2 s]$. Ясно, что при разных способах кодирования состояний получаются разные варианты функций выхода и перехода. Поэтому можно говорить о задаче экстремального кодирования, ибо сложность схемы зависит от вида функций входа и перехода. Если задача кодирования решена, то можно решить задачу синтеза автомата с памятью, сводя ее к решению задачи синтеза автомата без памяти. В качестве исходного базиса рассмотрим базис, состоящий из трех

автоматов без памяти, у которых функции выхода есть конъюнкция, дизъюнкция и отрицание, соответственно и автомата $\mathfrak{M}_3$ с системой

$$\begin{aligned} y(t) &= z(t), \\ z(t+1) &= x(t), \\ z(0) &= 0. \end{aligned} \tag{8}$$

Такой базис удовлетворяет условию полноты. Последнее следует из возможности представить любой автомат с закодированными состояниями в виде схемы, данной на рис. 8.

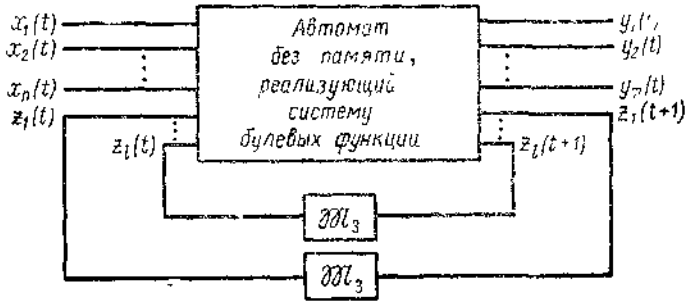


Рис. 8

Такое представление конечного автомата позволяет свести задачу его синтеза к задаче синтеза автомата без памяти. Продемонстрируем это на примере.

Пример 3. Построить схему автомата, данного в примере 1 п.7.1.

У автомата всего два состояния, поэтому кодирование их возможно одним из способов: либо код состояния q_0 — 0, а код состояния q_1 — 1, либо наоборот. Если использовать принцип реализации автомата без памяти, описанный в этом параграфе, и добавить обратные связи через автомат $\mathfrak{M}_3$,то получится схема, представленная на рис. 9.

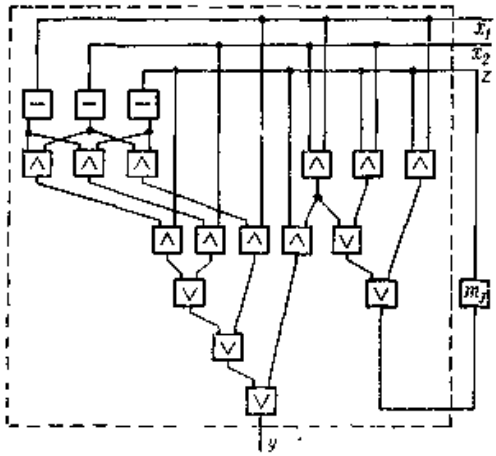


Рис. 9

Рассмотрим отображение F, индуцируемое автоматом.

Отображение F, индицируемое автоматом, удовлетворяет следующим двум условиям:

1) любому входному слову $l_{вх}$ ставится в соответствие выходное слово $l_{вых} = F(l_{вх})$, имеющее с $l_{вх}$ одинаковую длину;

2) если l_1 — начальный отрезок слова $l_{вх}$, то слово $F(l_1)$ является начальным отрезком слова $l_{вых} = F(l_{вх})$.

Эти условия называются условиями *автоматности* отображения (оператора). Всякое отображение (оператор), удовлетворяющее этим условиям, будем называть *автоматным* отображением (оператором). Любое автоматное отображение может быть индуцировано некоторым автоматом (не обязательно конечным).

Пример. Для распределения труб, поступающих из нагревательной печи, между автоматическими линиями обработки служит автомат, называемый делительным устройством. В каждый такт t_j работы транспортера к делительному устройству поступает одна труба, диаметр которой d_j равен одному из n возможных диаметров $d_1, d_2, \ldots, d_n$. Для обработки труб k-го диаметра имеется группа из m_k автоматических линий. Автоматические линии занумерованы двойными номерами: сначала указывается номер группы k (соответствующий трубам данного диаметра d_k), а затем номер ν_k автоматической линии в группе ($k = 1, 2, \ldots, n$; $\nu_k = 1, 2, \ldots, m_k$). Автоматические линии загружаются в порядке очереди, т. е. если в момент t_j труба была направлена в автоматическую линию номер

k, ν_k и в момент t_{j+1} поступила труба того же диаметра, то она направляется в линию номер k, ν_{k+1}, когда $\nu_k < m_k$, или в первую линию данной группы, если $\nu_k = m$.

Функционирование делительного устройства можно описать в виде конечного автомата. Примем в качестве входного алфавита совокупность номеров диаметров труб $1, 2, \ldots, n$, а в качестве выходного алфавита — совокупность номеров автоматических линий k, ν_k. Внутренними состояниями автомата будем считать n-мерные векторы, составляющими которых являются номера ν_k автоматических линий в соответствующих группах, последними получивших для обработки трубу.

Будем считать, что в момент времени $t = 0$ трубы еще не поступили и все автоматические линии свободны. Начальное состояние автомага можно установить в виде $z_0 = (0,0, \ldots, 0)$, текущее состояние в виде $z = (\nu_1, \nu_2, \ldots, \nu_n)$; входные сигналы $x(j) = k_j$; выходные сигналы $y = k, \nu_k$.

Функция переходов в новое состояние $\varphi[z(t-1), x(t)]$ описывается соотношением

$$\nu_x(t) = \nu_x(t-1) + 1 \pmod{m_x} \quad \text{при } k = x,$$

$$\nu_k(t) = \nu_k(t-1) \quad \text{при } k \neq x;$$

функция выходов $\psi[z(t-1), x(t)]$ — соотношением

$$y(t) = (x(t), \nu_x(t)).$$

Если предыдущее состояние автомата $z(t-1)$ и поступающий входной сигнал $x(t)$ определяют не состояние $z(t)$, а распределение вероятностей p_{ij} перехода из состояний $z_i = z(t-1)$ в одно из состоянийв $z_j \in Z$ момент t, то такой автомат называется *автоматом со случайными переходами* (вероятностным автоматом). Для его задания, помимо элементов (X, Y, Z, ψ, z_0), вместо функции φ нужна система матриц вида $\| p_{ij}(x) \|$, определяющая для каждого x матрицу упомянутых распределений вероятностей $\| p_{ij} \|$,

$$p_{ij}(x) = \mathrm{P}\{z(t) = z_j / z(t-1) = z_i,\ x(t) = x\}.$$

Функционирование автомата со случайными переходами имеет следующий вид. Пусть в момент t в автомат, находившийся в состоянии $z(t-1) = z_i$, поступает входной сигнал $x(t) = x$. Тогда, из матрицы $\| p_{ij}(x) \|$ определяются вероятности p_{ij} того, что автомат перейдет в этот момент в состояние z_j; конкретное z_j

выбирается по жребию в соответствии с вероятностями $p_{ij}(x)$. В этот же момент выдается выходной сигнал

$$y(t) = \psi[z(t)].$$

Описание процесса изменения состояний автомата со случайными переходами можно свести к виду

$$z(t) = \varphi^*[z(t-1),\ x(t)],$$

где под φ* понимается случайный оператор, выполняющий последовательно два действия: 1) определение вероятностей $p_{ij}(x)$ и 2) случайный выбор конкретного z_j в соответствии с этими вероятностями.

Рассматриваются также вероятностные автоматы более общего вида, когда начальное состояние z_0 является случайным; оно выбирается в соответствии с распределением вероятностей $p_{01}, p_{02}, \ldots, p_{0k}$, заданным на множестве Z.

Наконец, представляют интерес автоматы, обладающие тем свойством, что по состоянию $z(t)$ определяется не выходной сигнал $y(t)$, а распределение вероятностей $q_t(z)$ на множестве *Y*. Тогда

$$y(t) = \psi^*[z(t)],$$

где ψ^*— случайный оператор, который:

1) определяет по заданному $z(t)$ вероятности $q_t(z)$;

2) выбирает по жребию конкретное $y(t)$ в соответствии с этими вероятностями.

7.6. Модель вероятностного автомата

Вероятностный автомат (ВА) есть объект

$$A = \langle X,\ Y,\ \mathfrak{A},\ \mu(a',\ y/a,\ x)\rangle. \tag{1}$$

Здесь $X = \{x_1, \ldots, x_m\}$ — конечное множество входных символов, $Y = \{y_1, \ldots, y_n\}$ — конечное множество выходных символов, $\mathfrak{A} = \{a_1, \ldots, a_k(, \ldots)\}$ — конечное или счетное множество состояний. Система чисел $\mu(a',\ y/a,\ x)$ удовлетворяет условиям:

$$\mu(a',\ y/a,\ x) \geqslant 0, \quad (a',\ y) \in \mathfrak{A} \times Y, \quad (a,\ x) \in \mathfrak{A} \times X,$$
$$\sum_{(a',y)\in\mathfrak{A}\times Y} \mu(a', y/a, x) = 1, \quad (a, x) \in \mathfrak{A} \times X. \tag{2}$$

При содержательном толковании ВА *A* как преобразователя информации число $\mu(a',\ y/a,\ x)$ означает условную вероятность его перехода из состояния *a* при входном сигнале *x* в состояние *a′* при выходном сигнале *y*.

Введем обозначения

$$\mu_{ij}(y/x) = \mu(a_j,\ y/a_i,\ x), \quad i,\ j = 1,\ \ldots,\ k. \tag{3}$$

Удобно задание ВА в виде системы конечномерных, а в случае счетности множества $\mathfrak{A}$ — счетномерных матриц с неотрицательными элементами:

$$A = \langle A(y/x),\ x \in X,\ y \in Y \rangle, \tag{4}$$

где $A(y/x) = (\mu_{ij}(y/x))$. Здесь, если положить

$$\sum_{y \in Y} A(y/x) = A(x), \tag{5}$$

то матрицы $A(x)$ — стохастические для каждого значения x.

Если ВА (1) рассматривается совместно с начальным состоянием $a_0 \in \mathfrak{A}$, то он называется *инициальным* и обозначается $(A,\ a_0)$ или $\langle X,\ Y,\ \mathfrak{A},\ \mu(a',\ y/a,\ x),\ a_0 \rangle$.

Наряду с состояниями ВА рассматриваются распределения вероятностей его состояний, задаваемые системами чисел $\mu(a)$, удовлетворяющих условиям:

$$\mu(a) \geqslant 0, \quad a \in \mathfrak{A}, \quad \sum_{a \in \mathfrak{A}} \mu(a) = 1. \tag{6}$$

При матричном представлении ВА (4) распределение вероятностей состояний (6) удобно задавать в форме стохастического вектора-строки

$$\mu = (\mu(a_1),\ \ldots,\ \mu(a_k)(,\ldots,)). \tag{7}$$

В частности, распределение вероятностей начальных состояний ВА называется *начальным вектором состояний* и обозначается $\mu(e)$. ВА (4), рассматриваемый совместно с начальным вектором состояний, обозначается $(A,\ \mu(e))$ или $\langle A(y/x),\ x \in X,\ y \in Y,\ \mu(e) \rangle$. Здесь через e обозначено пустое слово. Условимся слова в алфавите X, как правило, обозначать буквой p, возможно, с индексами, слова в алфавите Y — буквой q, также, возможно, с индексами. Длину слова p будем обозначать через $|p|$.

Пусть $p = x_{i_1} \ldots x_{i_s}$, $q = y_{j_1} \ldots y_{j_s}$. Введем обозначение

$$A(q/p) = A\left(y_{j_1}/x_{i_1}\right) \ldots A\left(y_{j_s}/x_{i_s}\right), \tag{8}$$

причем будем полагать, что

$$A(e/e) = E \tag{9}$$

— единичная матрица.

Пусть p_1p_2 и q_1q_2 — результат приписывания слов p_2 и q_2 соответственно к словам p_1 и q_1 справа, причем $|q_1| = |p_1|$, $|q_2| = |p_2|$. Тогда справедливо преобразование

$$A(q_1/p_1)A(q_2/p_2) = A(q_1q_2/p_1p_2). \tag{10}$$

Нас будут интересовать следующие основные характеристики ВА.

1. Введем обозначение

$$\mu(a, q/p) =$$

$$= \sum_{a_0, \ldots, a_{s-1}} \mu(a_0)\, \mu\left(a_1, y_{j_1}/a_0, x_{i_1}\right) \ldots \mu\left(a, y_{j_s}/a_{s-1}, x_{i_s}\right). \quad (11)$$

Если ввести в рассмотрение стохастический вектор-строку $\boldsymbol{\mu}(q/p) = (\mu(a_1, q/p), \ldots, \mu(a_k, q/p)(, \ldots))$, то в матричной форме соотношение (11) будет иметь вид

$$\boldsymbol{\mu}(q/p) = \boldsymbol{\mu}(e) A(q/p). \quad (12)$$

Вектор-строка $\boldsymbol{\mu}(q/p)$ называется *вектором состояний В А* $(A, \boldsymbol{\mu}(e))$. В частности, вследствие формулы (9), если $p = q = e$, то

$$\boldsymbol{\mu}(e/e) = \boldsymbol{\mu}(e), \quad (13)$$

и вектор состояний совпадает с начальным вектором состояний.

При содержательном толковании i-я координата вектора состояний $\boldsymbol{\mu}(q/p)$ означает вероятность реакции ВА $(A, \boldsymbol{\mu}(e))$ на входное слово p выходным словом q и перехода при этом в состояние a_i при условии, что начальное состояние ВА случайно и определено распределением вероятностей $\boldsymbol{\mu}(a)$ (начальным вектором состояний $\boldsymbol{\mu}(e)$).

2. Обозначим через $\mathbf{e}$ вектор-столбец со всеми координатами, равными единице:

$$\mathbf{e}^T = (1, \ldots, 1(, 1, \ldots)). \quad (14)$$

Далее введем обозначение

$$\tau(q/p) = \tau_A^{\boldsymbol{\mu}(e)}(q/p) = \sum_{a \in \mathfrak{A}} \mu(a, q/p). \quad (15)$$

В матричной форме соотношение (13) будет иметь вид

$$\tau(q/p) = \boldsymbol{\mu}(e) A(q/p) \mathbf{e}. \quad (16)$$

Величина $\tau(q/p)$ называется *входно-выходным отношением* ВА $(A, \boldsymbol{\mu}(e))$ или *многотактным каналом,* представленным в ВА $(A, \boldsymbol{\mu}(e))$. В содержательной интерпретации величина $\tau(q/p)$ означает вероятность реакции на входное слово p выходным словом q при заданном начальном векторе состояний $\boldsymbol{\mu}(e)$. Аналогично (13) в случае $p = q = e$ будем имедъ

$$\tau(e/e) = 1. \quad (17)$$

3. Введем обозначение

$$\tau(q/a, p) = \sum_{a' \in \mathfrak{A}} \mu(a', q/a, p), \quad (18)$$

которое в матричной форме будет выглядеть следующим образом:

$$\tau(q/p) = A(q/p) \mathbf{e}. \quad (19)$$

В наших рассмотрениях будут постоянно встречаться как вектор-строки, так и вектор-столбцы. Иногда вместо термина вегктор-столбец используется термин поствектор (в матричном произведении он всегда стоит последним). Вектор-столбец $\tau(q/p)$ называется *поствектором состояний* ВА *A*. Координата $\tau(q/a,\ p)$ поствектора состояний означает вероятность реакции на входное слово *p* выходным словом *q* при фиксированном начальном состоянии *a*.

Из (10) и (12) получаем, что для любых пар слов $|p_1| = |q_1|,\ |p_2| = |q_2|$

$$\mu(q_1q_2/p_1p_2) = \mu(q_1/p_1)A(q_2/p_2). \tag{20}$$

Далее, из (10), (16) и (19) следует, что

$$\tau(q_1q_2/p_1p_2) = \mu(q_1/p_1)\tau(q_2/p_2). \tag{21}$$

В частности,

$$\tau(q/p) = \mu(e)\tau(q/p), \quad \tau(q/p) = \mu(q/p)\mathbf{e}. \tag{22}$$

Представляет интерес выделить случай, когда выходные последовательности символов ВА игнорируются. Такой ВА называется ВА *без выхода*. Формально ВА без выхода есть объект

$$A = \langle X,\ \mathfrak{A},\ \mu(a'/a,\ x)\rangle, \tag{23}$$

где множества *X* и $\mathfrak{A}$ определяются, как и ранее, а система чисел $\mu(a'/a,\ x)$ удовлетворяет условиям

$$\mu(a'/a, x) \geqslant 0, \quad a' \in \mathfrak{A}, \quad \sum_{a' \in \mathfrak{A}} \mu(a'/a, x) = 1. \tag{24}$$

В матричной форме ВА без выхода определяется как система стохастических матриц

$$A = \langle A(x),\ x \in X\rangle, \tag{25}$$

счетномерных в случае счетности множества состояний $\mathfrak{A}$ (ср. с (4) и (5)). Вновь будем полагать для слова $p = x_{i_1} \ldots x_{i_s}$ что

$$A(x_{i_1}) \ldots A(x_{i_s}) = A(p), \tag{26}$$

причем

$$A(e) = E. \tag{27}$$

Рассмотрим основные характеристики ВА без выхода.

1. Определим *вектор состояний* (ср. с (12)) как вектор

$$\mu(p) = \mu(e)A(p). \tag{28}$$

Стохастический вектор $\mu(p)$ определяет распределение вероятностей состояний при условии, что распределение вероятностей начальных состояний определяется *начальным вектором состояний* $\mu(e)$ и на вход ВА поступило слово *p*. Аналогично (20) получаем из (26) и (28)

$$\mu(p_1p_2) = \mu(p_1)A(p_2). \tag{29}$$

2. Пусть $F = \{a_{i_1}, \ldots, a_{i_r}(, \ldots)\}$ — некоторое подмножество множества состояний $\mathfrak{A}$, $F \subset \mathfrak{A}$. Будем обозначать через $\mathbf{n}_F$ поствектор, определенный следующими условиями:

$$n_F^i = \begin{cases} 1, & a_i \in F, \\ 0, & a_i \notin F. \end{cases} \tag{30}$$

Поствектор (30) называется *решающим поствектором* для ВА без выхода *A*. Введем обозначение

$$\chi(p) = \chi_A^{\mu(e), F}(p) = \sum_{\substack{a_0 \in \mathfrak{A} \\ a \in F}} \mu(a_0)\, \mu(a/a_0, p). \tag{31}$$

В матричной форме соотношение (31) примет следующий вид:

$$\chi(p) = \boldsymbol{\mu}(e) A(p) \mathbf{n}_F. \tag{32}$$

Из формулы (31) видна содержательная интерпретация величины $\chi(p)$. Это вероятность того, что состояние ВА без выхода $(A, \boldsymbol{\mu}(e))$ оказалось в множестве состояний *F* после подачи на вход слова *p*. Вероятность $\chi(p)$ называется *характеристической функцией* ВА.

3. Положим

$$\tau(a, p) = \sum_{a' \in F} \mu(a'/a, p). \tag{33}$$

В матричной форме соотношение (33) будет иметь вид

$$\boldsymbol{\tau}(p) = A(p) \mathbf{n}_F. \tag{34}$$

i-я координата поствектора $\boldsymbol{\tau}(p)$ есть вероятность попадания вероятностного автомата *A* в множество состояний *F* из начального состояния *a,* после подачи на вход слова *p*.

Из соотношений (28), (32) и (34) видно, что справедливы формулы

$$\chi(p_1 p_2) = \boldsymbol{\mu}(p_1) \boldsymbol{\tau}(p_2), \tag{35}$$

$$\boldsymbol{\tau}(p_1 p_2) = A(p_1) \boldsymbol{\tau}(p_2). \tag{36}$$

Таким образом, с каждым ВА общего вида $(A, \boldsymbol{\mu}(e))$ ассоциируется некоторое множество векторов специального вида

$$L_A = \{\boldsymbol{\mu}(q/p),\ (p,\ q) \in (X \times Y)^*\}. \tag{37}$$

Обозначение $(X \times Y)^*$, употребляемое впредь в такой форме для краткости, расшифровывается как мпожество упорядоченных пар слов *p* и *q,* принадлежащих соответственно множествам *X** и *Y** и имеющих одинаковые длины.

Аналогично с каждым ВА без выхода ассоциируется множество стохастических векторов

$$L_A = \{\boldsymbol{\mu}(p),\ p \in X^*\}. \tag{38}$$

Симметрия, наблюдающаяся в строении формул (21) и (35), позволяет дать им двойственное толкование. Введем в рассмотрение множества поствекторов

$$\mathscr{L}_A = \{\tau(q/p), \quad (p, q) \in (X \times Y)^*\}, \tag{39}$$

$$\mathscr{L}_A = \{\tau(p), \quad p \in X^*\} \tag{40}$$

соответственно в случае ВА общего вида и ВА без выхода. В соответствии с формулой (19) для ВА общего вида и формулой (34) для ВА без выхода аналогично множествам (37) и (38) с ВА A ассоциируется множество поствекторов (39) и множество поствекторов (39). Множества (37), (38) и аналогично множества (39), (40) обозначаются одинаковым образом. Это не приводит к недоразумениям, так как они встречаются в разных задачах и из контекста обычно ясно, о каком множестве идет речь.

Обозначим через E_A линейное пространство, натянутое на множество векторов L_A. Аналогично обозначим через $\mathscr{E}_A$ линейное пространство, натянутое на мпожество поствекторов $\mathscr{L}_A$. Иначе говоря, обозначая через Lin V линейную оболочку множества V, положим $E_A = \mathrm{Lin}\, L_A$, $\mathscr{E}_A = \mathrm{Lin}\, \mathscr{L}_A$. В том случае, когда рассматриваемый ВА имеет конечное число состояний, размерность линейных пространств E_A и $\mathscr{E}_A$, очевидно, будет конечна. В таких случаях применяются также обозначения вида E^k и $E^{(k)}$, где натуральный параметр k, взятый без скобок, означает размерность пространства, а взятый в скобки — число координат элементов пространства, заданного в виде векторного пространства. В случае счетномерного линейного пространства скобка опускается.

Пусть последовательность векторов $\mu(q_1/p_1), \ldots, \mu(q_s/p_s)$ обравует базис линейного пространства E_A. Матрица полного ранга

$$M(e) = \begin{pmatrix} \mu(q_1/p_1) \\ \cdot \quad \cdot \quad \cdot \\ \mu(q_s/p_s) \end{pmatrix} \tag{41}$$

называется *базисной матрицей* пространства E_A. Соответственно пусть последовательность поствекторов $\tau(q_1/p_1), \ldots, \tau(q_r/p_r)$ образует базис линейного пространства $\mathscr{E}_A$. Матрица полного ранга

$$N(e) = [\tau(q_1/p_1), \ldots, \tau(q_r/p_r)] \tag{42}$$

называется *базисной матрицей* пространства $\mathscr{E}_A$. Аналогичные базисные матрицы рассматриваются и для ВА без выхода. Поскольку $M = M(e)$ является базисной матрицей, то для каждого вектора $\mu(q/p)$ найдется такой вектор ее, что верно равенство

$$\mu(q/p) = \alpha M. \tag{43}$$

Обозначим через $M(q/p)$ матрицу

$$M(q/p) = \begin{pmatrix} \mu(q_1 q/p_1 p) \\ \cdots \\ \mu(q_s q/p_s p) \end{pmatrix} = M(e) A(q/p). \qquad (44)$$

Тогда для любой пары слов из $(X \times Y)^*$ следствием (43) будет соотношение

$$\mu(qq'/pp') = \alpha M(q'/p'). \qquad (45)$$

В частности, если начальный вектор состояний $\mu(e)$ представляется через базисную матрицу M в виде

$$\mu(e) = \alpha(e) M, \qquad (46)$$

то для произвольного вектора состояний ВА получим

$$\mu(q/p) = \alpha(e) M(q/p). \qquad (47)$$

Аналогично получаются двойственные соотношения. Положим

$$N(q/p) = (\tau(qq_1/pp_1) \ldots \tau(qq_r/pp_r)). \qquad (48)$$

Если поствектор $\mathbf{e}$ представляется через базисную матрицу в виде

$$\mathbf{e} = N\beta(e), \qquad (49)$$

то для произвольного поствектора состояний $\tau(q/p)$ получим

$$\tau(q/p) = N(q/p)\beta(e). \qquad (50)$$

Из соотношения (43) вытекает, что каждая пара базисных матриц M_1 п M_2 связана соотношением

$$M_1 = RM_2, \qquad R = \begin{pmatrix} \alpha_1 \\ \cdots \\ \alpha_s \end{pmatrix}, \qquad (51)$$

где R — неособенная матрица. Аналогичное соотношение имеет место для базисных матриц пространства $\mathscr{E}_A$:

$$N_1 = N_2 Q, \quad Q = (\beta_1 \ldots \beta_r). \qquad (52)$$

Здесь также матрица Q — неособенная. Аналогичные соотношения могут быть получены и для ВА без выхода.

Рассмотрим еще один способ задания ВА, принятый в теории автоматов,— графический. Задание автомата посредством графа удобно своей наглядностью и для некоторых несложных графов позволяет просто рассчитывать вероятности переходов. Пусть $A = \langle X, Y, \mathfrak{A}, \mu(a', y/a, x)\rangle$ — конечный ВА. *Граф ВА* $\Gamma(A)$ строится следующим образом: множество вершин графа $\Gamma(A)$ совпадает с множеством состояний ВА так, что каждая вершина взаимно однозначно соответствует определенному состоянию. Для каждой пары символов (x, y) в том и только том случае, если вероятность перехода $\mu(a', y/a, x)$ не равна нулю, граф $\Gamma(A)$ содержит дугу (a, a'), помеченную вероятностью $\mu(x, y) = \mu(a', y/a, x)$. Аналогично в случае ВА без выхода дуга помечается вероятностью

$\mu(x) = \mu(a'/a,\ x)$. Вероятности $\mu(x,\ y)$ или $\mu(x)$ удобно читать так, как если бы значения вероятностей были коэффициентами при символах, записанных в скобках.

На рис. 1 изображен граф ВА с тремя состояниями, двумя входными, двумя выходными символами и со следующими матрицами переходов:

$$A(y_1/x_1) = \begin{pmatrix} 0 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}, \quad A(y_2/x_1) = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{pmatrix},$$

$$A(y_1/x_2) = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 0 & 1/2 \\ 0 & 1/2 & 0 \end{pmatrix}, \quad A(y_2/x_2) = \begin{pmatrix} 0 & 0 & 1/2 \\ 0 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{pmatrix}.$$

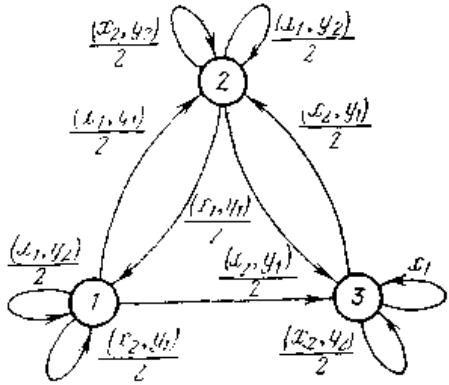


Рис. 1

Пусть путь W на графе ВА образован последовательностью дуг W_i: $W = \{W_1, \ldots, W_s\}$, причем каждая дута W_i помечена вероятностью $\mu(a_{i+1},\ y_i/a_i,\ x_i)$ и соединяет вершину a_i с вершиной a_{i+1}, $a_1 = a$, $a_{s+1} = a'$. Тогда вероятность пути W, $p(W)$ вычисляется как произведение вероятностей, которыми помечены все дуги, составляющие этот путь:

$$p(W(a, a')) = \prod_{i=1}^{s} \mu(a_{i+1}, y_i/a_i, x_i). \qquad (53)$$

Аналогичная формула имеет место для ВА без выхода:

$$p(W(a, a')) = \prod_{i=1}^{s} \mu(a_{i+1}/a_i, x_i). \qquad (54)$$

Дадим определения некоторых частных типов ВА. Рассмотрим частные условные вероятностные распределения

$$\mu(a'/a, x) = \sum_{y \in Y} \mu(a', y/a, x),$$

$$\mu(y/a, x) = \sum_{a' \in \mathfrak{A}} \mu(a', y/a, x).$$

ВА называется *ВА типа Мили,* если выполнено условие

$$\mu(a',\ y/a,\ x) = \mu(a'/a,\ x)\mu(y/a,\ x). \tag{55}$$

Иначе говоря, условные случайные коды $\xi = a'(a,\ x)$ и $\eta = y(a,\ x)$ — последующее состояние ВА и выходной символ — взаимно независимы для любого значения входного символа x и текущего состояния a. Положим

$$\mu(y/a,\ x,\ a') = \mu(a',\ y/a,\ x)/\mu(a'/a,\ x)$$

всюду, где $\mu(a'/a,\ x) \neq 0$. Тогда для ВА типа Мили получаем, что при $\mu(a'/a, x) \neq 0$

$$\mu(y/a,\ x,\ a') = \mu(y/a,\ x).$$

ВА называется *ВА типа Мура,* если всюду, где $\mu(a'/a,\ x) \neq 0$, выполнено соотношение

$$\mu(y/a,\ x,\ a') = \mu(y/a').$$

Таким образом, для ВА типа Мура имеем

$$\mu(a',\ y/a,\ x) = \mu(a'/a,\ x)\mu(y/a'). \tag{56}$$

ВА называется ВА *со случайными реакциями* (с *детерминированной функцией переходов),* если условное вероятностное распределение $\mu(a'/a,\ x)$ принимает только значения 0 или 1. Отсюда уже следует, что существует такая функция $\delta(a,\ x)$, что

$$\mu(a'/a, x) = \begin{cases} 1, & \text{если } a' = \delta(a, x), \\ 0 & \text{в противном случае.} \end{cases}$$

Нетрудно показать, что ВА со случайными реакциями является ВА типа Мили.

Условие детерминированности поведения ВА при переходе из состояния в состояние или формировании выходного символа можно ослабить. Пусть существует такая функция $a' = \varphi(a, x, y)$, что условное вероятностное распределение $\mu(a',\ y/a,\ x)$ отлично от нуля тогда и только тогда, когда для заданных пар $(a,\ x)$ и $(a',\ y)$ имеем $a' = \varphi(a,\ x,\ y)$. Тогда ВА называется *полудетерминированным ВА.* Такой ВА, вообще говоря, не обладает детерминированной функцией переходов, однако если выходной символ ВА известен, то внутреннее состояние a' по тройке $(a,\ x,\ y)$ может быть точно установлено.

ВА называется *марковским автоматом (с детерминированной функцией выходов),* если условное вероятностное распределение

$\mu(y/a,\ x)$ принимает только значения 0 или 1. Следовательно для некоторой функции $\lambda(a,\ x)$

$$\mu(y/a, x) = \begin{cases} 1, & \text{если } y = \lambda(a, x), \\ 0 & \text{в противном случае.} \end{cases}$$

Марковский автомат также является ВА типа Мили.

Частным случаем марковского автомата является бернуллиевский автомат или бернуллиевский датчик, источник случайных кодов. Для бернуллиевского автомата все строки матриц перехода равны между собой, следствием чего является независимость распределения вероятностей последующих состояний от начального распределения вероятностей состояний. Бернуллиевский автомат есть ВА типа Мили, однако чаще он рассматривается вообще без выхода.

В теории распознавания важную роль играет ВА общего вида с одноэлементным множеством состояний. Такой ВА называется *управляемым источником случайных кодов* и задается в форме $V = \langle X,\ Y,\ \mu(y/x)\rangle$. Поскольку управляемый источник обладает единственным состоянием, то является ВА с детерминированной функцией переходов и ВА типа Мили.

В теоретических построениях распознавания важную роль играют два типа ВА со счетным множеством состояний, определяемых специальным образом. Введем некоторую нумерацию $N(p)$: $X^* \to N$ слов свободной полугруппы X^*. Для определенности удобно приписать пустому слову номер единица, а остальные слова упорядочить по лексикографическому принципу, установив приоритет среди букв алфавита X и полагая, что слова большей длины имеют большие номера. Часто вместо номера $N(p)$ используется само слово p.

Будем рассматривать матрицы счетного порядка, строки и столбцы которых нумеруются в соответствии с выбранной нумерацией. В строке с номером p_1 и столбце с номером p_2 расположен элемент матрицы, который обозначается $d_{p_1 p_2}$.

Свободным ВА *без выхода* называется система счетномерных стохастических матриц $D = \langle D(x),\ x \in X\rangle$, где каждая матрица $D(x) = \left(d_{p_1 p_2}(x)\right)$ определена условием:

$$d_{p_1 p_2}(x) = \begin{cases} 1, & \text{если } p_2 = p_1 x, \\ 0 & \text{в противном случае.} \end{cases} \qquad (57)$$

Из определения видно, что свободный ВА без выхода является фактически детерминированным автоматом. Его дополнительные возможности по сравнению с детерминированным свободным автоматом заключаются в отсутствии ограничений на выбор начального вектора состояний.

Введем некоторую нумерацию $N(p, q)$: $(X \times Y)^* \to N$ пар слов (p, q) одинаковой длины из свободных полугрупп X^* и Y^* соответственно. В этом случае также естественно выбирать лексикографическую нумерацию пар слов, дополнительно полагая, что слова во входном алфавите обладают приоритетом перед словами той же длины в выходном алфавите. Как и выше, в качестве номера пары мы будем рассматривать саму пару слов (p, q). Будем рассматривать матрицы счетного порядка, строки и столбцы которых нумеруются парами слов одинаковой длины (p, q) в соответствии с выбранной нумерацией. В строке с номером (p_1, q_1) и столбце с номером (p_2, q_2) расположен элемент матрицы, который обозначается $d_{(p_1,q_1)(p_2,q_2)}$.

Пусть дана система счетномерных матриц с неотрицательными элементами

$$D = \langle D(y/x),\ x \in X,\ y \in Y \rangle, \tag{58}$$

удовлетворяющих следующим условиям:

1) Если $d_{(p_1,q_1)(p_2,q_2)}(y/x)$ — произвольный элемент матрицы $D(y/x)$, то при $p_2 \neq p_1 x$ и $q_2 \neq q_1 y$

$$d_{(p_1,q_1)(p_2,q_2)}(y/x) = 0; \tag{59}$$

2)
$$\sum_{y \in Y} D(y/x)\,\mathbf{e} = \mathbf{e}. \tag{60}$$

Иначе говоря, каждая матрица вида

$$D(x) = \sum_{y \in Y} D(y/x)$$

— стохастическая. Система матриц (58) есть *свободный ВА общего вида.* Условие (59) означает, что свободный ВА общего вида есть ВА с детерминированной функцией переходов. Отсюда вытекает, что он является ВА типа Мили.

Введем в рассмотрение новый математический объект.

Общим линейным автоматом (ОЛА) над полем P называется детерминированный автомат (ДА) $L = \langle X, Y, \mathfrak{A}, \delta(a, x), \lambda(a, x) \rangle$, где X — конечное множество входных символов, Y — линейное пространство выходных символов над полем P, $\mathfrak{A}$ — линейное пространство состояний над полем P, $\delta(a, x)$ — линейный оператор в $\mathfrak{A}$, определяющий функцию переходов автомата, и $\lambda(a, x)$ — линейное отображение $\mathfrak{A}$ на Y, определяющее выходной символ автомата.

Размерностью ОЛА L называется размерность пространства $\mathfrak{A}$. *Размерностью выхода* называется размерность пространства Y. В том случае, когда размерности пространств $\mathfrak{A}$ и Y конечны, отображения $\delta(a, x)$ и $\lambda(a, x)$ могут быть заданы посредством

систем матриц конечного порядка. Пусть k и n — соответственно размерности линейных пространств $\mathfrak{A}$ и Y. Тогда существует система $k \times k$-матриц $L(x)$ и система $k \times n$-матриц $T(x)$ такие, что отображения $\delta(a, x)$ и $\lambda(a, x)$ задаются в форме

$$\mathbf{a}' = \delta(\mathbf{a}, x) = \mathbf{a}L(x), \quad \mathbf{y} = \lambda(\mathbf{a}, x) = \mathbf{a}T(x). \tag{61}$$

Естественно в этом случае определить ОЛА как систему матриц

$$L = \langle L(x), T(x), x \in X \rangle. \tag{62}$$

В соответствии с уже установленными обозначениями типа (26) получаем расширение отображений (61) на всю свободную полугруппу X^*:

$$\mathbf{a}(p) = \mathbf{a}(e)L(p), \quad \mathbf{y}(px) = \mathbf{a}(p)T(x). \tag{63}$$

Выделим тот частный случай, когда линейное пространство выходных символов Y одномерно, а выходное отображение $\lambda(a, x)$ не зависит от входа x. Мы получаем детерминированный автомат вида

$$L = \langle L(x), \mathbf{m}, x \in X \rangle, \tag{64}$$

который называется *линейным автоматом* (ЛА). Здесь поствектор $\mathbf{m}$ реализует линейное выходное отображение.

С каждым конечным ВА типа Мили ассоциирован некоторый ОЛА. Пусть $A = \langle A(y/x), x \in X, y \in Y \rangle$ — конечный ВА типа Мили. Положим, что

$$T(x) = (A(y_1/x)\mathbf{e} \ldots A(y_n/x)\mathbf{e}) \tag{65}$$

— система $k \times n$-матриц со стохастическими строками, где n — число выходных символов A. Пусть, далее, $\mathfrak{A}$ — линейное пространство, натянутое на множество векторов L_A, и $\mathfrak{B}$ — линейное пространство, натянутое на множество векторов $\{\boldsymbol{\mu}(p)T(x), px \in X^*\}$. Тогда с ВА типа Мили A *ассоциирован* ОЛА $L = \langle X, \mathfrak{B}, \mathfrak{A}, \boldsymbol{\mu}A(x), \boldsymbol{\mu}T(x) \rangle$.

Для ВА типа Мили характеристика $\boldsymbol{\mu}(a', y/a, x)$ распадается на две характеристики $\boldsymbol{\mu}(a'/a, x)$ и $\boldsymbol{\mu}(y/a, x)$, которые и представлены элементами матриц соответственно $A(x)$ и $T(x)$. Для фиксированного начального вектора состояний $\boldsymbol{\mu}(e)$ наряду с вектором состояпий

$$\boldsymbol{\mu}(px) = \boldsymbol{\mu}(p)A(x)$$

будем рассматривать вектор выхода ВА типа Мили

$$\mathbf{y}(px) = \boldsymbol{\mu}(p)T(x).$$

Выписанные соотношения задают функцию переходов и функцию выхода ОЛА L, ассоциированного с ВА типа Мили A. Таким образом, ОЛА L, ассоциированный с ВА типа Мили A, вычисляет вектор состояний $\boldsymbol{\mu}(p)$ и вектор выхода ВА $\mathbf{y}(px)$ типа Мили A для любого входного слова $px \in X^*$.

Конечный ВА без выхода *(A,* $\mathbf{m}$), рассматриваемый совместно с решающим поствектором $\mathbf{m}$, является линейным автоматом. Действительно, в этом случае функция переходов ВА, как и прежде, определена соотношением

$$\boldsymbol{\mu}(px) = \boldsymbol{\mu}(p)A(x),$$

а характеристическая функция вычисляется или по формуле $\chi_A(p) = \boldsymbol{\mu}(e)A(p)\mathbf{m}$, где $\boldsymbol{\mu}(e)$— начальный вектор состояний, или по формуле

$$\chi_A(p) = \mu(p)m.$$

Таким образом, характеристическая функция $\chi_A(p)$ в этом случае выступает как частный случай выходного отображения $\mathbf{y} = \boldsymbol{\lambda}(\boldsymbol{\mu}(e),\ p)$ для ОЛА.

7.7. Инициальная эквивалентность вероятностных автоматов

Важнейшей характеристикой ВА является многотактный канал, который он представляет. Поэтому естественно считать эквивалентными ВА, представляющие один и тот же многотактный канал. Таким образом, мы приходим к одному из возможных определений эквивалентности ВА.

Рассмотрим понятие инициальной эквивалентности.

Пусть $A = \langle X, Y, \mathfrak{A}, \mu_A(a',\ y/a, x)\rangle$ и $B = \langle X, Y, \mathfrak{B},\ \mu_B(b',\ y/b, x)\rangle$— ВА с одинаковыми множествами входных и выходных символов.

Определение 1. Состояние *a* ВА *A эквивалентно* состоянию *b* ВА *B,* если тождественно совпадают многотактные каналы, представляемые соответственно ВА *A* и ВА *B* при фиксированных начальных состояниях *a* и *b:*

$$\tau_A^a(q/p) \equiv \tau_B^b(q/p).$$

Определение 2. ВА *B инициально-эквивалентно вложен (вкладывается)* в ВА *A,* т. е. $B \underset{\approx}{\subset} A$, если для каждого состояния ВА *B* найдется эквивалентное ему состояние ВА *A.*

Определение 3. ВА *A и B инициально-эквивалентны,* если оба они инициально-эквивалентно вложены друг в друга.

Инициальная эквивалентность ВА *A* и *B* обозначается как $A \approx B$.

Теорема 1. *Для каждого ВА A существует инициально-эквивалентный ему ВА B типа Мура. Если ВА A конечен, то конечен и ВА B, причем числа состояний автоматов связаны неравенством* $k_B \leqslant k_B^2 m$, *где* $k_A = |\mathfrak{A}|$, $k_B = |\mathfrak{B}|$ *и* $m = |X|$.

Доказательство. Пусть $A = \langle X, Y, \mathfrak{A}, \mu(a', y/a, x)\rangle$ — ВА. Рассмотрим ВА типа Мура $B = \langle X, Y, \mathfrak{B}, \mu_B(b'/b, x), \mu_B(y/b')\rangle$, где $\mathfrak{B} = \mathfrak{A} \times X \times \mathfrak{A}$,

$$\mu_B(b'/b, x) = \begin{cases} \mu_A(a_2/a, x), & \text{если } b = (\tilde{a}, \tilde{x}, a_1), \quad b' = (a, x, a_2), \\ 0 & \text{в противном случае,} \end{cases}$$

$$\mu_B(y/b') = \begin{cases} \mu_A(a_2, y/a_1, x)/\mu_A(a_2/a_1, x), & \text{если } b' = (a_1, x, a_2) \text{ и } \mu_A(a_2/a_1, x) > 0, \\ \tilde{\mu}(y) & \text{в противном случае,} \end{cases}$$

где $\tilde{\mu}(y)$ — произвольное вероятностное распределение.

Покажем, что ВА A и B инициально-эквивалентны, т. е. для каждой пары $a_1, a_2 \in \mathfrak{A}$ и $x \in X$ состояние (a_1, x, a_2) ВА B эквивалентно состоянию a_2 ВА A.

Для слов длины единица имеем

$$\mu_B(y/(a_1, \tilde{x}, a_2)\, x) = \sum_{b'} \mu_B(b'/(a_1, \tilde{x}, a_2), x)\, \mu_B(y/b') =$$

$$= \sum_{a'} \mu_A(a'/a_2, x)\, \mu_A(a', y/a_2, x)/\mu_A(a'/a_2, x) =$$

$$= \mu_A(y/a_2, x), \mu_A(a'/a_2, x) \neq 0.$$

Пусть для слов длины k верно, что

$$\mu_B(q/(a_1, \tilde{x}, a_2), p) = \mu_A(q/a_2, p), \quad |p| = |q|,$$

тогда для слов длины $k + 1$ получим

$$\mu_B(yq/(a_1, \tilde{x}, a_2), xp) = \sum_{b'} \mu_B(b'/(a_1, \tilde{x}, a_2), x)\, \mu_B(y/b')\, \mu_B(q/b', p) =$$

$$= \sum_{a'} \mu_A(a'/a_2, x)\, \mu_A(a', y/a_2, x)/\mu_A(a'/a_2, x)\, \mu_B(q/(a_2, \tilde{x}, a'), p) =$$

$$= \sum_{a'} \mu_A(a', y/a_2, x)\, \mu_A(q/a', p) = \mu_A(yq/a_2, xp), \mu_A(a'/a_2, x) \neq 0.$$

В конечном случае число состояний легко подсчитывается.

Теорема 2. *Для каждого ВА типа Мура A существует инициально-эквивалентный ему марковский ВА типа Мура B. Если ВА A конечен, то конечен и ВА B, причем* $k_B = k_A^n$, *где* $k_A = |\mathfrak{A}|$, $k_B = |\mathfrak{B}|$ *и* $n = |Y|$.

Доказательство. Пусть $A = \langle X, Y, \mathfrak{A}, \mu_A(a'/a, x), \mu_A(y/a')\rangle$ — ВА типа Мура. Рассмотрим ВА $B = \langle X, Y, \mathfrak{B}, \mu_B(b'/b, x), y = \lambda(b')\rangle$, где $\mathfrak{B} = \mathfrak{A} \times Y$. Для $b = (a_1, \tilde{y})$ и $b' = (a_2, y)$ имеем $\mu_B(b'/b, x) = \mu_A(a_2/a_1, x)\mu_A(y/a_2)$, $\lambda(b') = y$. Автоматы A и B инициально-эквивалентны, причем для каждого $y \in Y$ состояния (a, y) эквивалентны состоянию a. Действительно, для слов длины единица имеем

$$\mu_B\,(y/(a, \widetilde{y}), x) = \sum_{b'} \mu_B\,(b'/(a, \widetilde{y}), x)\,\mu_B\,(y/b') =$$

$$= \sum_{b'=(a', y)} \mu_B\,(b'/(a, \widetilde{y}), x) = \sum_{a'} \mu_A\,(a'/a, x)\,\mu_A\,(y/a') = \mu_A\,(y/a, x).$$

По индукции для слов вида xp и yq, где $|p| = |q|$, получим

$$\mu_B\,(yq/(a, \widetilde{y}), xp) = \sum_{b'} \mu_B\,(b'/(a, \widetilde{y}), x)\,\mu_B\,(y/b')\,\mu_B\,(q/b',\ p) =$$

$$= \sum_{b'=(a', y)} \mu_B\,((a', y)/(a, \widetilde{y}), x)\,\mu_B\,(q/(a', y), p) =$$

$$= \sum_{a'} \mu_A\,(a'/a, x)\,\mu_A\,(y/a')\mu_A(q/a', p) = \mu_A\,(yq/a, xp).$$

Ниже мы получим некоторое стандартное представление для конечного ВА. Введем предварительно определения и докажем необходимые леммы.

Пусть $\mathbf{p} = (p_1, \ldots, p_k(, \ldots))$ и $\mathbf{q} = (q_1, \ldots, q_l(, \ldots))$ — конечные или счетные распределения вероятностей (стохастические векторы **p** и **q**). Обозначим через $\mathcal{I}$ и $\mathcal{J}$ соответственно множества индексов координат векторов **p** и **q**.

Определение 4. Распределение вероятностей (стохастический вектор) **p** *имплицирует* распределение вероятностей (стохастический вектор) **q**, если множество индексов $\mathcal{I}$ можно так разбить на систему непересекающихся подмножеств индексов $\{\mathcal{I}_s, s \in \mathcal{J}\}$, $\bigcup\limits_{s \in \mathcal{J}} \mathcal{I}_s = \mathcal{I}$, $\mathcal{I}_s \cap \mathcal{I}_t = \varnothing$, $s \neq t$, что

$$\sum_{i \in \mathcal{I}_s} p_i = q_s, \quad s \in \mathcal{J}. \tag{1}$$

Лемма 1. *Для того чтобы распределение вероятностей* **p** *имплицировало распределение вероятностей* **q**, *необходимо и достаточно, чтобы существовало однозначное отображение множества индексов* $\mathcal{I}$ *на множество индексов* $\mathcal{J}$, $\varphi\colon \mathcal{I} \to \mathcal{J}$, *такое, что выполнено условие имплицируемости*

$$\sum_{\varphi(i) = s} p_i = q_s, \quad s \in \mathcal{J}. \tag{2}$$

Доказательство. Пусть **p** имплицирует **q**. Рассмотрим отображение φ, определяемое следующим образом: $\varphi(i) = s$, если $i \in \mathcal{I}_s$. Тогда из (1) следует (2). Предположим, что существует отображение $\varphi\colon \mathcal{I} \to \mathcal{J}$ такое, что выполнено (2). Отображение φ определяет разбиение множества $\mathcal{I}$ так, что выполняется (1).

Отображение φ, определенное условием имплицируемости (2), называется *имплицирующей функцией.*

Лемма 2. *Для любого конечного множества конечных распределений вероятностей* $\Sigma = \{\mathbf{q}_1, \ldots, \mathbf{q}_N\}$ *существует конечное распределение*

вероятностей $\mathbf{q}$, *имплицирующее каждое распределение вероятностей множества* Σ.

Доказательство. Каждому распределению вероятностей $\mathbf{q} = (q_1, \ldots, q_n)$ взаимно однозначно соответствует разбиение Ω полуинтервала $(0, 1]$ на систему полуинтервалов $\left(\sum_{i=0}^{k-1} q_i, \sum_{i=0}^{k} q_i\right]$, $k = 1, \ldots, n$. Рассмотрим произведение $\Omega = \prod_{i=1}^{N} \Omega_i$ разбиений, соответствующих всем распределениям вероятностей $\mathbf{q}_i$, $i = 1, \ldots, N$, и пусть $\mathbf{q}$ — распределение вероятностей, соответствующее Ω. Так как Ω есть подразбиение каждого из разбиений Ω_i, то $\mathbf{q}$ имплицирует каждое из $\mathbf{q}_i$.

Пусть $\mathbf{p} = (p_1, \ldots, p_n)$ — распределение вероятностей и X — множество символов. Произвольное взаимно однозначное отображение $\varphi\colon X \to \mathbf{p}$ называется *источником* или *случайным кодом.* Если в отображении φ вероятности p_x соответствует символ x, то пишем $\mathbf{P}(\varphi = x) = p_x$. Будем говорить, что случайный код φ *имплицирует* случайный код ψ, если распределение вероятностей $(\mathbf{P}(\varphi = x),\ x \in X)$ имплицирует распределение вероятностей $(\mathbf{P}(\psi = y),\ y \in Y)$.

Определение 5. ВА $A = \langle X, Y, \mathfrak{A}, \mu_A(a', y/a, x)\rangle$ *детерминированно-изоморфен* ВА $B = \langle X, Y, \mathfrak{B}, \mu_B(b', y/b, x)\rangle$, если между множествами их состояний можно установить такое взаимно однозначное соответствие $\varphi\colon \mathfrak{A} \to \mathfrak{B}$, что для любой пары состояний a, a' ВА A и любых входных и выходных символов

$$\mu_B(\varphi(a'), y/\varphi(a), x) = \mu_A(a', y/a, x).$$

Ясно, что если ВА детерминпрованно-изоморфны, то они и инициально-эквивалентны. Обратное утверждение, вообще говоря, неверно.

Введем одно частное понятие последовательного соединения автоматов.

Пусть $\Gamma = \langle X, Z, \mu(z/x)\rangle$ — управляемый источник случайных кодов и $B = \langle Z, Y, \mathfrak{A}, \mu_B(a', y/a, z)\rangle$ — конечный вероятностный автомат.

Определение 6. *Последовательным соединением* управляемого источника Г и ВА B называется ВА $A = \langle X, Y, \mathfrak{A}, \mu_A(a', y/a, x)\rangle$, у которого условное распределение вероятностей $\mu_A(a', y/a, x)$ определено соотношением

$$\mu_A(a', y/a, x) = \sum_{z \in Z} \mu(z/x)\, \mu_B(a', y/a, z). \qquad (3)$$

Лемма 3. *ДА является В А типа Мили.*

Доказательство. Пусть $A = \langle X, Y, \mathfrak{A}, \delta(a, x), \lambda(a, x)\rangle$ — ДА. Определим условные распределения вероятностей соотношениями

$$\mu(a'/a, x) = \begin{cases} 1, & \text{если } a' = \delta(a, x), \\ 0, & \text{если } a' \neq \delta(a, x), \end{cases}$$

$$\mu(y/a, x) = \begin{cases} 1, & \text{если } y = \lambda(a, x), \\ 0, & \text{если } y \neq \lambda(a, x). \end{cases}$$

Тогда видно, что ДА A может быть представлен в виде ВА типа Мили $A = \langle X, Y, \mathfrak{A}, \mu(a'/a, x), \mu(y/a, x)\rangle$.

Мы будем далее рассматривать последовательное соединение управляемого источника и ДА.

Теорема 3. *Пусть* $A = \langle X, Y, \mathfrak{A}, \mu_A(a', y/a, x)\rangle$*—ВА с k состояниями. Существует управляемый источник случайных кодов* $\Gamma = \langle X, Z, \mu_\Gamma(z/x)\rangle$ *и ДА с k состояниями* $B = \langle Z, Y, \mathfrak{A}, \delta(a, z), \lambda(a, z)\rangle$ *такие, что ВА А детерминированно-изоморфен последовательному соединению* Γ *и ДА В.*

Доказательство. Рассмотрим семейство случайных кодов $\xi_{a, x}$, принимающих значения пары (a', y) с вероятностью $\mu_A(a', y/a, x)$ для каждого состояния a и фиксированного значения входного символа x. Положим

$$\Sigma_x = \{\xi_{a,x} : \mathbf{P}(\xi_{a,x} = (a', y)) = \mu(a', y/a, x), a \in \mathfrak{A}\}.$$

Поскольку это конечная система случайных кодов, то в соответствии с леммой 2 существует случайный код ζ_x, имплицирующий каждый случайный код семейства Σ_x. Пусть случайный код ζ_x принимает значения z из некоторого конечного множества Z с вероятностями $\mathbf{P}(\zeta_x = z) = \mu(z/x)$. Условие имплицируемостя определяет в соответствии с леммой 1 функцию импликации $(a', y) = \varphi_{a, x}(z)$, которая связывает вероятности значений имплицирующего и имплицируемого случайных кодов соотношением

$$\mu(a', y/a, x) = \sum_{(a', y) = \varphi_{a,x}(z)} \mathbf{P}(\zeta_x = z) = \sum_{(a', y) = \varphi_{a,x}(z)} \mu(z/x).$$

Функцию импликации $\varphi_{a, x}(z)$ можно переписать в форме $a' = \delta_x(a, z)$, $y = \lambda_x(a, z)$.

Рассмотрим множество $Z = \bigcup_x Z_x$ и произведем следующие доопределения.

1. Пусть случайные коды ζ_x принимают значения из множества Z со следующими вероятностями:

$$\mathbf{P}(\zeta_x = z) = \begin{cases} \mu(z/x), & \text{если } z \in Z_x, \\ 0 & \text{в противном случае.} \end{cases}$$

2. Введем в рассмотрение функции $\delta(a, z)$ и $\lambda(a, z)$ условиями

$$\delta(a, z) = \delta_x(a, z), \quad \text{если} \quad z \in Z_x,$$
$$\lambda(a, z) = \lambda_x(a, z), \quad \text{если} \quad z \in Z_x.$$

Пара функций δ и λ, определяет некоторый ДА $B = \langle Z, Y, \mathfrak{A}, \delta(a, z), \lambda(a, z)\rangle$ с k состояниями. Далее, в соответствии с леммой 2 для конечной системы случайных кодов $\Sigma = \{\zeta_x, x \in X\}$ существует случайный код ζ, имплицирующий каждый случайный код из семейства Σ. Пусть он принимает значения u из множества U с вероятностями $\mathbf{P}(\zeta = u)$. Условие имплицируемости в соответствии с леммой 1 определяет функцию импликации $z = \psi(x, u)$, которая связывает вероятности значений имплицирующего и имплицируемого случайных кодов соотношением

$$\mathbf{P}(\zeta_x = z) = \sum_{z=\psi(x,u)} \mathbf{P}(\zeta = u) = \begin{cases} \mu(z/x), & \text{если } z \in Z_x, \\ 0 & \text{в противном случае.} \end{cases}$$

Пусть C есть последовательное соединение управляемого источника $\Gamma = \langle X, Z, \mu(z/x)\rangle$ и ДА B, как это показано па рис. 1.

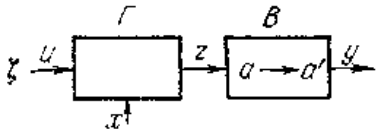


Рис. 1

Вычислим условное распределение вероятностей ВА C. Оно будет равно

$$\mu_C(a', y/a, x) = \sum_{z \in Z_x} \mu(z/x)\, \mu_B(a', y/a, x) =$$
$$= \sum_{z \in Z_x} \mu(z/x)\, \mu_B(a', y/a, x) = \sum_{\substack{a' = \delta_x(a,z) \\ y = \lambda_x(a,z)}} \mu(z/x) = \mu_A(a', y/a, x).$$

Теорема 3 указывает способ конструирования вероятностного автомата как последовательного соединения управляемого источника случайных кодов и детерминированного автомата. Как мы видим, при решении задачи синтеза возникает задача получения распределения вероятностей, имплицирующего каждое распределение из некоторого конечного семейства — задача синтеза имплицирующего вектора. Рассмотрим пример такого синтеза.

Пример 1. В качестве иллюстрации метода синтеза имплицирующего вектора рассмотрим доказательство известной теоремы Кенига о разложении стохастической матрицы в выпуклую линейную комбинацию простых матриц. Линейная комбинация

$$\mathbf{b} = \sum_{i=1}^{n} \alpha_i \mathbf{a}_i$$

векторов $\mathbf{a}_1, \ldots, \mathbf{a}_n$ называется *выпуклой,* если все коэффициенты α_i неотрицательны, причем

$$\sum_{i=1}^{n} \alpha_i = 1.$$

Матрица C — *простая,* если она стохастическая и все ее коэффициенты равны либо нулю, либо единице.

Теорема 4. *Каждая стохастическая $n \times n$-матрица может быть представлена как выпуклая линейная комбинация не более чем $n^2 - n + 1$ простых матриц.*

Доказательство. Пусть матрица A имеет вид

$$A = \begin{pmatrix} \mathbf{p}_1 \\ \cdots \\ \mathbf{p}_n \end{pmatrix} = (p_{ij}).$$

Введем обозначение

$$\alpha_1 = \min_i \max_j (p_{ij}), \quad \alpha_1 > 0.$$

Если C_1 — простая матрица, у которой в каждой строке единица стоит на том месте, где в соответствующей строке матрицы A стоит максимальный элемент (или один из них), то матрица

$$A_1 = \frac{1}{(1 - \alpha_1)} (A - \alpha_1 C_1)$$

стохастическая, причем имеет по крайней мере на единицу больше нулевых элементов, чем матрица A. Применяя тот же алгортм к матрице A_1 и далее последовательно к каждой возникающей в алгоритме стохастической матрице A_i, не более чем за $n^2 - n + 1$ шагов мы придем к тому, что число нулей в полученной стохастической матрице будет не менее $n^2 - n$, т. е. это будет простая матрица. Поэтому получим разложение

$$A = \alpha_1 C_1 + \ldots + \alpha_N C_N, \quad N = n^2 - n + 1. \tag{4}$$

Легко видеть, что распределение вероятностей $\mathbf{p} = (\alpha_1, \ldots, \alpha_N)$ имплицирует каждую строку стохастической матрицы A, причем

единицы в фиксированной строке всех простых матриц разложения определяют группы вероятностей распределения **p**, определяющих в сумме вероятности имплицируемой строки.

Метод, использованный в доказательстве теоремы 4, можно было применить и при доказательстве теоремы 3. В частности, пусть A — ВА без выхода. Тогда, применяя теорему 4 к каждой матрице перехода $A(x)$ ВА A, получим следующее

Следствие 1. *Конечный ВА без выхода детерминированно-изоморфен последовательному соединению управляемого источника случайных кодов и конечного ДА, имеющего то же число состояний, что и В А.*

Доказательство. Формула $A(x) = \alpha_1(x)C_1 + \ldots + \alpha_N(x)C_N$ следует немедленно из формулы (4). Введем целочисленный параметр $u \in U = \{1, \ldots, N\}$ и запишем

$$A(x) = \sum_{u \in U} \alpha(u, x)\, C(u).$$

Для того чтобы узнать в этой формуле соотношение (3), положим $\alpha(u, x) = \mu(u/x)$, обозначим через

$$c_{ij}(u) = \mu_C(a_j/a_i, u)$$

элемент матрицы $C(u)$. Тогда

$$a_{ij}(x) = \mu_A(a_j/a_i, x) = \sum_{u \in U} \mu(u/x)\, \mu_C(a_i/a_j, u).$$

В качестве еще одного примера рассмотрим алгоритм синтеза ВА с рациональными элементами матриц перехода, опирающийся на метод доказательства, примененный в теореме 3. Суть алгоритма понятна из приведенного ниже конкретного примера.

Пример 2. Пусть $\langle M(y/x), x \in X, y \in Y \rangle$ _ ВА, $X = \{x_1, x_2\}$, $Y = \{y_1, y_2\}$ и матрицы $M(y/x)$ определены равенствами

$$M(y_1/x_1) = \begin{pmatrix} 1/4 & 3/8 \\ 3/8 & 1/8 \end{pmatrix}, \quad M(y_2/x_2) = \begin{pmatrix} 1/8 & 1/4 \\ 1/4 & 1/4 \end{pmatrix},$$

$$M(y_1/x_2) = \begin{pmatrix} 3/8 & 1/4 \\ 1/4 & 1/4 \end{pmatrix}, \quad M(y_2/x_2) = \begin{pmatrix} 1/8 & 1/4 \\ 3/8 & 1/8 \end{pmatrix}.$$

Шаг 1. Переводим элементы матриц в двоично-рациональную форму:

$$M(y_1/x_1) = \begin{pmatrix} 0{,}010 & 0{,}011 \\ 0{,}011 & 0{,}001 \end{pmatrix}, \quad M(y_2/x_1) = \begin{pmatrix} 0{,}001 & 0{,}010 \\ 0{,}010 & 0{,}010 \end{pmatrix},$$

$$M(y_1/x_2) = \begin{pmatrix} 0{,}011 & 0{,}010 \\ 0{,}010 & 0{,}010 \end{pmatrix}, \quad M(y_2/x_2) = \begin{pmatrix} 0{,}001 & 0{,}010 \\ 0{,}011 & 0{,}001 \end{pmatrix}.$$

Шаг 2. Разлагаем в линейную комбинацию простых матриц матрицы $A(x_i)$:

$$A(x_1) = \begin{pmatrix} 0{,}010 & 0{,}011 \\ 0{,}011 & 0{,}001 \end{pmatrix} + \begin{pmatrix} 0{,}001 & 0{,}010 \\ 0{,}010 & 0{,}010 \end{pmatrix} =$$

$$= \begin{pmatrix} 0{,}011 & 0{,}101 \\ 0{,}101 & 0{,}011 \end{pmatrix} = \begin{pmatrix} 0 & 0{,}100 \\ 0{,}100 & 0 \end{pmatrix} + \begin{pmatrix} 0{,}011 & 0{,}001 \\ 0{,}001 & 0{,}011 \end{pmatrix}.$$

Поскольку в каждой строке матрицы $A(x_i)$ ровно два слагаемых (в общем случае k), то хотя бы одно слагаемое не меньше 1/2, т. е. в первой позиции после запятой в каждой строке ветречается хотя бы одна единица и можно выделить матрицу типа

$$\begin{pmatrix} 0 & 0{,}100 \\ 0{,}100 & 0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

аналогичная ситуация будет иметь место для матрицы

$$\begin{pmatrix} 0{,}011 & 0{,}001 \\ 0{,}001 & 0{,}011 \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0{,}001 & 0{,}001 \\ 0{,}001 & 0{,}001 \end{pmatrix}$$

(сумма элементов этой матрицы по строкам теперь равна половине, и, соответственно, речь идет о второй позиции после запятой). Это рассуждение справедливо на любом этапе разложения, и мы получим

$$1(x_1) = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \frac{1}{4}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{8}\left\{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right\} =$$

$$= \frac{5}{8}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \frac{3}{8}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$1(x_2) = \begin{pmatrix} 0{,}100 & 0{,}100 \\ 0{,}101 & 0{,}011 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0{,}100 \\ 0{,}001 & 0{,}011 \end{pmatrix} =$$

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0{,}011 \\ 0 & 0{,}011 \end{pmatrix} + \begin{pmatrix} 0 & 0{,}001 \\ 0{,}001 & 0 \end{pmatrix} =$$

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} + \frac{3}{8}\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} + \frac{1}{8}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

В общем случае первая после запятой позиция, в которой имеется хотя бы одна единица, будет встречаться позже, но не более чем через $[\log_2 k]$ позиций, в которых могут оказаться только нули. В целом алгоритм разложения остается тем же самым. ДА *B,* фигурирующий в последовательном соединении (рис. 1), имеет систему простых матриц перехода, получающихся в результате разложения стохастических матриц $A(x_1)$ и $A(x_2)$ на шаге 2. Таким образом, ДА в данном случае имеет 4 входных символа и его таблица переходов определяется системой простых матриц:

$$B(z_1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad B(z_2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B(z_3) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad B(z_4) = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

Таблица переходов состояний автомата *B* имеет вид:

	z_1	z_2	z_3	z_4
a_1	a_2	a_1	a_1	a_2
a_2	a_1	a_2	a_1	a_2

Шаг 3. Следующий шаг состоит в получении функции выхода ДА *B*. Разложим в линейную комбинацию простых матриц, вообще говоря, уже не квадратных, каждую из матриц, полученных следующим образом: для каждого фиксированного $x \in X$ складываем все столбцы каждой из матриц $M(y/x)$ и составляем новые матрицы из полученных столбцов:

$$C(x_1) = (M(y_1/x_1)\, \mathbf{e} M(y_2/x_1)\, \mathbf{e}) = \begin{pmatrix} 0{,}101 & 0{,}011 \\ 0{,}100 & 0{,}100 \end{pmatrix},$$

$$C(x_2) = \begin{pmatrix} 0{,}101 & 0{,}011 \\ 0{,}100 & 0{,}100 \end{pmatrix}.$$

Процедура разложения в линейную комбинацию простых матриц, $C(x_1)$ и $C(x_2)$ равносильна получению имплицирующего вектора для системы случайных кодов $\zeta_{x,a}$ $(x \in X,\ a \in \mathfrak{A})$, определяющих выход ВА *M:*

$$\mathsf{P}(\zeta_{x,a} = y) = \mu(y/x, a).$$

Повторяя процедуру, использованную на предыдущем шаге алгоритма, получаем

$$C(x_1) = C(x_3) = \begin{pmatrix} 0{,}100 & 0 \\ 0 & 0{,}100 \end{pmatrix} + \begin{pmatrix} 0{,}001 & 0{,}011 \\ 0{,}100 & 0 \end{pmatrix} =$$

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0{,}001 & 0 \\ 0{,}001 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0{,}011 \\ 0{,}011 & 0 \end{pmatrix} =$$

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{3}{8}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \frac{3}{8}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}.$$

В нашем примере имплицирующий вектор совпал с имплицирующим вектором для матрицы $A(x_2)$. Поэтому целесообразно ввести обозначения для входной переменной *z* в соответствии с ее обозначениями в разложении для $A(x_2)$. В этом случае таблица выходов ДЛ *B* будет иметь вид:

	z_1	z_2	z_3	z_4
a_1	y_1	произвольное	y_1	y_2
a_2	y_1	произвольное	y_2	y_1

Осталось построить управляемый источник случайных кодов Г, который определяет случайный влод ДЛ *B* в зависимости от значения входной переменной *x* ВА *M*.

Шаг 4. Мы имеем систему из двух случайных кодов ζ_1 с $\mathbf{p}_1 = (5/8,\ 3/8,\ 0,\ 0)$ и ζ_2 с $\mathbf{p}_2 = (1/8,\ 0,\ 1/2,\ 3/8)$. В качестве имплицирующего вектора можно взять распределение вероятностей $\mathbf{p} = (1/8,\ 1/2,\ 3/8)$. Случайный код ζ, принимающий значения $\{u_1, u_2, u_3\}$ с вероятностями $\{1/8,\ 1/2,\ 3/8\}$ соответственно, имплицирует каждый из случайных кодов ζ_1 и ζ_2, определяющих случайный вход ДА *B* при значениях входа ВА *B*, равных x_1 и x_2. Соответствующие функции импликации составляют таблицу выхода управляемого источника случайных кодов Г:

	u_1	u_2	u_3
x_1	z_1	z_1	z_2
x_2	z_1	z_3	z_4

7.8. Некоторые проблемы теории вероятностных автоматов

7.8.1. Проблема редукции

При анализе возможностей ВА как математических моделей объектов статистической природы следует учитывать практические возможности статистического эксперимента. Эксперимент по выяснению принадлежности слова *p* языку $S = T(A, \lambda)$,. представленному в данном ВА *A*, состоит в следующем. Мы должны установить, чему равна вероятность $\chi_A(p) = \mu(e)A(p)\mathbf{n}_F$, и сравнить эту вероятность с константой *K*. Однако при содержательной интерпретации ВА как устройства со случайным поведением приближенное значение вероятности $\chi_A(p)$ может быть получено лишь путем многократного ввода слова *p* в автомат *A* и вычисления частоты получения заключительного состояния из множества *F* в результате этого эксперимента. При этом мы всегда вынуждены ограничить число испытаний некоторым, возможно, достаточно большим числом. Это обстоятельство обусловливает возможность ошибочного заключения о результате эксперимента.

Ситуация была бы приемлемой, если бы можно было для любого сколь угодно малого вещественного $\varepsilon > 0$ указать такое «равномерное»

для всей полугруппы X^* число $N(\varepsilon)$, что для эксперимента с любым словом p вероятность неверного вывода была бы не более ε. Однако более подробный анализ показывает, что в общем случае это невыполнимо.

Действительно, пусть A — ВА и $\lambda \in (0, 1)$. Мы вводим слово p в автомат N раз и подсчитываем число случаев N_1, когда автомат оказывается в состоянии, принадлежащем множеству F. Отношение N_1/N есть выборочное значение случайной величины $\chi_A(p)$ в конкретном эксцерименте, и мы считаем, что $p \in T(A, \lambda)$, если оказалось, что $N_1/N > \lambda$, в противном случае принимается, что $p \notin T(A, \lambda)$. Возможность ошибки заключается в том, что можно принять неверное решение о принадлежности слова p языку $T(A, \lambda)$, поскольку значение N_1/N в эксперименте случайно. Применяя неравенство Чебышева

$$\mathrm{P}(|\xi - M\xi| > \varepsilon) \leqslant \frac{\mathrm{D}\xi}{\varepsilon^2}$$

к случайной величине $\xi = \frac{1}{N}\sum_i \xi_i(p)$, где $\xi_i(p)$ принимает значение 1, если при вводе слова p в i-м эксперименте автомат оказывается в заключительном состоянии из F, и значение 0 в противном случае, получим

$$\mathrm{P}\left(\left|\frac{N_1}{N} - \chi_A(p)\right| > \varepsilon\right) \leqslant \frac{\sqrt{\chi_A \bar{\chi}_A}}{N\varepsilon^2}. \qquad (1)$$

Из формулы (1) видно, что число испытаний N, которое должно быть выполнено, чтобы гарантировать наперед заданную вероятность 1-ε правильного распознавания о принадлежности языку слова p, зависит от значения $\chi_A(p)$. Однако именно в вычислении $\chi_A(p)$ и состоит задача.

Определение 1. Точка сечения $\lambda \in (0, 1)$ называется *изолированной относительно ВА A,* если существует положительное число δ такое, что для всех слов из свободной полугруппы X^* имеем

$$|\chi_A(p) - \lambda| \geqslant \delta. \qquad (2)$$

Из формулы (1) вытекает, что тогда существует такая целочисленная функция N (δ, ε), что для данной изолированной точки сечения λ и любого слова $p \in X^*$ вероятность ошибки при распознавании слова p на основе сравнения числа N_1/N с λ не превышает ε. Это обстоятельство естественным образом приводит к рассмотрению ВА с изолированной точкой сечения и исследованию их возможностей. Верна следующая

Теорема 1 *(редукции). Пусть A—ВА с n состояниями, λ — изолированная точка сечения и язык* $S = T(A, \lambda)$ *представлен в этом автомате точкой сечения* λ.

Тогда язык S регулярен, и число состояний k минимального ДА, представляющего язык S, имеет оценку сверху

$$k \leqslant (1 + 1/\delta)^{n-1}. \tag{3}$$

Доказательство. Пусть слова $p_1, \ldots, p_k$ попарно неэквивалентны относительно языка *S*. Тогда для каждой пары слов p_i и p_j, $i \neq j$, $i, j \leqslant k$, существует слово *p* такое, что $p_i p \in S$, $p_j p \notin S$, или наоборот. Иначе говоря, мы имеем

$$\chi(p_i p) > \lambda, \quad \chi(p_j p) \leqslant \lambda. \tag{4}$$

Но поскольку λ — изолированная точка сечения, то из (2) и (4) следует

$$\chi(p_i p) - \chi(p_j p) \geqslant 2\delta.$$

Учитывая определение $\chi(p)$, получим

$$(\mu(e)A(p_i) - \mu(e)A(p_j))A(p)\mathbf{n}_F \geqslant 2\delta.$$

Поствектор $A(p)\mathbf{n}_F = \mathbf{n}_F(p)$ имеет неотрицательные координаты, не превосходящие единицу. Поэтому

$$2\delta \leqslant (\mu(p_i) - \mu(p_j))\,\mathbf{n}_F(p) \leqslant \sum_{s=1}^{n} |\mu^s(p_i) - \mu^s(p_j)| \quad \text{для} \quad i \neq j. \tag{5}$$

С другой стороны, рассмотрим геометрическую картину отображения, производимого ВА *A*. Пусть σ_i $(1 \leqslant i \leqslant k)$ — множества, определяемые следующим образом: $\sigma_i = \{\mu\colon \mu^s(p_i) \leqslant \mu^s,\ s = 1, \ldots, n, (\mu - \mu(p))e = \delta\}$. Каждое множество σ_i есть параллельный перенос множества $\sigma = \{\mu\colon \mu^s \geqslant 0,\ s = 1, \ldots, n,\ \mu e = \delta\}$, являющегося симплексом $\delta\Delta^{(k)}$. Объем фигуры σ, очевидно, равен $c\delta^{k-1}$, где константа *c* зависит только от размерности пространства E_A и не зависит от δ. Соотношение $(\mu - \mu(p))e = \delta$ можно переписать в виде $\mu e = 1 + \delta$, откуда видно, что все σ_i принадлежат области $\tau = \{\mu\colon \mu^s \geqslant 0,\ s = 1, \ldots, n,\ \mu e = 1 + \delta\}$.

Покажем, что области σ_i, σ_j $(i \neq j)$ не имеют общих внутренних точек. Действительно, из предположения, что μ является внутренней точкой областей σ_i и σ_j $(i \neq j)$, вытекает, что $\mu^s - \mu^s(p_i) > 0$, $\mu^s - \mu^s(p_j) > 0$ для всех $s = 1, \ldots, n$. Следовательно,

$$|\mu^s(p_i) - \mu^s(p_j)| < |\mu^s - \mu^s(p_i)| + |\mu^s - \mu^s(p_j)|,$$

$$\sum_{s=1}^{n} |\mu^s(p_i) - \mu^s(p_j)| < \sum_{s=1}^{n} |\mu^s - \mu^s(p_i)| + |\mu^s - \mu^s(p_j)| = 2\delta,$$

что противоречит (5).

Сравнив объемы фигур σ_i $(i = 1, \ldots, k)$ и τ, получаем $kc\delta^{n-1} =$

$= c(1+\delta)^{n-1}$. Тогда $k \leqslant (1+1/\delta)^{n-1}$.

Обозначим через *H(A)* точечное множество $H(A) = \{\chi_A(p),\ p \in X^*\}$ на $[0,\ 1]$. В тех случаях, когда множество *H(A)* не является всюду плотным на $[0,\ 1]$, появляется возможность рассмотрения изолированных точек сечения λ таких, что все языки $T(A,\ \lambda)$ будут заведомо регулярными. С другой стороны, если точка сечения λ, является точкой сгущения множества *H(A),* то, как мы убедились в начале параграфа, задача распознавания произвольного слова $p \in X^*$ по принадлежности языку $T(A,\ \lambda)$ не разрешима посредством статистического эксперимента с наперед заданной достоверностью. Какое бы натуральное число *N* ни было задано, найдется такое слово *p,* для которого вероятность неверного вывода окажется больше наперед заданной вероятности ошибки ε.

Создается впечатление, что реальное использование ВА не расширяет возможностей ДА по распознаванию слов. Тем не менее теорема редукции не только не обесценивает теорию ВА, но открывает новые возможности ее применений.

Докажем теорему, которая показывает, что в определенном смысле ВА представляют более мощный аппарат представимости языков, нежели детерминированные.

Теорема 2. *Существует ВА A с двумя состояниями и последовательность* $\lambda_n,\ n = 1,\ \ldots,$ *изолированных точек сечения такая, что для каждого n ДА* A_n *с наименьшим числом состояний, представляющий регулярный язык* $S_n = T(A,\ \lambda_n)$, *имеет по крайней мере n состояний.*

Доказательство. Пусть переходные матрицы ВА *A* суть

$$A(0) = \begin{pmatrix} 1 & 0 \\ 2/3 & 1/3 \end{pmatrix}, \quad A(2) = \begin{pmatrix} 1/3 & 2/3 \\ 0 & 1 \end{pmatrix}.$$

В соответствии с известной леммой, если начальный вектор состояний есть $\mu(e) = (1,\ 0)$ и решающий поствектор задан в виде $n_F = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, то для слова $p = \delta_1,\ \ldots,\ \delta_s$ в алфавите $\{0, 2\}$ получим $\chi(p) = 0,\ \delta_s\delta_{s-1}\ldots\delta_1$, где число $\chi(p)$ записано в троичной системе счисления. Поскольку $\delta_j \in \{0,\ 2\},\ j = 1,\ \ldots,\ s,$ то *CH* — топологическое замыкание *H* — есть канторов дисконтинуум. Ввиду этого все точки сечения, принадлежащие дополнению *CH,* являются изолированными точками сечения относительно автомата *A*. Пусть $\lambda_n = 0,\ 22\ldots211$ записано в троичной системе счисления и число цифр равно $(n+1)$. Точка сечения λ_n при любом *n* = 1, ... является изолированной, а для того чтобы имело место неравенство $\lambda_n < \chi_A(p)$, необходимо и достаточно, чтобы слово *p*

имело вид $p = p_1 22 \ldots 2$, где $p_1 \in \{0, 2\}^*$, а число двоек было не меньше n. Итак, множество $S_n = T(A, \lambda_n)$ непусто, и если $p \in S_n$, то $|p| \geqslant n$, а минимальный ДА, представляющий язык с минимальной длиной слова, равной n, имеет по крайней мере $(n + 1)$ состояние.

В связи с доказанной теоремой уместно высказать следующее замечание. При сравнении сложности детерминированных и вероятностных вычислений следует учитывать, что реальная природа представимости языков в ВА и ДА различна. Строго говоря, конечный ВА и конечный ДА как математические модели реальных устройств несравнимы между собой, поскольку различны способы оценки результата их функционирования. При сравнении между собой различных ВА естественно учитывать число экспериментов, которое требуется для получения вывода о результате функционирования с наперед заданной достоверностью, одинаковой для сравниваемых автоматов.

Рассмотрим пример приложения теоремы редукции 1.

Предположим, что предстоит вычисление элементов произведения матриц из некоторого конечного множества $\{A_1, \ldots, A_n\}$ с некоторой наперед заданной точностью ε, с помощью ЭВМ, обладающей конечной памятью при условии, что число сомножителей произведения заранее не ограничено. Для достаточно большого числа сомножителей эта задача не может быть решена при заданном уровне ошибки ε и данном числе разрядов округления N. Покажем, что существуют условия, при которых эта задача может быть решена для любого числа сомножителей с применением теоремы редукции. Если порядок сомножителей определен словом $p = i_1 \ldots i_s$, $i_j \in \{1, \ldots, n\}$, а элемент произведения фиксирован заданием соответствующей строки и столбца, то вычисление элемента $a_{ij}(p)$ произведения матриц $A(p)$ сводится к определению значения словарной функции $\varphi(p) = a_{ij}(p)$ в точке $p \in \{1, \ldots, n\}^*$.

Задача может быть сведена к вычислению значения характеристической функции $X(p) = \alpha^{|p|+1}\varphi(p) + 1/(k+2)$ конечного ВА.

Пример 1. Пусть $\Sigma = \{A_1, \ldots, A_n\}$ — конечное множество стохастических $k \times k$-матриц и пусть $\varepsilon > 0$ — некоторое заданное действительное число. Пусть в дискретные моменты времени $m = 1, \ldots$ подаются матрицы $A_{i_m} \in \Sigma$ и $p = i_1 \ldots i_m$. В каждый момент времени m мы хотим знать с точностью ε элемент $a_{1n}(p)$ произведения матриц $A_{i_1} \ldots A_{i_m}$.

Применим теорему редукции 1 для того, чтобы показать, что при определенных условиях, наложенных на множество матриц Σ,

проблема приближенного вычисления может быть решена для любого ε.

Теорема 3. *Пусть* $X = \{1, \ldots, n\}$ *и словарная функция* χ_p, *где* $p = i_1 \ldots i_m$, *обозначает* $(1, k)$-*элемент произведения матриц* $A_{i_1} \ldots A_{i_m}$. *Предположим, что матричное множество* Σ *таково, что множество* $H(A)$ *нигде не плотно в* $[0, 1]$.

Тогда для каждого $\varepsilon > 0$ *существует целое положительное* N, *действительные* $\lambda_1, \ldots, \lambda_N$ *и ДА* $\bar{A}_1, \ldots, \bar{A}_N$ *над входным алфавитом* X *такие, что*

$$0 = \lambda_1 < \ldots < \lambda_N = 1, \quad \lambda_{i+1} - \lambda_i < \varepsilon, \quad 1 \leqslant i \leqslant n, \tag{6}$$

и $\lambda_i < \chi_p < \lambda_{i+1}$ *тогда и только тогда, когда*

$$p \in T(A_i, \lambda_i)/T(A_{i+1}, \lambda_{i+1}), \quad 1 \leqslant i \leqslant n. \tag{7}$$

Доказательство. То, что множество $H(A)$ нигде не плотно, означает, что его топологическое замыкание $C\Sigma$ не содержит ни одного интервала. Таким образом, для некоторого целого положительного, достаточно большого числа N существует последовательность (6), причем при $2 \leqslant i \leqslant N-1$ $\lambda_i \notin C\Sigma$.

Рассмотрим ВА A над входным алфавитом X, с множеством состояний $\{a_0, a_1, \ldots, a_{n-1}\}$, где a_0 — начальное, а a_{n-1} — финальноесостояния, с системой переходных матриц $A(i) = A_i$, $i = 1, \ldots, n$.

Для любого слова $p = i_1 \ldots i_m$ $\chi(p) = \chi_p$. Числа λ_i $(2 \leqslant i \leqslant N-1)$ являются изолированными точками сечения для автомата A. Таким образом, по теореме 1 все множества $T(A, \lambda_i)$ являются регулярными, представленными некоторыми конечными ДА $\bar{A}_i$. Следовательно, для $2 \leqslant i \leqslant N-1$ имеем, что $\lambda_i < \chi_p$ тогда и только тогда, когда $p \in T(A_i)$ (равенство $\chi_p = \lambda_i$ невозможно, поскольку X, — изолированная точка сечения).

Пусть $\bar{A}_1$ и $\bar{A}_N$ — такие конечные ДА, что $T(\bar{A}_1) = X^*$, $T(\bar{A}_N) = \varnothing$. Тогда система автоматов $\bar{A}_i$ $(i = 1, \ldots, N)$ удовлетворяет условиям теоремы 1.

Используя теорему 3 можно предложить следующий способ приближенного вычисления словарной функции $\chi(p)$. Если дано $\varepsilon > 0$, то пусть λ_i, $\bar{A}_i$ $(1 \leqslant i \leqslant N)$ удовлетворяют условиям этой теоремы. Применяя строго фиксированную величину памяти вычислительной машины, возможно промоделировать ДА $\bar{A}_i$. Когда подаются матрицы произведения $A_{i_1} \ldots A_{i_m}$, то индексы $i_1, \ldots, i_m$ поступают в промоделированные автоматы параллельно. В каждый момент времени вычислительная машина определяет, какой

из автоматов $\bar{A}_i$ представляет слово $p = i_1 \ldots i_m$. Приближенное значение функции $\chi(p)$ определяется как полусумма значений соседних точек сечения λ_i, λ_{i+1}, где индекс i определяется из условий $p \in T(\bar{A}_i)$ и $p \notin T(\bar{A}_{i+1})$.

Топологический анализ условий, при которых оказывается справедливой теорема 1 для конечных ВА, дает возможность сформулировать более общую теорему, определяющую условия редуцируемости для ДА со счетным числом состояний.

Пусть $A = \langle X, \mathfrak{A}, \delta(a, x)\rangle$ — инициальный ДА с множеством входных символов X, счетным множеством состояний $\mathfrak{A}$ и функцией переходов $\delta(a, x)$. Во многих задачах распознавания, в которых автомат A фигурирует в реальном «контексте», множество состояний $\mathfrak{A}$ может оказаться принадлежащим некоторому метрическому пространству E. Именно с такой ситуацией мы сталкиваемся, когда рассматриваем конечный ВА как конечномерный ЛА. Даже в том случае, если рассматривается абстрактный автомат A, всегда можно «погрузить» множество состояний $\mathfrak{A}$ в метрическое пространство. Тривиально это можно выполнить, отобразив множество $\mathfrak{A}$ на натуральный ряд чисел, принимая индекс данного состояния за его образ в отображении. Натуральный же ряд метризуем, и можно предложить метрику

$$\rho(m, n) = \sum_{s=1}^{\infty} \frac{\chi_s(m, n)}{2^s}.$$

Здесь m, n — произвольные натуральные числа, а функция $\chi_s(m, n)$ определена условиями

$$\chi_s(m, n) = \begin{cases} 0, & \text{если } m \equiv n \pmod{s}, \\ 1, & \text{если } m \not\equiv n \pmod{s}. \end{cases}$$

Вообще говоря, всегда можно построить автомат, изоморфный (или, более обще, гомоморфный) данному, такой, что множество состояний последнего окажется компактным множеством в некотором метрическом пространстве. При этом функция переходов автомата $\delta(a, x)$ определяет в множестве $\mathfrak{A}$ некоторую систему автоморфизмов, обладающих свойствами, связанными с наличием метрики в пространстве E. Следующая теорема характеризует условия, при которых такой автомат представляет регулярный язык.

Теорема 4. *(Обобщенная теорема редукции). Пусть ДА* $B = \langle X, \mathfrak{B}, \delta_B(a, x)\rangle$, *представляющий язык S подмножеством своих состояний Q, является изоморфным образом ДА* $A = \langle X, \mathfrak{A}, \delta_A(a, x)\rangle$, *где множество* $\mathfrak{A}$ *есть компактное подмножество некоторого метрического пространства E, и пусть отображение* $a' =$

$= \delta_A(a, x)$ *является «нерастягивающим» в метрике пространства E, т. е.*

$$\rho(a_1', a_2') \leqslant \rho(a_1, a_2) \tag{8}$$

для любой пары состояний a_1, a_2, *для произвольного значения входного символа* $x \in X$.

Пусть R — подмножество $\mathfrak{A}$, *являющееся прообразом множества Q в изоморфизме автоматов A и B и удовлетворяющее условию «изолированности»*

$$(a_1)(a_2)(a_1 \in R_1) \,\&\, (a_2 \notin R) \to \rho(a_1, a_2) \geqslant \delta\} \quad \delta > 0. \tag{9}$$

Тогда язык $S = \{p\colon \delta_A(a_0, p) \in R\}$ *является регулярным. Минимальное число состояний n конечного ДА, представляющего этот язык S, имеет оценку сверху* $n \leqslant 2^{H_{\delta/2}(\mathfrak{A})}$, *где* $H_\varepsilon(\mathfrak{A})$ — ε-*энтропия компактного множества* $\mathfrak{A}$. (ε-энтропией компактного множества $\mathfrak{A}$ называется двоичный логарифм числа элементов минимального покрытия $\mathfrak{A}$ подмножествами $\mathfrak{A}$ диаметра не более 2 ε.)

Доказательство. Введем эквивалентность на множестве слов X^* условием

$$p_1 \equiv_S p_2 = (r)[r \in X^* \to p_1 r \in S \sim p_2 r \in S].$$

Задача заключается в доказательстве конечности множества классов эквивалентности $\equiv_S$ и оценке сверху мощности n этого множества.

Пусть $K_1, \ldots, K_n, \ldots$ — классы эквивалентности $\equiv_S$ и p_1, p_2 — слова из разных классов. По определению эквивалентности, существует слово r такое, что $p_1 r \in S$, $p_2 r \notin S$, т. е. $a_1 = a_0 p_1 r \in R$, $a_2 = a_0 p_2 r \notin R$ (или наоборот). Из условия (9) тогда следует, что $\rho\{a_1, a_2) \rho \geq \delta$. Но в этом случае, в соответствии с условием (8), мы должны получить

$$\delta \leqslant \rho(a_1, a_2) = \rho(a_0 p_1 r, a_0 p_2 r) \leqslant \rho(a_0 p_1, a_0 p_2).$$

Выберем в каждом классе эквивалентности по одному представителю p_i, $i = 1, \ldots$, и рассмотрим множество слов $p_1, p_2, \ldots$ и соответствующее множество состояний $a_0 p_1 = a_1, \ldots, a_n, \ldots$ Среди состояний $a_1, a_2, \ldots$ не может быть совпадающих, так как для любой пары состояний должно найтись слово, переводящее их в заведомо различные состояния так, чтобы выполнялись условия $\rho(a_i, a_j) \geqslant \delta$, $i \neq j$. Таким образом, число различных состояний $a_1, a_2, \ldots$ ограничено сверху максимумом числа элементов в компактном множестве $\mathfrak{A}$ метрического пространства E, попарно удаленных друг от друга не менее чем на δ. Известно, что это число $n_\varepsilon(\mathfrak{A})$, $\varepsilon = \delta/2$, связано с ε-емкостью $h_\varepsilon(\mathfrak{A})$ множества $\mathfrak{A}$ соотношением $h_\varepsilon(\mathfrak{A}) = \log_2 n_\varepsilon(\mathfrak{A})$, а последняя ограничена сверху

ε-энтропией, которая конечна для любого ε в силу компактности множества $\mathfrak{A}$: $h_\varepsilon(\mathfrak{A}) \leqslant H_\varepsilon(\mathfrak{A})$. Отсюда вытекает оценка, завершающая доказательство. (ε-емкостью компактною множества $\mathfrak{A}$ называется двоичный логарифм мощности максимального множества моментов из $\mathfrak{A}$, попарно удаленных друг от друга строго более чем на 2ε.)

Теорема редукции для конечных ВА получается из теоремы 4 как следствие. Если A — конечный ВА, то в качестве множества $\mathfrak{A}$ состояний счетного ДА в теореме 4 следует рассматривать множество векторов состояний $L_A = \{\mu_A(p),\ p \in X^*\}$, где $\mu(e)$ — начальный вектор состояния и $A(x)$ система переходных матриц. Множество R определено условием $\mu(p)\mathbf{n}_F > \lambda$. Метрическое пространство E есть гиперплоскость $\mu\mathbf{e} = 1$, а множество $\mathfrak{A}$ компактно в нем, так как вполне ограничено условиями

$$\mu^i \geqslant 0,\ \sum_{i=1}^{n} \mu^i = 1.$$

Регулярность языка, представленного в конечном ВА A множеством состояний и изолированной точкой сечения λ $(|\chi(p) - \lambda| > \delta)$, является прямым следствием теоремы 4. Действительно, из известных лемм вытекает, что стохастическая матрица реализует отображение гиперплоскости $\mu\mathbf{e} = 1$ в себя, удовлетворяющее условию «нерастягиваемости» (8). С другой стороны, из условия изолированности вытекает, что выполняется условие (9).

Теорема редукции, доказанная непосредственно для ВА, естественно, дает лучшую оценку для числа состояний моделирующего ДА, поскольку в процессе доказательства дополнительно используется условие линейности автомата.

Замечание 1. Обратим внимание на то обстоятельство, что в доказательстве теоремы 4 не использовалось свойство линейности преобразования $a' = \delta(a, x)$, а использовалось только свойство «нерастягиваемости» в заданной метрике. Отсюда следует, что теорема редукции может быть применена к вероятностным дискретным преобразователям с памятью, операторы переходов которых не являются линейными. В качестве одного из следствий теоремы 4 рассмотрим теорему редукции для следующего математического объекта.

Определение 2. Произвольный автоморфизм R: $\Delta^{(n)} \to \Delta^{(n)}$ координатного симплекса $\Delta^{(n)}$ в себя называется *стохастическим* оператором.

Определение 3. Объект $R = \langle X, \mathfrak{A}, R(x)\rangle$, где X — конечное множество входных символов, $\mathfrak{A}$ — множество стохастических

векторов состояний $\mathfrak{A} \subseteq \Delta^{(n)}$, $R(x)$ — множество стохастических операторов, отображающих в себя множество $\mathfrak{A}$, называется *автоматным* оператором над множеством распределений вероятностей (стохастических векторов) $\mathfrak{A}$.

Теорема 5. *Пусть R— автоматный оператор над множеством стохастических векторов* $\mathfrak{A}$, *а система стохастических операторов R(x) и подмножество* $Q \subseteq \mathfrak{A}$ *таковы, что для метрики*

$$\rho(\mu_1, \mu_2) = \max_i \left| \mu_1^i - \mu_2^i \right| \qquad (10)$$

выполнены следующие условия:

1) $\rho(\mu_1 R(x), \mu_2 R(x)) \leqslant \rho(\mu_1, \mu_2)$ *для любых векторов* μ_1, μ_2 *из* $\mathfrak{A}$ *и любого входного символа из X;*

2) $(\mu_1)(\mu_2)[(\mu_1 \in Q) \,\&\, (\mu_2 \notin Q) \to \rho(\mu_1, \mu_2) \geqslant \delta], \ \delta > 0.$

Тогда язык $S = \{p\colon \mu(e)R(p) \in Q\}$ *регулярен. Минимальное число состояний N ДА, представляющего язык S, имеет оценку сверху*

$$N \leqslant C_{n+\rfloor 2/\delta \lfloor - 1}^{\rfloor 2\ \delta \lfloor}.$$

Доказательство. В метрике (10) симплекс $\Delta^{(n)}$ является вполне ограниченным множеством — диаметр его равен единице. Следовательно, это компактное множество, как и его подмножество $\mathfrak{A}$. Поэтому для автоматного оператора *R* как ДА выполняются условия теоремы 4, и язык *S*, представленный в этом автомате множеством состоянии *Q*, является регулярным. Нам остается оценить число состояний минимального ДА, представляющего этот язык.

Пусть $F_m^n = \left\{\mu\colon \mu^i = \frac{k_i}{m},\ i = 1, \ldots, n,\ \sum_{i=1}^{n} k_i = m\right\}$, где k_i и m —

Натуральные числа. Для всякого вектора из симплекса $\Delta^{(n)}$ найдется вектор μ' в множестве F_m^n такой, что расстояние между ними в метрике (10) не превосходит $1/m$: $\rho(\mu, \mu') \leqslant 1/m$. Однако это означает, что семейство векторов F_m^n представляет собой l/*m*-сеть симплекса $\Delta^{(n)}$, откуда следует, что $H_{1/m}(\Delta^{(n)})$ не больше двоичного логарифма числа элементов множества F_m^n. Число элементов $N(n, m)$ множества F_m^n равно числу способов размещения *m* элементов по *n* ячейкам, или C_{n+m-1}^m, поэтому имеем

$$H_{1/m}(\Delta^{(n)}) \leqslant \log_2 C_{n+m-1}^m.$$

H_ε представляет собой монотонно возрастающую функцию ε для $0 < \varepsilon < 1/2$. Выберем $m = \rfloor 1/\varepsilon \lfloor$. Здесь $\rfloor \gamma \lfloor$ означает минимальное целое число, не меньшее чем γ. Тогда $H_\varepsilon(\Delta^{(n)}) \leqslant H_{1/m}(\Delta^{(n)})$. Следовательно,

$$H_\varepsilon\left(\Delta^{(n)}\right) \leqslant \log_2 C_{n+[1/\varepsilon]-1}^{[1/\varepsilon]-1}.$$

Пример 2. Рассмотрим следующую задачу, возникающую в эволюционной генетике. Пусть изучается эволюция популяции, состоящей из очень большого числа особей, по отношению к наличию свойства, определенного рецессивным геном «А», аллель которого «*а*» является рецессивным летальным геном. Следовательно, в популяции существуют только пары вида *«aA», «Aa», «AA»,* частоты которых равны

$$Aa,\ aA \to p,$$
$$AA \to q,\quad p+q=1,\quad p,\ q \geqslant 0.$$

Будем предполагать, что встреча любой пары генов в период размножения во всей популяции равновероятна, т. е. если частоты аллелей равны $a \to p/2,\ A \to q+p/2,$ то встреча пар аллелей определяется схемой независимой выборки пар из совокупности с возвращениями. Следовательно, в процессе эволюции получим сначала пары с частотами

aa	aA	Aa	AA
$p^2/4$	$p/2(q+p/2)$	$p/2(q+p/2)$	$(q+p/2)^2,$

и затем вследствие летальности гена *«aa»* частоты станут равными:

$$Aa, aA \to \frac{2\frac{p}{2}(q+p/2)}{1-p^2/4} = \frac{p}{1+p/2} = p_1,\ AA \to \frac{(q+p/2)^2}{1-p^2/4} = \frac{q+p/2}{1+p/2} = q_1.$$

Итак, оператор преобразования имеет вид

$$R(p, q) = \left(\frac{p}{1+p/2}, \frac{q+p/2}{1+p/2}\right),$$

т. е. является нелинейным. Мы получили автоматный оператор над множеством стохастических векторов вида $\mu(p,\ q).$ Язык $S \subset \{x\}^*,$ определенный в однобуквенном алфавите $\{x\}$ условием

$$\underbrace{x\,x\,\ldots\,x}_{r} \in S \sim R^r(p, q)\begin{pmatrix}0\\1\end{pmatrix} > \lambda,$$

означает перечисление тех поколений популяции, когда частота свойства *«AA»* более λ. В общем случае этот язык не обязан быть регулярным, однако в данном случае он регулярен. В частности, к этому автоматному оператору применима обобщенная теорема редукции 4. Будем рассматривать метрику $\rho(p_1,\ p_2) = |p_1 - p_2|.$ Тогда

$$\left|\frac{p_1}{1+p_1/2} - \frac{p_2}{1+p_2/2}\right| \leqslant |p_1 - p_2|.$$

Действительно, пусть $p_1 \geqslant p_2.$ Тогда

$$\frac{p_1}{1+p_1/2} - \frac{p_1}{1+p_2/2} = \frac{p_1 - p_2}{(1+p_1/2)(1+p_2/2)} \leqslant p_1 - p_2,$$

так как $p_1,\ p_2 \geqslant 0$.

То, что расстояние между множествами $Q = \{x^r : q_r > \lambda\}$ и $\bar{Q}$ имеет конечную величину, проверяется тривиально, так как последовательность $q_i,\ i = 1, \ldots,$ оказывается монотонно возрастающей. Из-за монотонности последовательности отпадает и необходимость в специальном построении языка S по общей схеме. Языку S принадлежит любое слово длины t, начиная с длины r, $t \geqslant r$, где r определено условием

$$\frac{q_r + p_r/2}{1+p_r/2} > \lambda, \quad \frac{q_{r-1} + p_{r-1}/2}{1+p_{r-1}/2} \leqslant \lambda.$$

Положим $p_i/2 = t_i$. Тогда $t_{i+1} = t_i/(1+t_i)$, откуда следует, что

$t_k = t_0/(1+kt_0)$, $k = 0, 1, \ldots,$ т. е. $p_k = p\Big/\Big(1 + \frac{k}{2}p\Big)$, $k = 0, 1, \ldots$ Далее $\frac{1-p_r/2}{1+p_r/2} > \lambda$ или $\frac{1+(r-1)\,p/2}{1+(r+1)\,p/2} > \lambda$, т. е. $r > \frac{1+\lambda}{1-\lambda} - \frac{2}{p}$.

7.8.2. Проблема распознавания.

Проблема распознавания возникает во всякой научной теории, исследующей аспекты распознавания формально описываемых объектов. В общем случае она состоит в том, чтобы по некоторым сведениям о распознаваемом объекте с разумной степенью точности определить сам объект. При этом желательно знать наименьший возможный объем исходных сведений.

Примером решения задачи распознавания в теории автоматов является распознавание конечного ДА на основе конечного эксперимента. При этом, вообще говоря, сам автомат не восстанавливается, но возможно построить минимальный по числу состояний эквивалентный ему автомат. Ситуация в вероятностном случае оказывается более сложной. В этом случае можно распознать полное поведение конечного ВА на основе сведении о поведении, заданных на конечном начальном сегменте полугруппы X^*, но делается это путем построения эквивалентного и, в общем случае, конечномерного ЛА минимальной размерности.

С другой стороны, как мы уже неоднократно убеждались, использование ЛА для моделирования ВА и применение в этой связи

липейно-алгебрапческой методологии исследования естественно и эффективно. Результаты, приводимые ниже, обобщают некоторые из важнейших результатов по распознаванию конечных ДА и показывают, что они имеют скорее лииейпо-алгебрапческую, нежели просто комбинаторную природу.

Нестрого говоря, смысл доказываемых утверждений этого параграфа заключается в том, что наличие или отсутствие того или иного свойства данного конечного автомата определяется по начальному сегменту его функционирования.

Первая такая задача связана с проблемой нахождения базисной матрицы N конечного ВА. Эта задача, безусловпо, относится к классу задач распознавания. Знание базисной матрицы N позволяет выявить эквивалентные векторы состояний автомата или различных автоматов, а следовательно, и строить автомат, эквивалентный данному.

Определение 1. Состояния a_1 и a_2 *(векторы состояний* $\boldsymbol{\mu}_1$ **и** $\boldsymbol{\mu}_2$) ВА A *различимы,* если существует слово $p \in X^*$ такое, что

$$\boldsymbol{\mu}_1 A(p)\mathbf{n}_F \neq \boldsymbol{\mu}_2 A(p)\mathbf{n}_F. \qquad (1)$$

Определение 2. *Степенью различимости ВА* A называется наименьшее целое $\rho(A)$ такое, что если произвольные состояния a_1 и a_2 ($\boldsymbol{\mu}_1$ **и** $\boldsymbol{\mu}_2$) различимы, то они ρ-различимы, т. е. существует слово p, $|p| \leqslant \rho$, такое, что верно (1).

Из формулы (1) вытекает, что степень различимости ВА можно определить в терминах базисной матрицы N.

Замечание 1. Для того чтобы степень различимости ВА A была равна ρ, необходимо и достаточно, чтобы существовала базисная матрица N_A с номерами строк $p_1, \ldots, p_k$, удовлетворяющими условию $|p_i| \leqslant \rho$, $i = 1, \ldots, k$, причем ρ неуменьшаемо.

То же самое мы можем выразить формулой

$$\mathrm{Lin}\,\{A(p)\mathbf{n}_F,\ |p| \leqslant \rho\} = \mathrm{Lin}\,\{A(p)\mathbf{n}_F,\ p \in X^*\}.$$

Определение 3. *Степенью достижимости* ВА A с начальным вектором состояний $\boldsymbol{\mu}(e)$ называется наименьшее целое $\delta(A)$ такое, что для всех $p \in X^*$ $\boldsymbol{\mu}(e)A(p) \in \mathrm{Lin}\,\{\boldsymbol{\mu}(e)A(p),\ |p| \leqslant \delta\}$, т. е. такое, что

$$\mathrm{Lin}\,\{\boldsymbol{\mu}(e)A(p),\ |p| \leqslant \delta\} = \mathrm{Lin}\,\{\boldsymbol{\mu}(e)A(p),\ p \in X^*\}.$$

Теорема 1. *Пусть* $L = \langle L(x),\ x \in X^*,\ \mathbf{a},\ \mathbf{m}\rangle$ *— ЛА размерности n и счетномерные матрицы N и M равны*

$$N = (\mathbf{m}, \ldots, L(p)\,\mathbf{m}, \ldots), \quad M = \begin{pmatrix} \mathbf{a} \\ \cdot\ \cdot\ \cdot \\ \mathbf{a}L(p) \\ \cdot\ \cdot\ \cdot \end{pmatrix}.$$

Тогда 1) $\rho(L) \leqslant \operatorname{rg} N - 1$; 2) $\delta(L) \leqslant \operatorname{rg} M - 1$.

Доказательство. Сначала получим вспомогательное утверждение общего характера. Пусть E — линейное пространство, δ_x — система линейных операторов в E и L — подпространство E. Скажем, что подпространство L' есть δ-*расширение* L, если

$$L' = L + \sum_{x \in X} \delta_x(L), \tag{2}$$

где через $\delta_x L$ мы обозначили образ L в отображении δ_x и знаки + и Σ относятся к суммированию линейных пространств.

Лемма 1. *Пусть последовательность подпространств*

$$L_0 \subseteq L_1 \subseteq \ldots \tag{3}$$

линейного пространства E получена последовательными δ-расширениями подпространства L_0. Тогда из $L_i = L_{i+1}$ следует $L_i = L_{i+s}$, $s = 1, \ldots$

Доказательство следует непосредственно из (2).

Следствие 1. *Для всякой последовательности подпространств типа* (3) *имеет место альтернатива: либо для каждого номера i $\dim L_i < \dim L_{i+1}$, либо существует максимальное значение размерности* $\dim L_i = k$, *причем оно достигается не более чем за $k - 1$ расширений (иначе говоря, заведомо $\dim L_{k-1} = k$).*

Понятно, что если линейное пространство E конечномерно, то имеет место второе положение.

Обратимся теперь к доказательству теоремы. По существу, оно вытекает из следствия 1 и конечномерности линейного пространства E_L (или $\mathscr{E}_L$), ассоциированного с конечномерным ЛА L.

Положим $E = \mathscr{E}_L$. В качестве системы линейных операторов в $\mathscr{E}_L$ рассмотрим систему матриц перехода $A(x)$ $(x \in X)$. Если положить $L_0 = \operatorname{Lin}\{\mathbf{m}\}$, то подпространство $L_1 = \operatorname{Lin}\{A(p)\mathbf{m}, \ |p| \leqslant i\}$. Вследствие конечномерности $\mathscr{E}_L$ размерность подпространства L_i не может возрастать неограниченно. Таким образом, существует целое $r \leqslant \dim \mathscr{E}_L$ такое, что $\operatorname{Lin}\{A(p)\mathbf{m}, \ |p| \leqslant r\} = \mathscr{E}_r$.

Предельное значение $\dim L_r$ в данном случае можно указать, оно равно размерности пространства $\mathscr{E}_r$, которая, в свою очередь, равна рангу матрицы N. Первая часть теоремы доказана.

Вторая часть доказывается аналогично, с той разницей, что вместо линейного пространства $\mathscr{E}_L$ мы рассматриваем линейное пространство E_L, и система операторов δ_x задается системой матриц перехода $L(x)$. В этом случае

$$L_0 = \operatorname{Lin}\{\mathbf{a}\}, \ L_r = \operatorname{Lin}\{\mathbf{a}A(p), \ |p| \leqslant r\} = E_L, \dim L_r = \operatorname{rg} M.$$

Следствие 2. *Пусть A — конечномерный ЛА (или конечный ВА). Тогда* $\rho(A),\ \delta(A) \leqslant |\mathfrak{A}| - 1$.

Числа $\rho(L)$ и $\operatorname{rg} N$ являются характеристиками неинициального ЛА $L = \langle L(x),\ \mathbf{m},\ x \in X\rangle$, числа $\delta(L)$ и $\operatorname{rg} M$ характеризуют инициальный ЛА L с заданным начальным вектором состояний **a**.

Перейдем теперь к задаче распознавания словарной функции, определяемой конечномерным ЛА $L = \langle L(x),\ \mathbf{m},\ x \in X\rangle$ с заданным начальным вектором состояний **a**.

Определение 4. *Обобщенной ганкелевой матрицей* (далее просто *ганкелевой),* соответствующей словарной функции $\varphi(p)$, называется счетномерная матрица с лексикографической индексацией номеров строк и столбцов словами в алфавите X, такая, что $\operatorname{Han}(\varphi) = (h_{p_1 p_2}) = (\varphi_{p_1}(p_2))$.

Из определения ганкелевой матрицы $\operatorname{Han}(\varphi)$ видно, что строки ее суть упорядоченное перечисление всех состояний $\varphi_{p_1}(p)$ словарной функции $\varphi(p)$. В этом случае линейная оболочка множества всех строк ганкелевой матрицы совпадает с линейным пространством $E_\varphi = \operatorname{Lin}\{\varphi_p,\ p \in X^*\}$. Ранг счетномерной матрицы $\operatorname{Han}(\varphi)$, по определению равный максимальному порядку неособенной подматрицы в матрице $\operatorname{Han}(\varphi)$, равен размерности линейной оболочки строк матрицы $\operatorname{Han}(\varphi)$. Таким образом, верно

Замечание 2.

$$\dim L_{r-1} = \dim \operatorname{Lin}\{\varphi_p,\ |p| \leqslant r-1\} = \dim E_\varphi = \operatorname{rg} \operatorname{Han}(\varphi).$$

Теорема 2. *Пусть* $\operatorname{Han}(\varphi)$ *— ганкелева матрица, соответствующая словарной функции* $\varphi(p)$. *Если* $\operatorname{Han}(\varphi)$ *имеет конечный ранг, то существует базисное множество линейно независимых строк с номерами из множества* $\{p:\ |p| \leqslant r-1,\ p \in X^*\}$ *и базисное множество линейно независимых столбцов с номерами из множества* $\{p:\ |p| \leqslant r-1,\ p \in X^*\}$.

Доказательство аналогично доказательству теоремы 1. В линейном пространстве вектор-строк E_φ будем рассматривать системы линейных операторов правых p-вращений $D_p^{\mathrm{rt}}(\varphi) = \varphi_p^{\mathrm{rt}}$. Применим лемму 1 к последовательности линейных подпространств $L_0 = \operatorname{Lin}\{\varphi = \varphi_e^{\mathrm{rt}}\}$, $L_1 = \operatorname{Lin}\{\varphi_p^{\mathrm{rt}},\ |p| \leqslant 1\}, \ldots$. Поскольку, по предположению, линейное пространство E_φ имеет размерность r, то из следствия 1 получаем для некоторого номера k $\dim L_{k-1} = k = \dim E_\varphi = r,\ E_\varphi = L_{r-1}$. Базисное множество для строк $\operatorname{Han}\varphi$ может быть выбрано в множестве $\{\varphi_p:\ |p| \leqslant r-1\}$. Для базисного множества столбцов доказательство остается тем же самым, с заменой правых p-вращений левыми.

Пусть $p_1 = e$, p_2, ..., p_r — номера базисных строк в матрице $\mathbf{Han}(\varphi)$ и $q_1 = e$, q_2, ..., q_r — номера базисных столбцов, такие, что $|p_i|$, $|q_i| \leqslant r-1$, $i = 1$, ..., r. Матрица $F = (\varphi(p_i q_j))$ порядка r, стоящая на пересечении линейно независимых строк и столбцов матрицы $\mathbf{Han}(\varphi)$ — неособенная. Обозначим через $F(p)$ матрицу $(\varphi(p_i p q_j))$.

Теорема 3. *Для произвольных слов p, q*

$$F(p, q) = F(p)F^{-1}F(q).$$

Доказательство. Рассмотрим $2r \times 2r$-матрицу

$$\begin{pmatrix} F & F(q) \\ F(p) & F(pq) \end{pmatrix}.$$

Ее ранг равен r. В самом деле, ее строки состоят из элементов матрицы $\mathbf{Han}(\varphi)$, расположенных в строках с номерами p_1, p_2, ..., p_r, $p_1 p$, $p_2 p$, ..., $p_r p$ и столбцах с номерами q_1, q_2, ..., q_r, qq_1, qq_2, ..., qq_r. Следовательно, в этой матрице нижние r строк суть линейные комбинации верхних r строк, которые линейно независимы, так как содержат неособенную подматрицу F. Но матрица

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

в случае, если ее подматрица A неособенная, имеет ранг r тогда и только тогда, когда $D = CA^{-1}B$.

Следствие 3. *Пусть $p = x_1$, ..., x_s — произвольное слово, тогда* $F(p) = F(x_1)F^{-1} \ldots F^{-1}F(x_s)$.

Теорема 4. *Пусть словарная функция $\varphi(p)$ представлена в ЛА L размерности n. Тогда задание словарной функции $\varphi(p)$ на сегменте длины $2n-1$ однозначно определяет ее всюду на свободной полугруппе X^*.*

Если ранг матрицы $\mathbf{Han}(\varphi)$ равен r, то словарная функция $\varphi(p)$ однозначно определяется заданием ее на сегменте длины $2r-1$.

Существует способ построения ЛА, представляющего словарную функцию $\varphi(p)$ минимальной размерности, равной $\mathbf{rg\,Han}(\varphi)$.

Доказательство. Поскольку пустое слово e является номером первой базисной строки и первого базисного столбца матрицы $F(p)$, то в левом верхнем углу находится число $\varphi(p)$. Рассмотрим r-мерный ЛА $L = \langle F(x)F^{-1}, \mathbf{m}, x \in X \rangle$. Пусть начальный вектор состояний $\mathbf{a} = (1, 0, \ldots, 0)$, а решающий поствектор $\mathbf{m}$ равен первому столбцу матрицы F: $\mathbf{m}^T = (\varphi(p_1), \ldots, \varphi(p_r))$. Для слов $p = x_1, \ldots, x_s$ получаем $F(x_1)F^{-1} \ldots F^{-1}F(x_s)F^{-1} = F(p)F^{-1}$. Поэтому $\mathbf{a}F(p)F^{-1}\mathbf{b} = \varphi(p)$. Таким образом, построенный линейный автомат (ЛА) представляет словарную функцию $\varphi(p)$. Заметим, что матрицы F и $F(x)$ содержат значения функции $\varphi(p)$ на словах длины, не

большей $2r-1$. Следовательно, возможно эффективное вычисление этих подматриц счетномерной матрицы Han (φ) конечного ранга *r*.

С другой стороны, допустим, что нам удалось построить другой линейный автомат размерности *r*, представляющий словарную функцию $\varphi'(p)$, совпадающую с $\varphi(p)$ на сегменте длины $2r-1$. Ганкелева матрица словарной функции $\varphi'(p)$ должна иметь ранг не более *r*, поскольку существует линейный автомат размерности *r*, представляющий эту словарную функцию. Рассмотрим словарную функцию $\psi(p)=\varphi(p)-\varphi'(p)$. В соответствии с теоремой 2 все базисные строки и столбцы могут быть определены значениями словарной функции $\psi(p)$ на сегменте длины $2r-1$. Поскольку на этом сегменте функция всюду равна нулю, то ранг матрицы $\mathrm{Han}\,(\psi)$ равен нулю и словарная функция $\psi(p)$ тождественно равна нулю. Продолжение словарной функции $\varphi(p)$ определяется заданием на сегменте длины $2r-1$ единственным образом. Минимальность размерности автомата *L* как представляющего словарную функцию φ также ясна из соображений ранга — для любого ЛА размерности меньше *r* ранг ганкелевой матрицы оказался бы меньше *r*.

Подход к задаче распознавания словарной функции, изложенный в лемме 1 и теоремах 1—4, определяет метод с широкой сферой применений.

Следующая задача хронологически является первой задачей распознавания, рассмотренной для конечных ВА. Речь пойдет об распознавания функций конечных однородных цепей Маркова. Рассматриваемая далее задача распознавания функций конечных однородных цепей Маркова по результатам и методологии их получения тяготеет к уже доказанным теоремам по распознаванию свойств ЛА и ВА. Следующее определение показывает, что задача распознавания последовательностей случайных кодов представляет собой задачу распознавания словарных функций из некоторого специального класса.

Определение 5. Пусть *X* — конечный алфавит. *Последовательностью случайных кодов* с множеством значепий *X* называется словарная функция $\varphi(p)\colon X^*\to[0,\ 1]$, удовлетворяющая условиям

$$1)\qquad \varphi(e)=1;$$

$$2)\qquad \sum_{x\in X}\varphi(px)=\varphi(p). \tag{4}$$

Этим определением охватываются все конечнозначные случайные процессы с дискретным положительным временем $t=0,1,\ldots$ Последовательность случайных кодов, как она определена в (4), есть,

по существу, частный случай многотактного канала — последовательность случайных кодов может быть отождествлена с каналом, входной алфавит которого состоит из одной буквы, т. е. автономным последовательностным источником случайных кодов.

Конечная однородная цепь Маркова является весьма частным, но наиболее важным и исследованным случаем последовательности случайных кодов. Дадим определение функции конечной однородной цепи Маркова. Рассматриваются только однородные конечные цепи Маркова, поэтому слова «однородный» и «конечный», как правило, впредь опускаются.

Пусть дана цепь Маркова с n состояниями

$$A = \langle \mathfrak{A},\ A,\ \mu \rangle, \tag{5}$$

где $\mathfrak{A} = \{a_1,\ \ldots,\ a_n\}$— множество состояний, $\mu(e) = (\mu_1,\ \ldots,\ \mu_n)$— начальное распределение вероятностей состояний (или вектор состояний), $A = (p_{ij})$ — стохастическая матрица переходных вероятностей цепи. Пусть Σ — некоторое разбиение множества $\mathfrak{A}$ на систему блоков $\pi_1,\ \ldots,\ \pi_k$, т. е.

$$\Sigma = \{\pi_1,\ \ldots,\ \pi_k\},\quad \bigcup_i \pi_i = \mathfrak{A},\quad \pi_i \cap \pi_j = \varnothing,\quad i \neq j.$$

Введем в рассмотрение словарную функцию $\mu(p)\colon\ \Sigma^* \to [0,\ 1]$, определенную на свободной полугруппе Σ^* следующим образом. Если слово $p = \pi_{i_1} \ldots \pi_{i_s}$, то положим

$$\mu(p) = \sum_{i_k \in \pi_k} \mu_{i_1} p_{i_1 i_2} \ldots p_{i_{s-1} i_s}, \tag{6}$$

где суммирование производится по всем таким траекториям цепи Маркова $a_{i_1} \ldots a_{i_s}$, для которых $i_k \in \pi_k,\ k = 1,\ \ldots,\ s$.

Определение 6. Последовательность случайных кодов с множеством значений $\Sigma = \{\pi_1,\ \ldots,\ \pi_k\}$, являющаяся словарной функцией $\mu(p)$, определенной по цепи Маркова (5) в соответствии с формулой (6), называется *функцией цепи Маркова* (5). Для функции цепи Маркова (5), определенной разбиением Σ, применяется обозначение $\Phi = \langle \Sigma,\ \mathfrak{A},\ A,\ \mu \rangle$.

Так как последовательность случайных кодов (6) однозначно задается совокупностью $\langle \Sigma,\ \mathfrak{A},\ A,\ \mu \rangle$, то этот список тоже будем называть *функцией цепи Маркова.* Для функции цепи Маркова можно предложить другое определение, независимое от определения цепи Маркова, но приведенное определение удобнее для нас при решении задачи распознавания.

Если B — $n \times n$-матрица, обозначим через $B_{\pi_i \pi_j}$ подматрицу,

расположенную на пересечении всех строк с номерами из $\pi_i \subseteq \mathfrak{A}$ и всех столбцов с номерами из $\pi_j \subseteq \mathfrak{A}$. В частном случае вектор-строки или вектор-столбца будем применять аналогичные обозначения.

Лемма 2. *Пусть* $p = \pi_1 \ldots \pi_s$ $(p \in \Sigma^*)$, *тогда*

$$\mu(p) = \boldsymbol{\mu}_{\pi_1} A_{\pi_1\pi_2} A_{\pi_2\pi_3} \ldots A_{\pi_{s-1}\pi_s} \mathbf{e}_{\pi_s}.$$

Доказательство предоставляется читателю в качестве упражнения.

Обозначим через E_{π_i} $n \times n$-матрицу, r, l-й элемент которой равен единице, если $r = l$ и $a_r \subset \pi_i$, и равен нулю в противном случае. Видно, что

$$\sum_i E_{\pi_i} = E$$

и, кроме того, E_{π_i}, $i = 1, \ldots, k$, — матрицы-идемпотенты. Обозначим через $\mathcal{J}_{\pi_i}$ матрицу размерности $|\pi_i| \times n$, получающуюся из E_{π_i} вычеркиванием всех нулевых строк, и через $\overline{\mathcal{J}}_{\pi_i}$ — матрицу размерности $n \times |\pi_i|$, получающуюся из E_{π_i} вычеркиванием нулевых столбцов. Тогда верны следующие утверждения:

1) $\overline{\mathcal{J}}_{\pi_i} = \mathcal{J}_{\pi_i}^T$, $\mathcal{J}_{\pi_i}^T \mathcal{J}_{\pi_i} = E_{\pi_i}$, $i = 1, \ldots, k$;

2) для любых $n \times n$-матрицы B, вектор-столбца $\mathbf{m}$ и вектор-строки $\mathbf{a}$ имеем

$$B_{\pi_i\pi_j} = \mathcal{J}_{\pi_i} B \mathcal{J}_{\pi_j}^T,$$

$$\mathbf{a}_{\pi_i} = \mathbf{a}\mathcal{J}_{\pi_i}^T, \quad \mathbf{m}_{\pi_i} = \mathcal{J}_{\pi_i}\mathbf{m}, \quad i = 1, \ldots, k.$$

используя введенные обозначения и лемму 2, получим следующее представление для словарной функции $\mu(p)$, определяющей функцию цепи Маркова:

$$\mu(p) = \boldsymbol{\mu}\mathcal{J}_{\pi_1}^T \mathcal{J}_{\pi_1} A \mathcal{J}_{\pi_2}^T \mathcal{J}_{\pi_2} A \mathcal{J}_{\pi_3}^T \ldots \mathcal{J}_{\pi_{s-1}} A \mathcal{J}_{\pi_s}^T \mathcal{J}_{\pi_s}\mathbf{e} =$$
$$= \boldsymbol{\mu} E_{\pi_1} A E_{\pi_2} A \ldots A E_{\pi_s}\mathbf{e}. \quad (7)$$

Введем в рассмотрение систему $n \times n$-матриц $\Phi(\pi/x) = AE_\pi$. Система $\Phi = \langle \{x\}, \mathfrak{A}, \Sigma, \Phi(\pi/x) \rangle$ определяет автономный ВА с n состояниями и выходным множеством Σ. Обозначим через $\boldsymbol{\mu}_\pi(e)$ распределение вероятностей $\boldsymbol{\mu}E_\pi$. В соответствии с формулой (7) видим, что для каждого слова $\pi p \in \Sigma^*$

$$\mu(\pi p) = \boldsymbol{\mu}_\pi(e)\tau_\Phi(p/x^{|p|}). \quad (8)$$

Замечание 3. Для каждой функции цепи Маркова с n состояниями существует автономный ВА с n состояниями и выходным множеством, совпадающим с множеством значений функции цепи, такой, что

многотактный канал, представляемый ВА, и словарная функция $\mu(p)$, определяющая цепь Маркова, связаны соотношением (8).

Мы пришли к интерпретации функции цепи Маркова как частного случая конечно-автоматного канала. Было бы неточно говорить, что функция цепи Маркова есть конечно-автоматный канал из-за временного несоответствия в функционировании: первая выходная буква автоматного канала есть реакция на первую входную букву, тогда как в случае функции цепи Маркова первое значение последовательности случайных кодов есть ее начальное значение. Тем не менее замечание 3 гарантирует возможность применения общей методологии решения задачи распознавания для рациональных словарных функций к задаче распознавания функций цепей Маркова.

Теорема 5. *Пусть* $\Phi = \langle \Sigma, \mathfrak{A}, A, \mu \rangle$ *— функция цепи Маркова с n состояниями. Существует ЛА размерности* $\operatorname{rg} \mu$, *определяющий словарную функцию* $\mu(p)$, *который строится по множеству значений словарной функции* $\mu(p)$ *на начальном сегменте длины* $2 \operatorname{rg} \mu - 1$.

Доказательство аналогично доказательству теоремы 3.

Полученные результаты показывают, что если известны ранг словарной функции $\varphi(p)$ и множество ее значений на начальном сегменте длины не менее $2 \operatorname{rg} \varphi - 1$, то по этим сведениям эффективно строится единственное продолжение этого сегмента, т. е. восстанавливается словарная функция $\varphi(p)$. Если же задан только некоторый начальный сегмент словарной функции, но ранг ее неизвестен, то распознавание словарной функции, вообще говоря, невозможно, т. е. невозможно указать единственное продолжение начального сегмента словарной функции произвольной фиксированной длины. В такой ситуации мы получаем семейство словарных функций, совпадающих на заданном начальном сегменте.

Рассмотрим следующую задачу: пусть словарная функция $f(p)$ задана на начальном сегменте длины l. Требуется так доопределить ее на всех словах свободной полугруппы X^*, чтобы полученная словарная функция имела бы заданные свойства. Эта задача имеет и автоматную интерпретацию — можно считать, что речь идет о распознавании поведения конечномерного ЛА, когда неизвестна его размерность. Методология позволяет дать параметрическое описание класса возможных продолжений словарной функции для наперед заданного ранга. Рассмотрим задачу продолжения словарной функции с минимальным возможным рангом.

Существует взаимно однозначное соответствие между словарными функциями $f(p)$ и обобщенными ганкелевыми матрицами: сопоставим

функции $f(p)$ матрицу Han f, в строке p_1 и столбце p_2 которой стоит число $f(p_1p_2)$. Обратно, каждая обобщенная ганкелева матрица, строки и столбцы которой занумерованы словами свободной полугруппы X^*, задает единственную словарную функцию $f(p)$. Поэтому задачу минимального продолжения словарной функцииможе формулировать и как задачу такого доопределения обобщенной ганкелевой матрицы, чтобы ранг получившейся матрицы был наименьшим. В такой постановке возможны и иные, чисто матричные, толкования задачи доопределения. Для нас такая постановка предпочтительнее, поскольку она теснее связана с методами решения.

Если известные элементы матрицы принадлежат некоторому полю K, то удобно смотреть на неопределенные элементы как на многочлены над полем K от переменных из некоторого множества $\Lambda = (\lambda_i,\ i \in \mathcal{I})$. Обозначим через $K(\Lambda)$ кольцо многочленов над полем K. Все дальнейшие результаты параграфа сформулируем для произвольного поля K, так что интерпретация их для словарных функций требует замены поля K на поле действительных чисел R.

Пусть $A(\Lambda)$ — многочленная матрица над кольцом $K(\Lambda)$. Минор матрицы $A(\Lambda)$ называется *постоянным,* если его элементы не зависят от переменных $\lambda_i,\ i \in \mathcal{I}$.

Определение 7. *Рангом* $\operatorname{rg} A$ многочленной матрицы $A(\Lambda)$ называется максимальный среди порядков постоянпых ненулевых миноров этой матрицы. Если в случае счетномерной матрицы $A(\Lambda)$ такого максимально числа нет, то полагаем $\operatorname{rg} A = \infty$.

В случае постоянной матрицы это понятие совпадает с обычным определением ранга матрицы. Для произвольного фиксированного набора значений переменных $\Lambda_0 = \{\lambda_i^0,\ i \in \mathcal{I}\}$ верно неравенство $\operatorname{rg} A(\Lambda_0) \geqslant \operatorname{rg} A(\Lambda)$, которое, в частности, может быть строгим неравенством для произвольных наборов Λ_0 или тождественным равенством.

Определение 8. Квадратная многочленная матрица A надкольцом $K(\Lambda)$ называется *унимодулярной,* если $\det A = \text{const} \neq 0$.

Унимодулярная матрица имеет обратную, тоже унимодулярную, многочленную матрицу, вычисляемую по обычным правилам матричной алгебры. Можно сказать еще, что ранг многочленной матрицы равен максимальному среди порядков ее унимодулярных подматриц.

Пусть теперь словарная функция $f(p)$ задана на начальном сегменте длины l. Доопределим $f(p)$ на словах $p,\ |p| > l$, положив $f(p) = \lambda_p$,

где λ_p — переменная, и таким образом получим словарную функцию f: $X^* \to K(\Lambda_l) \subseteq K(\Lambda)$, где $\Lambda_l = \{\lambda_p\colon\ p \in X^*, |p| > l\}$, $\Lambda = \{\lambda_p\colon\ p \in X^*\}$. Можно говорить о ранге частично определенной словарной функции f, подразумевая под этим ранг соответствующей многочленной ганкелевой матрицы $\mathbf{Han}\, f$ над $K(\Lambda)$.

Теорема 6. *Пусть словарная функция f задана на начальном сегменте длины l, со значениями в поле K. Существует словарная функция g: $X^* \to K(I)$, где $I \subseteq \{\lambda_p\colon\ l < |p| \leqslant 2l+1\}$, которая при подстановке вместо $\lambda_p \in I$ произвольных чисел λ_p^0 обращается в словарную функцию g^0 такую, что*

1) $g^0(p) = f(p)$, $|p| \leqslant l$;
2) $\operatorname{rg} g^0 = \operatorname{rg} f$.

Любое продолжение f минимального ранга может быть получено подстановкой некоторых значений переменных из I в полином $g(p)$.

Доказательство. Введем бинарное отношение $\succ$ на $K(\Lambda)$: $a \succ b$ тогда и только тогда, когда либо $a \neq \mathrm{const}$, либо $a \equiv \mathrm{const}$ и $a = b$. Для вектор-строк (столбцов) $\mathbf{a}$, $\mathbf{b}$ и матриц A, B будем писать $\mathbf{a} \succ \mathbf{b}$, $A \succ B$ для обозначения того, что $\succ$ выполняется покомпонентно. Далее, скажем, что a есть *линейная $\succ$-комбинация* векторов $\mathbf{b}_1, \ldots, \mathbf{b}_k$, если существуют $c_1, \ldots, c_k$ из $K(\Lambda)$, для которых

$$\mathbf{a} \succ \sum_{i=1}^{k} c_i \mathbf{b}_i.$$

Будем просматривать строки $\mathrm{Han}\, f$ сверху вниз, и поступать так: если строка с номером p есть линейная $\succ$-комбинация предыдущих строк матрицы $\mathrm{Han}\, f$, то отбрасываем ее, если нет — вносим в список. Так получим последовательность строк с возрастающими номерами

$$e = p_1 < \ldots < p_n \tag{9}$$

такую, что:

а) ни одна из них не является $\succ$-комбинацией предыдущих строк матрицы $\mathbf{Han}\, f$,

б) любая строка p матрицы $\mathbf{Han}\, f$ есть $\succ$-комбинация строк с номерами $p_i \leqslant p$.

Подобно тому, как это делалось со строками, составим последовательность столбцов $\mathbf{Han}\, f$ с номерами

$$e = q_1 < \ldots < q_m \tag{10}$$

такую, что:

а) ни один из них не является $\succ$-комбинацией столбцов, стоящих левее,

б) любой столбец q может быть вычислен как $\succ$-комбинация столбцов с номерами $q_i \leqslant q$.

В дальнейшем слова p и q с нижними индексами обозначают номера из последовательностей (9) и (10) соответственно. Для завершения доказательства теоремы нам понадобятся следующие леммы.

Лемма 3. *Пусть для некоторого слова p*

$$f(pq_j) \succ \sum_{p_i \leqslant p} a_{p_i} f(p_i q_j), \quad j = 1, \ldots, m. \tag{11}$$

Тогда для каждого слова q

$$f(pq) \succ \sum_{p_i \leqslant p} a_{p_i} f(p_i q).$$

Доказательство. Пусть $f(qp) = \text{const}$. Тогда элементы $f(p_i q)$, $f(pq_j)$, $p_i \leqslant p$, $q_j \leqslant q$ — константы. Пусть q-столбец есть $\succ$-комбинация столбцов из (10) с коэффициентами $b_{q_1}, \ldots, b_{q_m}$. Тогда, в частности,

$$f(pq) = \sum_{q_j \leqslant q} b_{q_j} f(pq_j).$$

Подставим в эту формулу выражения для $f(pq_j)$ из (11) и переменим порядок суммирования

$$f(pq) = \sum_{q_j \leqslant q} b_{q_j} \sum_{p_i \leqslant p} a_{p_i} f(p_i q_j) =$$

$$= \sum_{p_i \leqslant p} a_{p_i} \sum_{q_j \leqslant q} b_{q_j} f(p_i q_j) = \sum_{p_i \leqslant p} a_{p_i} f(p_i q).$$

Из леммы 3 следует, что $n \leqslant m$. Следующая лемма двойственна к предыдущей.

Лемма 4. *Пусть для некоторого слова q*

$$f(p_i q) \succ \sum_{q_j \leqslant q} b_{q_j} f(p_i q_j), \quad i = 1, \ldots, n.$$

Тогда для каждого слова

$$p\ f(pq) \succ \sum_{q_j \leqslant q} b_{q_j} f(pq_j).$$

Итак, $m \leqslant n$ и, следовательно, $m = n$.

На самом деле в $\succ$-комбинации для p-й строки условие $p_i \leqslant p$ можно опустить. Более точно, пусть p-я. строка есть $\succ$-комбинация строк с номерами из (9) с коэффициентами $a_{p_1}, \ldots, a_{p_n}$, тогда если $p_s > p$, то $a_{p_s} \equiv 0$. Действительно, предположим обратное. Пусть p_s-наибольший такой номер, что $p_s > 0$ и $^* a_{p_s} \not\equiv 0$. Рассмотрим соотношения

$$f(pq_j) \vdash \sum_{p_i \neq p_s} a_{p_i} f(p_i q_j) + a_{p_s} f(p_s q_j), \quad j = 1, \ldots, n,$$

для тех случаев, когда $f(p_s q_j) = \text{const}$. Тогда $f(pq_j) = \text{const}$, так как $p_s > p$, и эти соотношения являются равенствами. В эти равенства вместо многочленов a_{p_i} можно подставить такие их значения $a^0_{p_1}, \ldots, a^0_{p_s}$, что $a^0_{p_s} \neq 0$, и разрешить эти равенства относительно $f(p_s q_j)$. По лемме 3 получим, что p_s-я строка есть $\vdash$-комбинация строк, лежащих выше ее в $\text{Han } f$, что противоречит построению (9).

Доказанное утверждение пригодится нам в следующей специальной форме. Буквой F обозначим $n \times n$-матрицу $(f(p_i q_j))$, через $F(p)$ — матрицу $(f(p_i pq_j))$. Пусть $p = rt$, тогда $F(p)$ — подматрица матрицы $\text{Han } f$, расположенная на пересечении строк с номерами $p_1 r, \ldots, p_n r$ и столбцов с номерами $tq_1, \ldots, tq_n$. В левом верхнем углу $F(p)$ находится элемент $f(p)$.

Лемма 5. *Если* $A(x) = (a_{p_i p_j}(x))$ *— такая матрица над* $K[\Lambda]$, *что* $F(x) \vdash A(x)F$, *то для* $p_i > p_i x$ $a_{p_i p_j}(x) \equiv 0$, *т. е.* $A(x)$ *имеет «близкий к левому треугольному» вид.*

Для доказательства достаточно применить доказанное выше утверждение к строкам с номерами $p_i x$, $i = 1, \ldots, n$.

Следствие 4. *В условиях леммы* 5 *для всех* q

$$F(xq) \vdash A(x)F(q). \tag{12}$$

Доказательство непосредственно следует из леммы 3, если считать $F(q)$ расположенной в строках с номерами $p_1, \ldots, p_n$ и столбцах с номерами $qq_1, \ldots, qq_n$, а $F(xq)$ — расположенной в строках с номерами $p_1 x, \ldots, p_n x$ и тех же столбцах.

Соотношение (12) означает, что всякий постоянный элемент $f(p_i xqq_j)$ матрицы $F(xq)$ можно вычислить, умножив p-ю строку $A(x)$ на q-й столбец

$$F(q)\colon\ f(p_i xqq_j) = \sum_{k=1}^{n} a_{p_i p_k}(x) f(p_k qq_j).$$

Если при этом $f(p_k qq_j) \not\equiv \text{const}$, то необходимо $p_k > p_i x$, тогда по лемме 5 $a_{p_i p_k} \equiv 0$.

Отсюда получаем

Следствие 5. *В условиях леммы* 5, *если* $F(q) \vdash G$, *то* $F(xq) \vdash A(x)G$.

Лемма 6. *Пусть для каждого* $x \in X$ *найдена матрица* $A(x)$ *такая, что* $F(x) \vdash A(x)F$. *Тогда для произвольного слова* $p \in X^*$ *выполняется* $F(p) \vdash A(p)F$.

Доказательство. Действительно, по следствию 5 если $F(p) \succ A(p)F$ для всех слов p длины k, то для произвольного $x \in X$ $F(xp) \succ A(x)A(p)F = A(xp)F$.

Лемма 7. $\det F \equiv \text{const} \neq 0$.

Доказательство. p-я строка в $\text{Han}\, f$ тогда и только тогда является $\succ$-комбинацией строк с номерами $p_i \leqslant p$, когда вектор $(f(pq))_{q \in \omega}$ $(\omega = \{q: |pq| \leqslant l\})$ является линейной комбинацией векторов $(f(p_i q))_{q \in \omega}$, $p_i \leqslant p$. Линейная зависимость с коэффициентами из поля $K' \supseteq K$ векторов с координатами из K равносильна их линейной зависимости с коэффициентами из K. Поэтому, в случае, когда $\det F$ неприводим над K, получим в подходящем поле $K' \supset K$ противоречивое равенство $\det F' = 0$, $F \succ F'$. Следовательно, $\det F \equiv \text{const} \neq 0$, так как по лемме 3 ни одна из строк F не является $\succ$-комбинацией других строк.

Теперь можно завершить доказательство теоремы 6.

Матрица $A(x) = F(x)F^{-1}$ удовлетворяет соотношению $F(x) \succ \succ A(x)F$. Утверждение леммы 6 для таких $A(x)$ приводит к формуле $\bar{F}(p) \succ \bar{F}(x_1)F^{-1} \ldots F^{-1}F(x_k) = \bar{F}(p)$ для $p = x_1 \ldots x_k$ или, что равносильно $\bar{F}(pq) \succ \bar{F}(p)F^{-1}\bar{F}(q)$ для всех p, q. Совокупность $\mathbf{a} = (1, 0, \ldots, 0)$, $A(x) = \bar{F}(x)F^{-1}$, $x \in X$, $\mathbf{b} = (f(p_1), \ldots, f(p_n))^T$, где $\mathbf{b}$ — первый столбец F, определяет ЛА над $K[I]$, где I — множество переменных, входящих в F и $F(x)$. Этот автомат определяет отображение $g: X^* \to K[I]$, $g(p) = \mathbf{a}A(p)\mathbf{b} = (1, 0, \ldots, 0) \times \times \bar{F}(p)(1, 0, \ldots, 0)^T (\bar{F}(e) = F)$. Очевидно, $f(p) \succ g(p)$ для всех p. Подставляя в F, $F(x)$ всевозможные значения вместо переменных множества I, получим некоторое семейство продолжений функции f над полем K ранга n. Или, говоря на матричном языке, получим семейство продолжений обобщенной ганкелевой матрицы наименьшего возможного ранга n. В каждом из таких продолжений строки с номерами $p_1, \ldots, p_n$ линейно независимы, а прочие — их линейные комбинации.

Для завершения доказательства теоремы осталось убедиться в том, что произвольное продолжение f функции f ранга n содержится в этом семействе. Соответствующие этому продолжению матрицы $\text{Han}\, \hat{f}$, $\hat{F}$, $\hat{F}(p)$ получены подстановкой соответствующих значений вместо переменных λ_p в матрицах $\text{Han}\, f$, F, $F(p)$.

Покажем, что $\hat{F}(pq) = \hat{F}(p)\hat{F}^{-1}\hat{F}(q)$ для всех p, q. В самом деле, если для некоторых p, q, i, j

$$\hat{f}(p_i pqq_j) \neq (\hat{f}(p_i pq_1), \ldots, \hat{f}(p_i pq_n))\,\hat{F}^{-1}\begin{pmatrix}\hat{f}(p_1 qq_j)\\ \cdots\\ \hat{f}(p_n qq_j)\end{pmatrix},$$

то определитель порядка $n + 1$

$$\begin{vmatrix} & \hat{F} & \hat{f}(p_1 qq_j)\\ & & \cdots\\ & & \hat{f}(p_n qq_j)\\ \hat{f}(p_i pq_1) & \ldots\ \hat{f}(p_i pq_n) & \hat{f}(p_i pqq_j)\end{vmatrix} \neq 0.$$

Этот определитель, однако, либо имеет две одинаковые строки, если $p_i p \in \{p_1, \ldots, p_n\}$, либо два одинаковых столбца, если $qq_j \in \{q_1, \ldots, q_n\}$, либо является минором $(n+1)$-го порядка матрицы $\mathrm{Han}\, f$ ранга n, т. е. в любом случае должен быть равен пулю. Требуемое равенство доказано. Но из него следует, что автомат построенного выше вида с $\hat{F}$, $\hat{F}(x)$ и $\hat{\mathbf{b}} = \mathbf{b}$ вместо F, $F(x)$ и $\mathbf{b}$ реализует $\hat{f}$.

Из леммы 7 вытекает, что $\mathrm{rg}\,\mathrm{Han}\, f \geqslant n$. Теперь, когда доказано существование продолжения $\mathrm{Han}\, f$ над K ранга n, мы можем утверждать, что ранг мпогочлепной матрицы $\mathrm{Han}\, f$ в смысле определения ранга, данного во введении, в точности равен n и, таким образом, F — максимальный постоянный ненулевой минор $\mathrm{Han}\, f$.

7.8.3. Проблема устойчивости

Задача устойчивости ВА, как и проблема редукции, возникает естественным образом, если задаться вопросом практического использования этой математической модели. Моделирование физического распознаваемого объекта посредством конечного ВА связано с предварительным статистическим анализом поведения моделируемого распознаваемого объекта и статистическими оценками элементов переходных матриц. Полученные статистические оценки задают значения вероятностей перехода в математической модели распознаваемого объекта с некоторой погрешностью. В этих условиях закопомерен вопрос: насколько чувствительно предельное поведение математической модели распознаваемого объекта к небольшим изменениям этих вероятностей перехода? Обладает ли ВА свойством «устойчивости» в том смысле, что малым флуктуациям элементов матриц перехода соответствуют небольшие изменения в предельном

поведении математической модели распознаваемого объекта? Оказывается ВА не всегда обладает свойством «устойчивости». Тем не менее могут быть описаны некоторые классы устойчивых ВА. Перейдем к строгой математической постановке задачи.

Пусть копсчпый ВА $A = \langle A(x),\ x \in X \rangle$ с начальным вектором состояний $\boldsymbol{\mu}(e)$ и решающим поствектором $\mathbf{n}_F$ представляет язык $T(A, \lambda)$ изолированной точкой сечения $\lambda \in [0, 1)$.

Определение 1. Конечный ВА A называется *устойчивым,* если существует такое положительное ε, что для всякого конечного ВАA', удовлетворяющего условиям:

$$1)\ |\mu(e) - \mu'(e)| < \varepsilon, \quad 2)\ |A(x) - A'(x)| < \varepsilon,$$

λ является изолированной точкой сечения и языки $T(A, \lambda)$ и $T(A', \lambda)$ совпадают.

Определение 2. Конечный ВА A называется *актуальным,* если все элементы матриц $A(x)$ строго положительны.

Определение 3. Язык S называется *дефинитным,* если для некоторого целого k справедливо следующее: если $|p| \geqslant k$, то $p \in S$ тогда и только тогда, когда $p = p_1 p_2$, где $|p_2| = k$, $p_2 \in S$.

Докажем теорему, представляющую как самостоятельный интерес, так и имеющую значение для решения проблемы устойчивости.

Теорема 1. *Если A — актуальный ВА и λ — изолированная точка сечения, то язык $T(A, \lambda)$ — дефинитный.*

Доказательство. Пусть $|\chi(p) - \lambda| \geqslant \delta > 0$. Предположим, что все элементы матриц A (x) больше $\Delta > 0$. Предположим, что ВА A имеет одно финальное состояние a_n и фиксировано начальное состояние a_1. (Доказательство в общем случае по существу остается тем же.)

Пусть число k таково, что $(1 - 2\Delta)^{k-1} < 2\delta$. Для любого слова $p = x_1 \ldots x_k$ матрица $A(p) = A(x_1) \ldots A(x_k)$ и, таким образом, согласно известному следствию удовлетворяет неравенству

$$\|A(p)\| \leqslant (1 - 2\Delta)^{k-1} < 2\delta.$$

Так как $\chi(p)$ есть в данном случае (1, n)-элемент матрицы $A(p)$, то согласно известному следствию

$$|\chi(qp) - \chi(p)| \leqslant |A(q)A(p)| \leqslant \|A(p)\|.$$

Таким образом, для слов p, $|p| = k$, имеет место неравенство $|\chi(qp) - \chi(p)| < 2\delta$. Следовательно, $qp \in T(A, \lambda)$ тогда и только тогда, когда $p \in T(A, \lambda)$, а это доказывает, что — $T(A, \lambda)$ дефинитный язык.

Классом ВА, обладающим свойством устойчивости, оказывается класс актуальных автоматов.

Теорема 2. *Пусть* $A = \langle A(x),\ x \in X\rangle$ *— актуальный автомат и* λ *— изолированная точка сечения. Существует* $\varepsilon > 0$ *такое, что для каждого ВА, удовлетворяющего условиям* 1) *и* 2) *определения* 1, *языки* $T(A, \lambda)$ *и* $T(A', \lambda)$ *совпадают.*

Доказательство. Пусть $A(x)$ и $A'(x)$ переходные матрицы автоматов A и A' соответственно, Δ и Δ' — соответственно наименьшие элементы матриц $A(x)$ и $A'(x)$. Покажем, что для любого $\delta_1 > 0$ можно найти такое $\varepsilon > 0$, что из системы неравенств

$$|A(x) - A'(x)| < \varepsilon \qquad (1)$$

следует система неравенств для $p = x_1 \dots x_m$:

$$|A(p) - A'(p)| = |A(x_1) \dots A(x_m) - A'(x_1) \dots A'(x_m)| < \delta. \qquad (2)$$

Последнее доказывает теорему в силу изолированности точки сечения λ.

Пусть число k таково, что $(1 - 2\Delta)^{k-1} < \delta_1/3$. Мы можем выбрать достаточно малое $\varepsilon > 0$ так, чтобы из (1) следовало:

а) $(1 - 2\Delta')^{k-1} < \delta_1/3$; этого возможно добиться, поскольку Δ есть предельное значение Δ' при ε, стремящемся к нулю;

б) для всех слов p_2 таких, что $|p_2| \leqslant k$, имеет место соотношение

$$|A(p_2) - A'(p_2)| < \delta_1/3.$$

Если $|p| \leqslant k$, то соотношение (2) тривиально выполняется в силу б). Если $|p| > k$, то $p = p_1 p_2$, где $|p_2| = k$. Матрица $A(p_2)$ есть произведение k матриц типа $A(x)$, поэтому ввиду известного следствия

$$\|A(p_2)\| \leqslant (1 - 2\Delta)^{k-1} < \delta_1/3.$$

Аналогично, применяя а), имеем $A'(p_2) < \delta_1/3$. Поэтому

$$|A(p) - A'(p)| \leqslant$$
$$\leqslant |A(p_1)A(p_2) - A(p_2)| + |A'(p_1)A'(p_2) - A'(p_2)| + |A(p_2) - A'(p_2)|.$$

Ввиду известного следствия и условия б) каждое слагаемое в правой части неравенства меньше $\delta_1/3$.

Замечание 1. Теорема 1 неверна для произвольного конечного ВА с изолированной точкой сечения. Действительно, пусть A — ВА с изолированной точкой сечения, представляющий недефинитный язык $T(A, \lambda)$. Автомат A' может быть актуальным и удовлетворять условиям 1) и 2) определения устойчивости. В этом случае либо λ не является изолированной точкой сечения для A', либо в силу теоремы 1 язык $T(A', \lambda)$ является дефинитным, следовательно, $T(A, \lambda) \neq T(A', \lambda)$.

Тем не менее класс устойчивых ВА оказывается существенно шире класса актуальных автоматов.

Теорема 3. *Всякий конечный регулярный ВА с изолированной точкой сечения является устойчивым.*

Доказательство. Так как λ является изолированной точкой сечения автомата *A,* то для всех *p* выполняется неравенство

$$|\chi(p) - \lambda| \geqslant \delta, \quad \delta > 0.$$

Поскольку *A* является регулярным ВА, то согласно известнной теореме существует натуральное *l* такое, что каждая матрица из системы матриц $\Xi = \{\bar{A}(p), \ |p| = l\}$ является стягивающей. Система матриц Ξ — конечная, содержит не более k^l элементов, где *k* — число элементов множества *X*. Обозначим через *c* минимальное *c(p)* по всем матрицам $A(p) \in \Xi$. Тогда для каждого слова *p* $(|p| = N)$ верно соотношение $\|\bar{A}(p)\| \leqslant (1-c)^{[N/l]}$.

Пусть число *N* таково, что $(1-c)^{[N/l]} < \delta/3|F|$. При соответствующем выборе ε система матриц вероятностей переходов $\{A'(x), \ x \in X\}$ любого автомата *A',* удовлетворяющего условиям 1) и 2) определения устойчивости, будет подчиняться эргодическому принципу. Кроме того, можно выбрать достаточно малое ε > 0 так, чтобы

$$1) \qquad (1-c')^{[N/l]} < \delta/3\,|F|, \quad \text{где} \quad c' = \min_{|p|=l} c'(p) \qquad (3)$$

(это можно сделать, так как $\lim\limits_{\varepsilon \to 0} c' = c$),

2) для всех p_1 таких, что $|p_1| \leqslant N$, имело место соотношение

$$|\mu(e)A(p_2) - \mu'(e)A'(p_2)| \leqslant \delta/3|F|.$$

Если для любого слова *p*

$$|\mu(e)A(p) - \mu'(e)A'(p)| < \delta/|F|, \qquad (4)$$

то $|\chi(p) - \chi'(p)| < \delta$, а это означает, что если $p \in T(A, \lambda)$, то $p \in T(A', \lambda)$, и наоборот, если $p \notin T(A, \lambda)$, то $p \notin T(A', \lambda)$. Покажем, что имеет место неравенство (4). Действительно, если $|p| \leqslant N$, то (4) выполняется в силу 2), если $|p| > N$, то $p = p_1 p_2$, где $|p_2| = N$. Матрица $A(p_2)$ есть произведение *N* матриц $A(x)$. Очевидно, имеет место неравенство

$$|\mu A(p) - \mu' A'(p)| \leqslant |\mu A(p_1)A(p_2) - \mu A(p_2)| + |\mu' A'(p_1)A'(p_2) - \mu' A'(p_2)| + |\mu A(p_2) - \mu' A'(p_2)|,$$

$$\mu = \mu(e), \quad \mu' = \mu'(e).$$

Каждое из слагаемых в правой части последнего неравенства меньше $\delta/3|F|$. Таким образом, верно (4), и тем самым теорема доказана.

Теорема 4. *Пусть ВА A с изолированной точкой сечения* λ $(|\chi(p) - \lambda| \geqslant \delta, \ \delta > 0)$ *и матрицы вероятностей переходов*

удовлетворяют условию $\|A(p)\| < 2\delta/|F|$ *для всех слов* p $(|p| = m)$, *для некоторого натурального числа* m. *Тогда* $T(A, \lambda)$ *есть дефинитный язык.*

Доказательство. Для любых p, $|p| \geqslant m$, $p = p_1p_2$, $|p_2| = m$ выполняется неравенство

$$|\chi(p) - \chi(p_2)| \leqslant |F|\|\mu A(p_1)A(p_2) - \mu A(p_2)\|,$$

откуда получим

$$|\chi(p) - \chi(p_2)| \leqslant |F|\|A(p_2)\| < 2\delta. \qquad (5)$$

Из (5) следует, что $p \in T(A, \lambda)$ тогда и только тогда, когда

$$p_2 \in T(A, \lambda).$$

Следствие 1. *Если ВА* A *с изолированной точкой сечения* λ *является эргодическим, то он с точкой сечения* λ *представляет дефинитный язык.*

Из рассмотрения теоремы 4 может возникнуть предположение, что свойство устойчивости ВА с изолированной точкой сечения является характеристикой дефинитного языка. Однако это не так. Покажем, что не всякий ВА, представляющий с изолированной точкой сечения λ дефинитный язык, является устойчивым.

Например, ВА A с матрицами переходов вида

$$A(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha(x) & \beta(x) \\ 0 & \alpha(x) & \beta(x) \end{pmatrix}, \qquad 0 < \alpha(x) < 1, \qquad x \in X,$$

$$\mu(e) = (\mu_1, \mu_2, \mu_3), \qquad n_F = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix},$$

при $\mu_1 > \beta(\tilde{x})$ для некоторого $\tilde{x}$ с точкой сечения $\lambda = \beta(\tilde{x})$ представляет дефинитный язык X^*. Действительно, $\chi(px) = \mu_1 + \beta(x) \times (\mu_2 + \mu_3)$. В этом случае точка сечения $\beta(\tilde{x})$ автомата A является изолированной. Но та же самая точка сечения $\beta(\tilde{x})$ уже не является изолированной по отношению к автомату A', отличающемуся от автомата A лишь матрицей

$$A'(\tilde{x}) = \begin{pmatrix} 1-\varepsilon & \varepsilon & 0 \\ 0 & \alpha(\tilde{x}) & \beta(\tilde{x}) \\ 0 & \alpha(\tilde{x}) & \beta(\tilde{x}) \end{pmatrix}, \qquad \varepsilon > 0.$$

В самом деле, матрица $A'(\tilde{x})$ является регулярной стохастической, причем

$$\lim_{n \to \infty} (A'(\tilde{x}))^n = \begin{pmatrix} 0 & \alpha(\tilde{x}) & \beta(\tilde{x}) \\ 0 & \alpha(\tilde{x}) & \beta(\tilde{x}) \\ 0 & \alpha(\tilde{x}) & \beta(\tilde{x}) \end{pmatrix}.$$

Следовательно, для любого $\varepsilon_1 > 0$ найдется k такое, что для всех слов $p = \tilde{x}\tilde{x}\ldots\tilde{x}$ $(|p| > k)$ имеет место неравенство $|\chi'(p) - \beta(\tilde{x})| < \varepsilon_1$.

Точка сечения $\beta(\tilde{x})$ не является изолированной относительно автомата A'. Более того, в этом случае $T(A, \beta(\tilde{x})) \neq T(A', \beta(\tilde{x}))$. Действительно, матрица $(A'(\tilde{x}))^k$, $k \geqslant 1$, имеет вид

$$(A'(\tilde{x}))^k = \begin{pmatrix} (1-\varepsilon)^k & m(k) & \beta(\tilde{x}) - (1-\varepsilon)^{k-1}\beta(\tilde{x}) \\ 0 & \alpha(\tilde{x}) & \beta(\tilde{x}) \\ 0 & \alpha(\tilde{x}) & \beta(\tilde{x}) \end{pmatrix},$$

где $m(k) = 1 - \beta(\tilde{x}) + (1-\varepsilon)^{k-1}\beta(\tilde{x}) - (1-\varepsilon)^k$. Для $x \neq \tilde{x}$ имеем

$$(A(\tilde{x}))^k A(x) = \begin{pmatrix} (1-\varepsilon)^k & \alpha(x)\left[1-(1-\varepsilon)^k\right] & \beta(x)\left[1-(1-\varepsilon)^k\right] \\ 0 & \alpha(x) & \beta(x) \\ 0 & \alpha(x) & \beta(x) \end{pmatrix},$$

так что

$$\chi'(\underbrace{\tilde{x}\ldots\tilde{x}x}_{k+1}) = \mu_1(1-\varepsilon)^k + \mu_1\beta(x)\left[1-(1-\varepsilon)^k\right] + \beta(x)(\mu_2+\mu_3).$$

Если $\beta(x) < \beta(\tilde{x})$, то существует t такое, что для всех $k \geqslant t$ правая часть последнего соотношения меньше $\beta(\tilde{x})$, а это как раз доказывает, что $T(A, \beta(\tilde{x})) \neq T(A', \beta(\tilde{x}))$.

На практике чаще всего случается рассматривать реакции автомата не на всей полугруппе X^*, а только по отношению к определенному типу последовательностей входных сигналов. Это позволяет нам подойти к проблеме устойчивости с другой стороны, а именно, рассмотреть задачу частичной устойчивости ВА.

Определение 4. Точка сечения λ, $0 \leqslant \lambda < 1$, называется *изолированной* точкой сечения ВА A *относительно языка* S, если существует $\delta > 0$ такое, что для любого слова $p \in S$ выполняется неравенство $|\chi(p) - \lambda| \geqslant \delta$.

Если λ — изолированная точка сечения ВА A, то она является изолированной точкой сечения и относительно любого языка S. Вообще, если λ — изолированная точка сечения ВА A относительно языка S, то она является изолированной точкой сечения относительно любого языка $S' \subseteq S$.

Определение 5. ВА A с изолированной относительно языка S точкой сечения λ *устойчив относительно языка* S, если существует $\varepsilon > 0$ такое, что для всякого автомата A', удовлетворяющего следующим условиям:

1) $|\mu(e) - \mu'(e)| < \varepsilon$; 2) $|A(x) - A'(x)| < \varepsilon$,

λ является изолированной точкой сечения относительно языка S и языки $T(A, \lambda) \cap S$ и $T(A', \lambda) \cap S$ совпадают.

Ясно, что если ВА A с изолированной относительно языка S точкой сечения λ устойчив относительно языка S, то он устойчив и относительно любого языка $S' \subseteq S$.

Теорема 5. *Если ВА A является эргодическим и λ, $0 \leqslant \lambda < 1$, есть изолированная точка сечения относительно языка S, то автомат A устойчив относительно языка S.*

Доказательство аналогично доказательству теоремы 3.

Система матриц $\{A(x),\ x \in X\}$ вероятностей переходов ВА A, не являясь эргодической, может, однако, обладать тем свойством, что ее замыкание относительно операции умножения матриц содержит непустое множество регулярных стохастических языков. Например, матрицы типа

$$\begin{pmatrix} * & * & 0 \\ * & * & 0 \\ 0 & 0 & * \end{pmatrix}, \quad \begin{pmatrix} * & * & * \\ 0 & * & 0 \\ 0 & 0 & * \end{pmatrix},$$

где звездочкой обозначены ненулевые элементы матриц, не являются регулярными. Однако их произведение

$$\begin{pmatrix} * & * & * \\ * & * & * \\ 0 & 0 & * \end{pmatrix}$$

определяет регулярную матрицу.

Теорема 6. *Пусть $A = \langle A(x),\ x \in X \rangle$ — BA, и $R \subseteq X^*$ — язык такой, что для всех слов $p \in R$ и только для них матрицы $A(p)$ представляют собой регулярные стохастические матрицы. Тогда язык R является регулярным.*

Доказательство. Обозначим через $\{B(x),\ x \in X\}$ систему булевских матриц, соответствующих матрицам $A(x)$ вероятностей переходов ВА A. Присоединив к системе $\{B(x),\ x \in X\}$ подходящую единичную матрицу E, которую для удобства обозначим через $B(\Lambda)$, получим систему $B^{(1)} = \{B(x),\ x \in X \cup \Lambda\}$. Пусть $B^{(k)},\ k = 1, 2, \ldots$ есть система матриц $B^{(k)} = \{B(p),\ |p| = k,\ p \in (X \cup \Lambda)^*\}$. Имеет место вложение

$$B^{(1)} \subseteq \ldots \subseteq B^{(k)} \subseteq \ldots \tag{6}$$

Так как число булевских матриц фиксированного порядка конечно, то наступит такой момент, когда $B^{(l)} = B^{(l+1)}$. В этом случае $B^{(l)} = B^{(l+r)},\ r = 1, 2, \ldots$ Действительно, поскольку, очевидно, $B^{(k+1)} = B^{(1)}B^{(k)}$, то по индукции

$$B^{(l+r)} = B^{(1)}B^{(l+r-1)} = B^{(l+r-1)} = \ldots = B^{(l)}.$$

Таким образом, существует наименьшее число $l = l(B^{(1)})$ такое, что $B^{(k)} = B^{(l)}$ для всех $k > l$, тогда как для всех $B^{(k)} \subset B^{(k+1)}$ $k < l$. Пусть $B^{(l)}$ представляет собой систему булевских матриц

$$B^{(l)} = \{B_1, \ldots, B_m\}. \qquad (7)$$

Система матриц $B^{(l)}$ содержит булевские шаблоны всех матриц $A(p)$. Согласно (6) в систему (7), в частности, войдут все матрицы системы $\{B(x),\ x \in X \cup \Lambda\}$.

Рассмотрим инициальный конечный ДА *B,* состояниями которого являются матрицы системы (7), входными буквами — буквы алфавита *X* и начальным состоянием — матрица $B(\Lambda)$. Функцию переходов автомата *B* определим следующим образом: $\delta(B_i, x) = B_i B(x)$. Видно, что функция переходов однозначно определена для любой пары (B_i, x). В качестве множества финальных состояний автомата *F* выберем совокупность всех регулярных булевских матриц системы $B^{(l)}$. ДА *B* множеством состояний *F* представляет регулярный язык *R*. Действительно, пусть $p \in R$, т. е. матрица *A(p)* является регулярной стохастической матрицей. После подачи слова *p* автомат *B* в соответствии с определением его функции переходов $\delta(B_i, x)$ перейдет из начального состояния $B(\Lambda)$ в состояние *B(p).* Так как по условию *A(p)*—регулярная стохастическая матрица, то соответствующая ей булевская матрица *B(p)* будет регулярной булевской матрицей, иначе говоря, $p \in T(B)$.

Обратно, пусть $p \in T(B)$, т. е. $B(p) \in F$. Но *F* есть совокупность регулярных булевских матриц. Следовательно, *B(p)* есть регулярная булевская матрица. Отсюда стохастическая матрица *A(p),* которой соответствует булевская матрица *B(p),* является регулярной, т. е. $p \in R$.

Определение 6. Множество булевских матриц *C* будем называть *эргодическим,* если его замыкание $[C]$ относительно умножения булевских матриц принадлежит множеству регулярных булевских матриц *B.*

Так как $B^{(l)}$ охватывает все типы матриц, которые содержатся в замыкании системы стохастических матриц $\{A(x),\ x \in X\}$, и *F* есть совокупность всех регулярных булевских матриц системы $B^{(l)}$, то для всякого эргодического подмножества $C \subseteq F$ замыкание $[C]$ принадлежит множеству *F*.

Естественным образом определяются максимальные эргодические подмножества. Всякое *максимальное эргодическое подмножество* $D \subseteq F$ представляет собой множество, замкнутое относительно операции умножения булевских матриц. В случае, если ВА *A*

является эргодическим, имеется максимальное эргодическое множество, совпадающее с множеством $B^{(l)} \setminus \{B(\Lambda)\}$.

Пусть D — произвольное максимальное эргодическое подмножество множества F автомата B, и M_D — язык, представленный инициальным ДА B множеством состояний D. Пусть, далее, N_D— произвольное конечное подмножество M_D. Для языка $(N_D)^k$ положим $l_k(N_D) = \max\limits_{p \in (N_D)^k} |p|$. Обозначим далее через $N_D(U, V, k)$ язык, определяемый выражением

$$N_D(U, V, k) = U[(N_D)^k \cup (N_D)^{k+1} \cup \ldots] \cup V, \qquad (8)$$

где V — некоторый конечный язык, а U — произвольный язык.

Теорема 7. *Пусть для некоторого натурального k точка сечения* λ ВА A *является изолированной точкой сечения относительно языка* $N_D(U, V, k)$. *Тогда существует такое натуральное* $m \geqslant k$, *что ВА* A *с точкой сечения* λ *устойчив относительно языка* $N_D(U, V, m)$.

Доказательство. По условию теоремы существует $\delta > 0$ такое, что для любого слова $p \in N_D(U, V, k)$ имеет место неравенство $|\chi(p) - \lambda| \geqslant \delta$.

Поскольку для любого $t \geqslant k N_D(U, V, t) \subseteq N_D(U, V, k)$, то точка сечения λ, ВА A будет изолированной относительно языка $N_D(U, V, t)$, причем $|\chi(p) - \lambda| \geqslant \delta$ для всех $p \in N_D(U, V, t)$. Обозначим через $\Xi_{N_D} = \{A_1, \ldots, A_t\}$ систему стохастических матриц, соответствующих словам из N_D, т. е. $A(p) \in \Xi_{N_D}$, $p \in N_D$.

Система матриц Ξ_{N_D} является эргодической. Таким образом, для нее $\lim \| A_{i_1} \ldots A_{i_N} \| = 0$, $A_{i_j} \in \Xi_{N_D}$, $j = 1, \ldots, N$.

Пусть натуральное $m \geq k$ таково, что $ml(N_D) \geqslant l(N)$, $l(S) = \max |p|$ для $S \subseteq X^*$, S — конечный язык и $\| A_{i_1} \ldots A_{i_m} \| < \delta/3|F|$, $A_{i_j} \in \Xi_{N_D}$, $j = 1, \ldots, m$.

Можно выбрать достаточно малое $\varepsilon > 0$ так, чтобы для любого автомата A', удовлетворяющего условиям 1), 2) определения относительной устойчивости, имели место неравенства

а) $\| A'_{i_1} \ldots A'_{i_m} \| < \delta/3|F|$,

где A'_{i_j}, $j = 1, \ldots, m$ — стохастические матрицы, принадлежащие системе матриц Ξ'_{N_D}, соответствующих словам из N_D для ВА A' (аналогично доказательству теоремы 3);

б) $$|\mu A(p_2) - \mu' A(p_2)| < \delta/3|F|$$

для всех q таких, что $|p_2| \leqslant ml(N_D)$. Если для любого $p \in N_D(U, V, m)$

$$|\mu A(p) - \mu' A(p)| < \delta/|F|, \qquad (9)$$

то для тех же p $|\chi(p) - \chi'(p)| < \delta$, а это означает, что если $\chi(p) > \lambda$, то $\chi'(p) > \lambda$, и наоборот, если $\chi(p) < \lambda$ то $\chi'(p) < \lambda$.

Покажем,что имеет место неравенство (9).

Если $p \in V$, то $|p| \leqslant l(V) \leqslant ml(N_D)$ и неравенство (9) следует из б). Если же $p \in U\{(N_D)^m \cup (N_D)^{m+1} \cup \ldots\}$, то $p = p_1 p_2$, где $p_2 \in (N_D)^m$. Матрица $A(p_2)$ есть произведение m матриц из Ξ_{N_D}, и поэтому

$$\|A(p_2)\| < \delta/3|F|.$$

Далее, согласно а)

$$\|A'(p_2)\| < \delta/3|F|.$$

Но

$$|\mu A(p) - \mu' A'(p)| \leqslant |\mu A(p_1)A(p_2) - \mu A(p_2)| +$$

$$+ |\mu' A'(p_1)A'(p_2) - \mu' A'(p_2)| + |\mu A(p_2) - \mu' A'(p_2)|.$$

Поэтому в силу известного следствия и условия б), поскольку $|p_2| \leqslant$ $\leqslant ml(N_D)$, каждое слагаемое в правой части неравенства меньше $\delta/3|F|$. Таким образом, $|\mu A(p) - \mu' A'(p)| < \delta/|F|$.

Если ВА с изолированной относительно языков $W_1, \ldots, W_k$ точкой сечения λ устойчив относительно каждого из них, то он, очевидно, устойчив и относительно языка

$$\bigcup_{i=1}^{k} W_i.$$

Лемма 1. *Пусть ВА A с изолированной относительно языков $N_{D_i}(U_i, V_i, k_i)$, $i = 1, 2, \ldots, k$, точкой сечения λ устойчив относительно каждого из них. Тогда существует такая система натуральных чисел $m_i \geqslant k_i$, $i = 1, \ldots, k$, что ВА A с точкой сечения λ устойчив относительно языка*

$$\bigcup_{i=1}^{k} N_{D_i}(U_i, V_i, m_i).$$

Доказательство. Точка сечения λ будет изолированной относительно каждого языка $N_{D_i}(U_i, V_i, m_i)$. Согласно теореме 7 автомат A с точкой сечения λ устойчив относительно каждого из них, а, следовательно, он устойчив и относительно языка $\bigcup_{i=1}^{k} N_{D_i}(U_i, V_i, m_i)$.

7.8.4. Представимость последовательностей пар случайных кодов

Перейдем к изучению зависимости между ВА и последовательностями случайных кодов. На вход ВА поступает последовавтельпость случайных символов и на выходе получается также некоторая последовательность случайных символов. Две последовательности случайных кодов оказываются коррелированными вследствие того, что каждая пара выборочных траекторий этих последовательностей связана ВА. Таким образом, последовательности случайных кодов, которые мы будем изучать, принимают парные значения, отвечающие соответственно входному и выходному символам автомата. В соответствии с определением 5 п. 7.8.2 *последовательностью пар случайных кодов* будем называть словарную функцию $\mu\colon (X \times Y)^* \to [0,\ 1]$, удовлетворяющую следующим условиям:

1) $$\mu(e,\ e) = 1;$$

2) $$\sum_{x \in X, y \in Y} \mu(px, qy) = \mu(p, q).$$

Будем применять для последовательности пар случайных кодов также обозначение

$$J = \langle \mu(p,\ q),\ (p,\ q) \in (X \times Y)^* \rangle. \qquad (1)$$

Пусть дана последовательность пар случайных кодов (1) над алфавитом $X \times Y$.

Определение 1. Последовательность случайных кодов J_X задаваемая словарной функцией

$$\mu(p) = \sum_{|q|=|p|} \mu(p, q),$$

называется *левой последовательностью случайныхкодов для J.*

Аналогично определяется правая последовательность случайных кодов для *J*.

Определение 2. Язык $d^{lt}(J) = \{p\colon \mu(p) \neq 0\}$ называется *левым детерминатором* последовательности случайных кодов *J*.

Аналогично определяется *правый детерминатор.*

Замечание 1. Детерминатор $d^{lt}(J)$, содержащий слово *p,* содержит и все начала этого слова. Множество несократимых справа слов $p'x$ дополнения детерминатора $d^{lt}(J)$ определяется условием $p'x \notin d^{lt}(J) \to p' \in d^{lt}(J)$.

Детерминатор может быть конечным. Условимся, что детерминатор не может быть пустым: в том случае, если все вероятности $\mu(p)$ равны нулю, он все же содержит пустое слово, на котором $\mu(e) = 1$. Однако для краткости будем называть *пустым* детерминатор, содержащий только пустое слово.

Определение 3. Многотактный канал

$$\tau(J) = \langle \mu(q/p),\ (p, q) \in (X \times Y)^* \rangle,$$

определенный условиями

$$\mu(q/p) = \begin{cases} 0, & \text{если } |p| \neq |q|, \\ \mu(p, q)/\mu(p), & \text{если } p \in d^{lt}(J) \text{ и } |p| = |q|, \\ \text{произвольное условное} \\ \text{вероятностное распределение, если } p \notin d^{lt}(J), \end{cases}$$

называется *прямым многотактным каналом, ассоциированным с J.*

Пусть дан ВА $A = \langle X, Y, \mathfrak{A}, \mu(a', y/a, x) \rangle$.

Определение 4. Последовательность пар случайных кодов (1) *представлена в ВА A* (начальным вектором состояний $\mu(e)$), если на левом детерминаторе $d^{lt}(J)$ прямой многотактный канал $\tau(J)$ совпадает с автоматным каналом $\tau_A(q/p)$, представленным в ВА *A*.

Замечание 2. Из определений следует, что последовательность (1) с пустым левым детерминатором представима в любом ВА.

Теорема 1. *Для того чтобы последовательность пар случайных кодов* $J = \langle \mu(p, q),\ (p, q) \in (X \times Y)^* \rangle$ была *представима в ВА, необходимо и достаточно, чтобы для любых пар слов* (p, q) *и* (p', q') *из* $(X \times Y)^*$, *таких, что* $p'p \in d^{lt}(J)$, *отношение* $\mu(q'q/p'p)/\mu(q'/p') = \mu_{p',q'}(q/p)$ *было условным вероятностным распределением при фиксированных значениях слов* p' *и* q'.

Доказательство. Необходимость доказывается так же, как в известной теореме. Докажем достаточность. Покажем, что для последовательности *J* можно построить многотактный канал, удовлетворяющий условиям теоремы. Положим

$$\tau(q/p) = \begin{cases} 0, & \text{если } |p| \neq |q|, \\ \mu(q/p), & \text{если } |p| = |q|, \quad p \in d^{lt}(J), \\ \tilde{\mu}(q/p), & \text{если } |p| = |q|, \quad p \notin d^{lt}(J), \end{cases}$$

где $\tilde{\mu}(q/p)$ — произвольное условное распределение вероятностей, удовлетворяющее условиям автоматности. Докажем, что многотактный канал, определяемый $\tau(q/p)$, автоматен. Условие 1) известной теоремы выполнено по определению. Пусть $\tau(q/p) = 0$ при $|p| = |q|$. Если $\mu(p) = 0$, то из автоматности $\tilde{\mu}(q/p)$ следует, что $\tau(qq_1/pp_1) = 0$. Пусть $\mu(p) \neq 0$. Тогда $\mu(q/p) = \mu(p, q)/\mu(p)$ и

$\mu(p,' q) = 0$. Но тогда для любых пар слов $|p_1| = |q_1|$ получаем $\mu(pp_1/qq_1) = 0$. Если $\mu(pp_1) \neq 0$, то $\tau(qq_1/pp_1) = \mu(pp_1/qq_1)/\mu(pp_1) = 0$. Если же $\mu(pp_1) = 0$, то $\tau(qq_1/pp_1) = \tilde{\mu}(qq_1/pp_1)$. Из условия автоматности канала $\tilde{\mu}(q/p)$ имеем $\tilde{\mu}(q/p) = \mu(q/p) = 0$, следовательно, $\tilde{\mu}(qq_1/pp_1) = 0$ или $\tau(qq_1/pp_1) = 0$. Наконец, пусть $\tau(q/p) \neq 0$ и $\mu(pp_1) \neq 0$. Тогда $\mu(p) \neq 0$ и $\tau(qq_1/pp_1)/\tau(q/p) = \mu(qq_1/pp_1)/\mu(q/p)$ — условное вероятностное распределение в соответствии с условием теоремы 1. Аналогичное утверждение следует при $\mu(p) = 0$ из определения канала $\tau(J)$, а при $\mu(p) \neq 0$ и $\mu(pp_1) = 0$ — из определения канала $\tilde{\mu}$ и условия теоремы 1.

Если построить теперь ВА A в соответствии с алгоритмом синтеза известной теоремы, то он представляет последовательность пар случайных кодов J. Действительно, $\tau_A^{\mu(e)}(q/p) = \tau(q/p)$ для пар слов $|p| = |q|$. Для того чтобы $\mu(q/p)$ было определено, необходимо и достаточно, чтобы выполнялось условие $\mu(p) \neq 0$. Но если это условие выполнено, то $\mu(q/p) = \tau(q/p) = \tau_A^{\mu(e)}(q/p)$ для всех пар слов $|p| = |q|$.

Следствие 1. *Для того чтобы последовательность пар случайных кодов* (1) *была представима в ВА, необходимо и достаточно, чтобы распределения вероятностей* $\mu(p, q)$ *удовлетворяли условию*

$$\sum_{|q'|=|p'|} \mu(qq'/pp') = \mu(q/p), \quad pp' \in d^{it}(J).$$

При получении условий представимости последовательностей пар случайных кодов в конечных ВА применим тот же метод, который использовался для решения аналогичной задачи представимости многотактных каналов и словарных функций. Введем в рассмотрение множество состояний автоматной последовательности случайных кодов.

Пусть $J = \langle \mu(p, q), (p, q) \in (X \times Y)^* \rangle$ — автоматная последовательность пар случайных кодов. Пусть для пары слов одинаковой длины $|p_1| = |q_1|$ $\mu(p_1, q_1) \neq 0$. Введем обозначение

$$\frac{\mu(p_1 p, q_1 q)}{\mu(p_1, q_1)} = \mu_{p_1, q_1}(p, q).$$

Определение 5. Последовательность пар случайных кодов

$$J_{p_1, q_1} = \langle \mu_{p_1, q_1}(p, q), \quad (p, q) \in (X \times Y)^* \rangle$$

называется *существенным состоянием* последовательности J.

Множеством состояний автоматной последовательности случайных кодов J называется множество его существенных состояний.

Замечание 3. Существенные состояния автоматной последовательности случайных кодов J, левой последовательности случайных кодов J_X для J и прямого многотактного канала, ассоциированного с J, связаны между собой соотношением

$$\mu_{p_1,q_1}(p,q) = \mu_{p_1}(p)\cdot\tau_{p_1,q_1}(q/p). \qquad (2)$$

Действительно, пусть $\mu(p_1,\ q_1) \neq 0$. Поскольку $\mu(p_1,\ q_1) = \mu(p_1)\tau(q_1/p_1)$, то $\mu(p_1) \neq 0$ и $\tau(q_1/p_1) \neq 0$. Поэтому $\frac{\mu(p_1p,\ q_1q)}{\mu(p_1,\ q_1)} = \frac{\mu(p_1p)}{\mu(p_1)}\cdot\frac{\mu(q_1q/p_1p)}{\mu(q_1/p_1)}$, так что выполняется (2).

Если обозначить через $n(J)$, $n(J_X)$ и $n(\tau(J))$ соответственно мощности множеств состояний J, J_X и $\tau(J)$, то из формулы (2) вытекает неравенство $n(J) \leqslant n(J_X)n(\tau(J))$.

Из замечания 3 видно, что множество состояний J конечно тогда и только тогда, когда конечны множества состояний J_X и $\tau(J)$. Но из определения представимости 4 вытекает, что конечность множества состояний J не является необходимым условием представимости J в конечном ВА.

Замечание 4. Для того чтобы последовательность пар случайных кодов J была представима в конечном ВА, необходимо и достаточно, чтобы существовал прямой многотактный канал $\tau(J)$, ассоциированный с J, представимый в конечном ВА.

Представляет интерес описание последовательности пар случайных кодов с конечным числом состояний в терминах корреляционных зависимостей случайных кодов, образующих последовательность. Пусть $J = \langle \mu(p,\ q),\ (p,\ q) \in (X \times Y)^* \rangle$ — последовательность пар случайных кодов.

Теорема 2. *Для того чтобы автоматная последовательность J имела конечное число состояний, необходимо и достаточно, чтобы имела конечное число состояний левая последовательность J_X для J и существовали:*

1) *конечная система условных вероятностных распределений*

$$\tau_a(y/x),\ a \in \mathcal{I} = \{1, \ldots, N\};$$

2) *конечнозначная целочисленная словарная функция* $a(p,\ q)$: $(X \times Y)^* \to \mathcal{I}$, *удовлетворяющая требованию*

$$a(p_1,\ q_1) = a(p_2,\ q_2) \to a(p_1x,\ q_1x) = a(p_2x,\ q_2y), \qquad (3)$$

такие, что для произвольной пары слов одинаковой длины $(p,\ q)$, $p \in d^{lt}(J)$, *условная вероятность для пары* $(px,\ qy)$ *определяется формулой*

$$\mu(qy/px) = \mu(q/p)\tau_{a(p,\ q)}(y/x). \qquad (4)$$

Доказательство. Ввиду замечания 3 «доказательство сводится к рассмотрению прямого многотактного канала $\tau(q/p) = \mu(p, q)/\mu(p)$, $\mu(p) \neq 0$, ассоциированного с последовательностью J.

Необходимость., Поскольку многотактный канал $\tau(J)$ представим в конечном ВА, возможно доопределить его до конечно-автоматного канала на всей полугруппе X^*. Сохраним за доопределенным каналом то же обозначение $\tau(q/p)$. Канал τ имеет конечное число состояний, т. е. множество $\mathscr{L}_\tau = \{\tau_{p_1, q_1}(q/p), (p_1, q_1) \in (X \times Y)^*\}$ является конечным множеством. Следовательно, обозначая одним индексом из множества $\mathscr{I} = \{1, \ldots, N\}$ все каналы τ_{p_1, q_1} в множестве $\mathscr{L}_\tau$, совпадающие между собой, получим некоторую целочисленную конечнозначную словарную функцию $a(p, q)$: $(X \times Y)^* \to \mathscr{I}$ такую, что

$$\tau_{p_1, q_1}(q/p) = \tau_{a(p_1, q_1)}(q/p).$$

Пусть каналы τ_{p_1, q_1} и τ_{p_2, q_2} совпадают. Тогда по известной лемме совпадают каналы $\tau_{p_1 x, q_1 y}$ и $\tau_{p_2 x, q_2 y}$. Таким образом, для функции $a(p, q)$ выполнено требование (3). Из определения состояния канала $\tau_{p, q}$

$$\tau_{p, q}(y/x) = \tau(qy/px)/\tau(q/p), \ \tau(q/p) \neq 0,$$

получим соотношение

$$\tau(qy/px) = \tau(q/p)\tau_{p, q}(y/x),$$

которое верно и тогда, когда $\tau(q/p) = 0$. В этом случае канал $\tau_{p, q}$ не определен. Доопределим его, положив равным одному из каналов множества $\mathscr{L}_\tau$ таким образом, чтобы сохранялось условие (3). Следовательно, соотношение (4) можно переписать в виде

$$\tau(qy/px) = \tau(q/p)\tau_{a(p, q)}(y/x),$$

где словарная функция $a(p, q)$ удовлетворяет всем условиям теоремы.

Достаточность. Построим полудетерминированный конечный ВА A с множеством состояний $\mathscr{L}_A$ и условными вероятностями $\mu_A(a', y/a, x)$, определенными следующим образом:

$$\mu_A(a', y/a, x) = \delta(a', y/a, x)\tau_a(y/x).$$

Здесь $\delta(a', y/a, x)$ — функция переходов полудетерминированного автомата такая, что если для пары слов (p, q) $a = a(p, q)$, $a_1 = a(px, qy)$, то

$$\delta(a', y/a, x) = \begin{cases} 1, & \text{если } a' = a_1, \\ 0 & \text{в противном случае.} \end{cases}$$

Определение функции δ корректно, поскольку из-за свойства (3) значение $a_1 = a(px, qy)$ функции a не зависит от выбора конкретной пары (p, q).

В качестве начального состояния ВА A рассмотрим состояние $a_0 = a(e, e)$, т. е. $\tau_{a_0} = \tau$. Покажем, что ВА A представляет многотактный канал τ. Действительно, если $p = x_1 \dots x_s$, $q = y_1 \dots y_s$, то

$$\tau_A(q/p) = \sum_{a_1, \dots, a_s} \mu_A(a_1, y_1/a_0, x_1) \dots \mu_A(a_s, y_s/a_{s-1}, x_s) =$$

$$= \tau(y_1/x_1)\, \tau_{x_1 y_1}(y_2/x_2) \dots \tau_{x_1 \dots x_{s-1}, y_1 \dots y_{s-1}}(y_s/x_s) = \tau(q/p).$$

Следствие 2. *Для того чтобы автоматная последовательность J имела конечное число состояний, необходимо и достаточно, чтобы конечное число состояний имела левая последовательность J_x для J и существовал полудетерминированный конечный автомат $D = \langle X, Y, \mathfrak{A}, \delta(a', y/a, x)\rangle$ такой, что условная вероятность для любой пары слов одинаковой длины (p, q) имела вид*

$$\mu(y_1 \dots y_s/x_1 \dots x_s) = \tau_{a_0}(y_1/x_1) \dots \tau_{a_{s-1}}(y_s/x_s),$$

где последовательность индексов $a_0, \dots, a_{s-1}$ есть последовательность состояний автомата D, соответствующая входно-выходной последовательности $(x_1 \dots x_s, y_1 \dots y_s)$.

Доказательство следует из формулы 4 и свойств функции $a(p, q)$.

Следствие 3. *Для того чтобы последовательность случайных кодов $J = \langle \mu(p), p \in X^* \rangle$ имела конечное число состояний, необходимо и достаточно, чтобы существовал конечный* ДА $A = \langle X, \mathfrak{A}, \delta(a, x)\rangle$ *такой, чтобы вероятность любого начала p этой последовательности вычислялась в форме*

$$\mu(x_1 \dots x_s) = \mu_{a_0}(x_1) \dots \mu_{a_{s-1}}(x_s),$$

где последовательность индексов $a_0, \dots, a_{s-1}$ есть последовательность состояний автомата A для входного слова.

Доказательство вытекает из того факта, что в данном случае полудетерминированный автомат из следствия 2, вырождается в обычный ДА.

Важным случаем представимости последовательностей пар случайных кодов в ВЛ является случай, когда представляющий автомат — детерминированный. Если последовательность J представлена в ДА A, то будем говорить, что автомат A *преобразует* левую «входную» последовательность случайных кодов J_x в правую «выходную» последовательность случайных кодов J_y.

Для получения критерия детерминированно-автоматной преобразуемости одной последовательности случайных кодов в другую применим общую теорию, развитую в начале параграфа. Докажем предварительно лемму.

Лемма 1. *Для того чтобы многотактный канал* $\tau(q/p)$ *был представим в ДА, необходимо и достаточно, чтобы он имел вид*

$$\tau(q/p) = \begin{cases} 1, & \text{если } q = \varphi(p), \\ 0 & \text{в противном случае,} \end{cases}$$

где φ: $X^* \to Y^*$ *— некоторое автоматное отображение. При этом существует взаимно однозначное соответствие между множествами состояний автоматного канала* $\tau(q/p)$ *и автоматного отображения* $q = \varphi(p)$.

Для доказательства следует воспользоваться выражением условного вероятностного распределения для ДА A в форме

$$\mu_A(a', y/a, x) = \begin{cases} 1, & \text{если } a' = \delta_A(a, x), \quad y = \lambda_A(a, x), \\ 0 & \text{в противном случае,} \end{cases}$$

и определением многотактного канала $\tau_A(q/p)$, представленного в ДА A в форме

$$\tau_A(y_1 \dots y_s/x_1 \dots x_s) = \sum_{a_0, a_1, \dots, a_{s-1}} \mu(a_0)\,\mu(a_1, y_1/a_0, x_1) \dots \mu(a_s, y_s/a_{s-1}, x_s).$$

Пусть $J_1 = \langle \mu(p), \ p \in X^* \rangle$ и $J_2 = \langle \mu(q), \ q \in Y^* \rangle$ — последовательности случайных кодов.

Теорема 3. *Для того чтобы существовал ДА Мили, преобразующий последовательность случайных кодов* J_1 *в последавательность случайных кодов* J_2, *необходимо и достаточно, чтобы для любого слова q выполнялось соотношение*

$$\mu(q) = \sum_{q=\varphi(p)} \mu(p), \tag{5}$$

где φ: $X^* \to Y^*$ *— некоторое автоматное отображение.*

Доказательство. Пусть A — ДА, преобразующий J_1 в J_2. Фиксируем длину слова i и рассмотрим случайный код J_1^i, принимающий значения p, $|p| = i$, с вероятностями $\mu(p)$. случайный код J_1^i имплицирует случайный код J_2^i, принимающий значения q, $|q| = i$, с вероятностями $\mu(q)$, причем функцией импликации является автоматное отображение φ_A, производимое автоматом A. Поэтому в силу известной леммы получаем (5) для всех слов q фиксированной длины i. Поскольку длина i фиксировалась произвольным образом, (5) верно на всей полугруппе Y^*.

Обратно, пусть для последовательностей случайных кодов J_1 и J_2 выполнено соотношение (5), где φ — автоматное отображение. Построим ДА *A,* производящий автоматное отображение φ. Он преобразует входную последовательность J_1 в некоторую выходную последовательность случайных кодов J_2'. Вероятности $\mu'(q)$ последовательности J_2' будут определяться соотношением

$$\mu'(q) = \sum_{q=\varphi(p)} \mu(p),$$

откуда следует по определению последовательности случайных кодов, что J_2 и J_2' совпадают.

Замечание 5. Из леммы 1 и теоремы 3 вытекает, что последовательность J_1 преобразуется в последовательность J_2 конечным ДА тогда и только тогда, когда автоматное отображение φ в соотношении (5) имеет конечное число состояний .

Определение 6. Последовательность случайных кодов J_1 *(конечно) автоматно имплицирует* последовательность случайных кодов J_2, если существует (конечный) ДА, преобразующий J_1 в J_2. Последовательности J_1 и J_2 *(конечно) автоматно-эквивалентны,* если каждая из них (конечно) автоматно имплицирует другую.

Определение 7. ДА $A = \langle X, Y, \mathfrak{A}, \delta, \lambda \rangle$ называется *автомато-перестановкой* над парой языков $S \subseteq X^*$ и $Q \subseteq Y^*$, если его автоматное отображение φ_A реализует взаимно однозначное отображение языка *S* на язык *Q*.

Теорема 4. *Для того чтобы последовательности случайных кодов J_1 и J_2 были (конечно) автоматно-эквивалентны, необходимо и достаточно, чтобы существовал (конечный) автомат-перестановка над парой языков $d(J_1)$ и $d(J_2)$, преобразующий последовательность J_1 в последовательность J_2.*

Доказательство. Предположим, что последовательности J_1 и J_2 автоматно-эквивалентны. Тогда существуют автоматные отображения $\varphi: X^* \to Y^*$ и $\psi: Y^* \to X^*$, такие, что для пар слов (p, q) одинаковой длины

$$\mu(q) = \sum_{q=\varphi(p)} \mu(p) \text{ и } \mu(p) = \sum_{p=\psi(q)} \mu(q).$$

Из первого соотношения следует, что число слов длины $|p|$ в детерминаторе $d(J_2)$ не больше, чем число слов *p* той же длины в детерминаторе $d(J_1)$. Из второго соотношения следует обратное. Таким образом, детерминаторы $d(J_1)$ и $d(J_2)$ содержат по равному количеству слов одинаковой длины. Тогда отображения φ и ψ определяют взаимно однозначное соответствие слов одинаковой длины

детерминаторов $d(J_1)$ и $d(J_2)$, причем для соответствующих в отображении слов p и q имеем $\mu(p) = \mu(q)$.

Предположим теперь обратное, что автомат-перестановка A преобразует последовательность J_1 в последовательность J_2. Покажем, что существует обратный автомат-перестановка A^{-1} который преобразует J_2 в J_1. ДА A определяет отображение φ, удовлетворяющее условию

$$\varphi(pp_1) = \varphi(p)\varphi_p(p_1) = qq_1, \tag{6}$$

где $pp_1 \in d(J_1)$ и $qq_1 \in d(J_2)$. Так как соответствие, определяемое (6), взаимно однозначное, то существует отображение ψ: $d(J_2) \to d(J_1)$, такое, что

$$\psi(qq_1) = \psi(q)\psi_q(q_1) = pp_1. \tag{7}$$

Из свойств детерминаторов вытекает, что $p \in d(J_1)$ и $q \in d(J_2)$.

Из (6) и (7) видно также, что $\psi(q) = p$, если $q = \varphi(p)$.

Таким образом, ψ — автоматное отображение $d(J_2)$ на $d(J_1)$. Доопределим ψ некоторым образом до автоматного отображения на всей полугруппе Y^* и построим ДА A^{-1}, реализующий это полное отображение ψ. ДА A^{-1} преобразует J_2 в J_1. Если автоматный канал $\tau_A(q/p)$ индуцирован ДА A, производящим автоматное отображение φ_A, то в соответствии с леммой 1 он определяется соотношением

$$\tau_A(q/p) = \begin{cases} 1, & \text{если } q = \varphi_A(p), \\ 0 & \text{в противном случае.} \end{cases} \tag{8}$$

Используя (8), для каждого слова p из $d(J_1)$ имеем

$$\mu_{A^{-1}}(p) = \sum_{|q|=|p|} \mu(q)\,\tau_{A^{-1}}(p/q) = \sum_{p=\psi(q)} \mu(q) =$$
$$= \sum_{p\in\psi(q)} \sum_{|r|=|q|} \mu(r, q) = \sum_{p=\psi(q)} \sum_{|r|=|q|} \mu(r)\,\tau_A(q/r) =$$
$$= \sum_{p=\psi(q)} \sum_{q=\varphi(r)} \mu(r) = \sum_{p=\psi\varphi(r)} \mu(r) = \mu(p).$$

Из доказательства видно, что теорема верна и в случае конечно-автоматной эквивалентности, когда отображения φ и ψ — конечно-автоматны.

В качестве иллюстрации применения теоремы 3 рассмотрим некоторые свойства функций конечных однородных цепей Маркова.

Определение 8. Последовательность случайных кодов $J = \langle \mu(p),\ p \in X^* \rangle$ называется *стационарной с независимыми значениями,* если для любого слова $p = x_1 \ldots x_s$ выполнено условие

$$\mu(x_1 \ldots x_s) = \mu(x_1) \ldots \mu(x_s). \tag{9}$$

Теорема 5. *Для того чтобы последовательность случайных кодов $J = \langle \mu(q),\ q \in Y^* \rangle$ была функцией конечной однородной цепи Маркова, необходимо и достаточно, чтобы она была выходной*

последовательностью конечного ДА со стационарной входной последовательностью с независимыми значениями и случайным начальным состоянием.

Доказательство. Пусть стационарная последовательность с независимыми значениями $J_1 = \langle \mu(p),\ p \in X^* \rangle$ есть входная последовательность для детерминированного конечного автомата $A = \langle X,\ Y,\ \mathfrak{A},\ \delta(a,\ x),\ \lambda(a,\ x) \rangle$ с распределением вероятностей начальных состояний $\mu(a)$. Последовательность состояний $a_0, \ldots, a_s$ автомата A будет иметь вероятность

$$\mu(a_0 \ldots a_s) = \mu(a_0) \sum_{\substack{a_1=\delta(a_0,x_1) \\ \cdots\cdots \\ a_s=\delta(a_{s-1},x_s)}} \mu(x_1 \ldots x_s) =$$

$$= \mu(a_0) \prod_{i=1}^{s} \sum_{a_i=\delta(a_{i-1},x_i)} \mu(x_i),$$

следовательно, вероятность состояния a_s для входного слова длины s будет равна

$$\mu(a_s) = \sum_{a_0,\ldots,a_{s-1}} \mu(a_0) \prod_{i=1}^{s} \sum_{a_i=\delta(a_{i-1},x_i)} \mu(x_i).$$

Введем обозначение $p_{aa'} = \sum_{a'=\delta(a,x)} \mu(x)$. Тогда получим

$$\mu(a_s) = \sum_{a_0,\ldots,a_{s-1}} \mu(a_0)\, p_{a_0 a_1} \cdots p_{a_{s-1} a_s}.$$

Таким образом, последовательность состояний автомата A образует конечную однородную цепь Маркова с матрицей переходов $A = (p_{aa'})$ и начальным распределением $\mu(a)$. Рассмотрим цепь Маркова с множеством состояний $\{(a,\ x),\ a \in \mathfrak{A},\ x \in X\}$, начальным распределением $\mu(a)\mu(x)$ и вероятностями переходов

$$p_{(a,x)(a',\tilde{x})} = \begin{cases} p_{aa'}, & \text{если } \tilde{x} = x, \\ 0 & \text{в противном случае.} \end{cases}$$

Выходная последовательность автомата A есть функция $\lambda(a,\ x)$ этой цепи Маркова.

Пусть, обратно, существует стохастическая матрица $A = (p_{aa'})$, начальное распределение $\mu(a)$ и функция $y = \lambda(a)$ такие, что последовательность J есть функция λ цепи Маркова с матрицей переходов A и начальным распределением $\mu(a)$.

Пусть случайный код с распределением вероятностей $\mu(x)$ имплицирует каждое распределение вероятностей $p_{aa'}$, $a \in \mathfrak{A}$ —

строки стохастической матрицы A. Если $y = \varphi_a(x)$, $a \in \mathfrak{A}$ — система функций импликации для каждой строки матрицы A, то положим $\delta(a, x) = \varphi_a(x)$. Положим также, что $\lambda(a, x) = \lambda(a)$. Нетрудно видеть, что ДА $\langle X, Y, \mathfrak{A}, \delta(a, x), \lambda(a)\rangle$ с распределением вероятностей начальных состояний $\mu(a)$ и стационарной входной последовательностью с независимыми значениями, определенной в соответствии с (9) распределением вероятностей $\mu(x)$, имеет последовательность J в качестве выходной.

Следствие 4. *Класс функций конечных однородных цепей Маркова замкнут относительно детерминированных конечно-автоматных преобразований.*

Доказательство следует из того, что последовательное соединение конечных ДА есть конечный ДА.

Следствие 5. *Для того чюбы последовательность случайных кодов была функцией конечной однородной цепи Маркова, необходимо и достаточно, чтобы она была выходной последовательностью конечного автономного В А.*

Доказательство. Пусть $J = \langle \mu(q), \quad q \in Y^* \rangle$ — функция конечной однородной цепи Маркова. В соответствии с теоремой 5 существует начальное распределение $\mu(a)$ и система конечно-автоматных отображений $\varphi_a(p)$, $a \in \mathfrak{A}$, таких, что для любого слова $q = y_1 \ldots y_s$ из Y^* имеем

$$\mu(y_1 \ldots y_s) = \sum_{a_0 \in \mathfrak{A}} \mu(a_0) \sum_{q = \varphi_{a_0}(p)} \mu(p),$$

где $\mu(p) = \mu(x_1) \ldots \mu(x_s)$. Введем обозначение

$$\sum_{\substack{y = \varphi_a(x) \\ b = ax}} \mu(x) = \mu_{ab}(y).$$

Получим

$$\mu(y_1 \ldots y_s) = \sum_{a_0 \in \mathfrak{A}} \mu(a_0) \sum_{y_1 \ldots y_s = \varphi_{a_0}(x_1 \ldots x_s)} \mu(x_1) \ldots \mu(x_s) =$$

$$= \sum_{a_0 \in \mathfrak{A}} \mu(a_0) \sum_{y_1 = \varphi_{a_0}(x_1)} \mu(x_1) \ldots \sum_{y_s = \varphi_{a_0 x_1 \ldots x_{s-1}}(x_s)} \mu(x_s) =$$

$$= \sum_{a_0 \in \mathfrak{A}} \mu(a_0) \sum_{a_1 \in \mathfrak{A}} \sum_{\substack{y_1 = \varphi_{a_0}(x_1) \\ a_1 = a_0 x_1}} \mu(x_1) \ldots \sum_{a_s \in \mathfrak{A}} \sum_{\substack{y_s = \varphi_{a_0 x_1 \ldots x_{s-1}}(x_s) \\ a_s = a_0 x_1 \ldots x_{s-1} x_s}} \mu(x_s) =$$

$$= \sum_{a_0, a_1, \ldots, a_s} \mu(a_0) \mu_{a_0 a_1}(y_1) \ldots \mu_{a_{s-1} a_s}(y_s).$$

Система матриц $A(y) = (\mu_{aa'}(y))$ определяет конечный автономный ВА, который при начальном распределении состояний $\mu(a)$ имеет последовательность J в качестве выходной.

Обратно, пусть для каждого слова $q=y_1,\ldots,y_s$ имеем представление

$$\mu(y_1 \ldots y_s) = \sum_{a_0,\ldots,a_s} \mu(a_0)\,\mu_{a_0a_1}(y_1) \ldots \mu_{a_{s-1}a_s}(y_s), \quad (10)$$

где системы чисел $\mu_{aa'}(y) = \mu_a(a', y)$ для каждого значения $a \in \mathfrak{A}$ представляют собой вероятностные распределения. Пусть случайный код с распределением вероятностей $\mu(x)$ имплицирует каждое из распределений вероятностей $\mu_a(a', y)$. Тогда существует функция импликации $(a', y) = \varphi_a(x)$, которую можно представить в виде $a' = \delta(a, x)$, $y = \lambda(a, x)$. Введем обозпачения

$$\mu_{aa'}(y) = \sum_{\substack{y=\lambda(a,x)\\ a'=\delta(a,x)}} \mu(x).$$

Получим

$$\mu(y_1 \ldots y_s) = \sum_{a_0,\ldots,a_s} \mu(a_0)\,\mu_{a_0a_1}(y_1) \ldots \mu_{a_{s-1}a_s}(y_s) =$$

$$= \sum_{a_0,\ldots,a_s} \mu(a_0) \sum_{\substack{y_1=\lambda(a_0,x_1)\\ a_1=\delta(a_0,x_1)}} \mu(x_1) \ldots \sum_{\substack{y_s=\lambda(a_{s-1},x_s)\\ a_s=\delta(a_{s-1},x_s)}} \mu(x_s) =$$

$$= \sum_{a_0} \mu(a_0) \sum_{y_1 \ldots y_s = \varphi_{a_0}(x_1 \ldots x_s)} \mu(x_1) \ldots \mu(x_s),$$

где $\varphi_{a_0}(p)$ есть конечно-автоматное отображение, определенное ДА с функцией переходов δ и функцией выходов λ. По теореме 5 последовательность J есть функция конечной однородной цепи Маркова, ф

Формула (10) определяет функцию цепи Маркова в рекуррентной форме как прямое обобщение цепи Маркова. Метод доказательства следствия 3 дает одновременно метод синтеза ДА и входной последовательности случайных кодов, которую полученный автомат преобразует в наперед заданную функцию конечной однородной цепи Маркова.

8. Распознаваемые объекты как системы массового обслуживания

8.1. Предмет теории массового обслуживания

Одним из математических методов распознавания стохастических объектов является теория массового обслуживания, занимающаяся ***анализом эффективности функционирования*** так называемых ***систем массового обслуживания***. В нашем случае под **эффективностью функционирования систем массового**

обслуживания будем понимать **эффективность распознавания объектов.** Работа любой такой системы заключается в обслуживании (распознавании) поступающего на нее потока требований распознавания. Требования поступают на систему одно за другим в некоторые, вообще говоря, случайные моменты времени. Распознавание поступившего требования продолжается какое-то время, после чего система освобождается для распознавания очередного требования. Каждая такая система может состоять из нескольких независимо функционирующих единиц, которые будем называть каналами распознавания, или распознающими аппаратами. Примерами таких систем могут быть: телефонные станции, билетные кассы, аэродромы, вычислительные центры, радиолокационные станции и т. д. Типичной системой массового обслуживания является автоматизированная система управления производством.

Математический аппарат теории массового обслуживания позволяет оценить эффективность распознавания системой заданного потока требований в зависимости от характеристик этого потока, числа каналов системы и производительности каждого из каналов.

В качестпе критерия эффективности системы распознавания могут быть использованы различные величины и функции, например: вероятность распознавания каждого из поступающих требований, средняя доля распознаваемых требований, среднее время ожидания распознавания, среднее время простоя каждого из каналов и системы в целом, закон распределения длины очереди, пропускная способность системы и т. д. Численное значение каждого из этих критериев в той или иной степени характеризует степень приспособленности системы к выполнению поставленной перед ней задачи — удовлетворение потока поступающих в систему требований.

Часто термин «пропускная способность» используется в следующем узком смысле: среднее число требрваний, которое система может распознать в единицу времени. Эффективность систем распознавания может быть оценена также величиной относительной пропускной способности—средним отношением числа распознаных требований к числу поступивших.

В силу случайного характера моментов поступления требований процесс их распознавания представляет собой случайный процесс. Теория массового обслуживания позволяет получить математическое описание этого процесса, изучение которого дает возможность оценить пропускную способность системы и сформировать рекомендации по рациональной организации распознавания.

Все системы массового обслуживания имеют вполне определенную структуру, схематически изображенную на рис. 1.

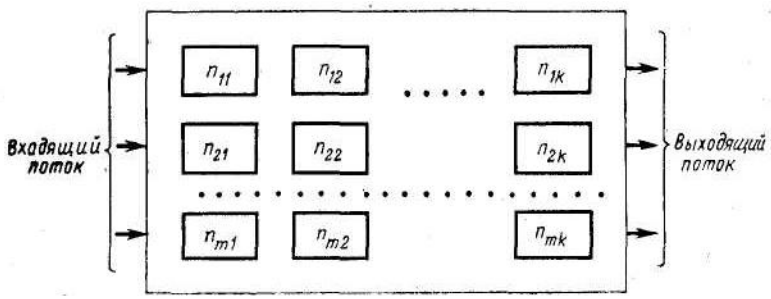


Рис. 1.

В соответствии с рисунком в любой системе массового обслуживания будем различать следующие основные элементы: входящий поток, выходящий поток, собственно система обслуживания.

Поток требований, нуждающихся в распознавании и поступающих в систему распознавания, называется ***входящим.*** Поток требований, покидающих систему распознавания, называется ***выходящим.*** Совокупность распознающих аппаратов вместе с системой правил, устанавливающих организацию распознавания, образуют ***систему распознавания***.

8.2. Входящий поток. Простейший поток и его свойства

События распознавания, образующие входяший поток, вообще говоря, могут быть различными, но здесь будет рассматриваться лишь однородный поток событий, отличающихся друг от друга только моментами появления. Такой поток можно представить в виде последовательности точек t_1, t_2, ..., t_k, ... на числовой оси (рис. 1), соответствующих моментам появления событий.

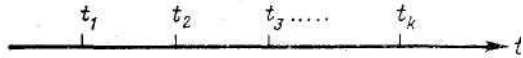


Рис. 1.

Поток событий называется ***регулярным,*** если события следуют одно за другим через строго определенные промежутки времени. Такие потоки редко встречаются при распознавании реальных требований, для которых типичным является именно случайность моментов поступления требований. Рассмотрим случайный входящий поток, обладающий особенно простыми свойствами.

Введем ряд определений.

1. Поток событий (требований) называется ***стационарным,*** если вероятность поступления заданного числа событий в течение интервала времени фиксированной длины зависит только от продолжительности этого интервала, но не зависит от его расположения на временной оси.

2. Поток событий называется ***ординарным,*** если вероятность появления двух или более событий в течение элементарного интервала времени Δt есть величина бесконечно малая по сравнению с вероятностью появления одного события на этом интервале.

3. Поток событий называется ***потоком без последействия,*** если для любых неперекрывающихся интервалов времени число событий, попадающих на один из них, не зависит от числа событий, попадающих на другие.

Если поток событий ***удовлетворяет*** всем трем перечисленным условиям (т. е. он стационарен, ординарен и не имеет последействия), то он называется ***простейшим потоком.*** Для простейшего потока число событий, попадающих на любой фиксированный интервал времени, распределено но закону Пуассона, поэтому его иначе называют ***стационарным пуассоновским***

Условию стационарности удовлетворяет поток требований, вероятностные характеристики которого не зависят от времени. В частности, постоянной является ***плотность потока*** — ***среднее число, требований в единицу времени***. Заметим, что свойство стационарности выполняется, по крайней мере на ограниченном отрезке времени, для многих реальных процессов распознавания.

Условие ординарности означает, что требования поступают в систему поодиночке, а не парами, тройками и т. д. Например, поток обстрелов, которому подвергается воздушная цель в зоне действия комплекса ЗРВ, является ординарным, если стрельба ведется одиночными ракетами, и не является ординарным, если стрельба идет одновременно двумя или тремя ракетами.

Условие отсутствия последействия является наиболее существенным для простейшего потока. Выполнение этого условия означает, что требования поступают в систему независимо друг от друга. Например, можно сказать, что ***последействие отсутствует*** для потока пассажиров, входящих в метро, так как ***отсутствует зависимость между причинами***, вызвавшими приход каждого из пассажиров на станцию. Но как только эта зависимость появляется, условие отсутствия последействия нарушается. Например, поток пассажиров, покидающих станцию метро, уже не обладает свойством последействия, так как моменты выхода для пассажиров, прибывших на станцию одним и тем же поездом, зависимы между собой.

Вообще следует заметить, что выходящие потоки требований, покидающих систему распознавания, обычно имеют последействие, даже если входящий поток его не имеет. В этом легко убедиться на примере рассмотрения выходящего потока для одноканальной системы массового обслуживания с фиксированным временем обслуживания $t_{об}$. Выходящий поток такой системы обладает тем свойством, что минимальный интервал между последовательными обслуженными заявками будет равен $t_{об}$. При этом, если в некоторый момент t_1 систему покинула заявка, то можно утверждать, что на интервале $(t_1, t_1 + t_{об})$ обслуженных заявок больше не появится и, таким образом, имеется зависимость между числом событий на неперекрывающихся интервалах.

Отметим, что, если на систему распознавания поступает самый простой, на первый взгляд, регулярный поток, анализ процессов функционирования системы является существенно более сложным, чем, например, при поступлении простейшего потока, именно вследствие жесткой функциональной зависимости, которая имеет место для требований регулярного потока.

В дальнейшем будет рассматриваться только простейший входящий поток в силу особой его роли в общей теории распознавания.

Дело в том, что простейшие или близкие к простейшим потоки требований часто встречаются в практике распознавания объектов. Кроме того, при анализе систем распознавания во многих случаях можно получить вполне удовлетворительные результаты, заменяя входящий поток требований любой структуры простейшим с той же плотностью. Наконец, важное свойство простейшего потока требований состоит в том, что при суммировании большого числа ординарных, стационарных потоков требований с практически любым последействием получается поток требований, сколь угодно близкий к простейшему. Условия, которые должны при этом соблюдаться, аналогичны условиям центральной предельной теоремы: *складываемые потоки требований должны оказывать на сумму равномерно малое влияние*.

Получим аналитическое описание простейшего потока требований и рассмотрим его свойства подробнее.

Рассмотрим на оси 0t простейший поток требований $П$ (рис. 2) как неограниченную последовательность случайных точек.

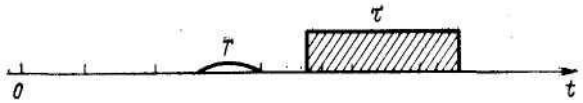


Рис. 2.

Выделим произвольный интервал времени длиной τ. Как уже отмечалось, если поток требований является простейшим, то число требований, попадающих на интервал τ, распределено по закону Пуассона с математическим ожиданием

$$a = \lambda\tau,$$

где λ — плотность потока.

В соответствии с законом Пуассона вероятность того, что за время τ поступит) ровно *m* требований, равна

$$P_m(\tau) = \frac{(\lambda\tau)^m}{m!} e^{-\lambda\tau}, \quad m = 0,\ 1,\ 2, \ldots \tag{1}$$

Тогда вероятность того, что не поступит ни одного требования, будет

$$P_0(\tau) = e^{-\lambda\tau}. \tag{2}$$

Отсюда вероятность того, что за время τ поступит хотя бы одно требование, равна

$$P_{\geq 1}(\tau) = 1 - P_0(\tau) = 1 - e^{-\lambda\tau}. \tag{3}$$

Введом величину $a = \lambda\tau$ (среднее число требований, происходящих за время τ). При этом формул» (1) перепишется в виде

$$P_m(\tau) = \frac{a^m}{m!} e^{-a}.$$

С ростом *a* распределение Пуассона асимптотически нормально со средним и дисперсией, равными *a.* Это значит, что если *Z(a)* — случайная величина, распределенная по закону Пуассона со средним значением *a,* то для больших *a*

$$P\{Z(a) \leqslant m\} \approx \frac{1}{\sqrt{2\pi a}} \int_{-\infty}^{m+1/2} \exp\left\{-\frac{(t-a)^2}{2a}\right\} dt =$$

$$= \Phi\left(\frac{m - a + 1/2}{\sqrt{a}}\right),$$

где

$$\Phi(x) = \frac{2}{\sqrt{2\pi}} \int_0^x \exp\left[-\frac{t^2}{2}\right] dt.$$

Важной характеристикой потока является закон распределения длин интервалов между требованиями. Пусть *T* — случайная длина интервала времени между двумя произвольными соседними требованиями в простейшем потоке (рис. 2) и *F(t)=P(T<t)*—искомый закон распределения продолжительности временного интервала между последовательными требованиями. С другой стороны, вероятность *P(T< t)* может быть интерпретирована как вероятность появления хотя бы одного требования в течение временного интервала

продолжительностью t, начинающегося в момент поступления в систему некоторого требования.

Поскольку простейший поток не обладает последействием, наличие требования в начале интервала t не оказывает никакого влияния на вероятность появления требования в дальнейшем. Поэтому вероятность $P(T<t)$ может быть вычислена по формуле

$$P(T<t) = 1—P(T\geq t) = 1-P_0(t), \qquad (4)$$

откуда, имея в виду (2),

$$P(T<t) = F(t)=1-e^{-\lambda t} \quad (t>0). \qquad (5)$$

Дифференцируя (5), находим плотность распределения длин интервалов между последовательными требованиями

$$f(t) = \lambda e^{-\lambda t} \quad (t>0). \qquad (6)$$

Закон распределения с плотностью (6) называется *показательным с параметром* λ.

Найдем математическое ожидание и дисперсию величины T, распределенной по показательному закону (6),

$$M[T] = \int_0^\infty t f(t)\,dt = \lambda \int_0^\infty t e^{-\lambda t} dt = \frac{1}{\lambda},$$

$$D[T] = M[(T - M[T])^2] = \int_0^\infty t^2 f(t)\,dt - \frac{1}{\lambda^2} =$$

$$= \lambda \int_0^\infty t^2 e^{-\lambda t} dt - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}. \qquad (7)$$

Выявим одно весьма важное свойство показательного закона, состоящее в следующем. Если промежуток времени, распределенный по показательному закону, уже длился некоторое время τ, то это никак не влияет на закон распределения оставшейся части промежутка: он будет таким же, как и закон распределения всего промежутка T. Для этого рассмотрим случайный промежуток времени T с функцией распределения

$$F(t)=1— e^{-\lambda t}. \qquad (8)$$

Предположим, что этот промежуток уже длился некоторое время τ, т. е. $T>\tau$. Найдем условный закон распределения оставшейся части промежутка $T_1 = T— \tau$, обозначив его через $F_\tau(t)$,

$$F_\tau(t) = P(T-\tau<t/T>\tau).$$

По теореме умножения вероятностей

$$P((T>\tau)(T-\tau<t)) = P(T>\tau)\,P(T-\tau<t/T>\tau) =$$

$$= P(T>\tau)\,F_\tau(t).$$

Отсюда

$$F_\tau(t) = \frac{P(T > \tau,\ T - \tau < t)}{P(T > \tau)}.$$

Но событие $(T > \tau) \wedge (T - \tau < t)$ равносильно событию $\tau < T < t + \tau$, вероятность которого равна

$$P(\tau < T < t + \tau) = F(t + \tau) - F(\tau).$$

С другой стороны,

$$P(T > \tau) = 1 - F(\tau),$$

следовательно,

$$F_\tau(t) = \frac{F(t + \tau) - F(\tau)}{1 - F(\tau)}$$

Отсюда, используя (8), имеем

$$F(t) = (e^{-\lambda\tau} - e^{-\lambda(t+\tau)}/e^{-\lambda\tau}) = 1 - e^{-\lambda t} = F(t), \qquad (9)$$

что и требовалось.

Таким образом, показано, что если промежуток времени распределен по показательному закону, то любая информация о его протяженности не влияет на закон распределения оставшегося времени. Доказано, что показательный закон является единственным, обладающим этим свойством. Свойство отсутствия последействия, присущее простейшему потоку, позволяет использовать для его анализа аппарат марковских цепей.

Введем состояния системы распознавания следующим образом: будем считать, что система распознавания находится в состоянии E_s в момент времени t, если к этому моменту в систему распознавания поступило s требований. Вычислим w_{ss} вероятность того, что в момент времени $t+dt$ система распознавания останется в том же состоянии. Ясно, что этому соответствует ситуация, когда за интервал dt в систему распознавания не поступит ни одного требования. В соответствии с (2) эта вероятность равна

$$w_{ss} = P_0(dt) = e^{-\lambda dt}, \quad s = 0,\ 1,2,...$$

Разлагая $e^{-\lambda dt}$ в ряд, имеем

$$w_{ss} = 1 - \lambda dt + \frac{1}{2!}(\lambda dt)^2 - ... = 1 - \lambda dt + o(dt) =$$
$$= 1 - \lambda dt, \quad s = 0,\ 1,\ 2, ... \qquad (10)$$

Вероятность поступления в систему распознавания хотя бы одного требования за интервал dt в соответствии с (3) равна

$$P_{\geq 1}(dt) = 1 - P_0(dt) = \lambda dt.$$

Учитывая свойство ординарности простейшего потока, получим

$$w_{s,\,s+1} = P_{\geqslant 1}(dt) = \lambda dt, \quad s = 0,\ 1,\ 2,\ \ldots,$$

$$w_{s,\,m} = 0, \quad m = s+2,\ s+3,\ \ldots;\ \ s = 0,\ 1,\ 2,\ \ldots \tag{11}$$

Ввиду стационарности простейшего потока, можно считать, что рассчитанные по формулам (10) и (11) вероятности перехода для момента времени t имеют то же значение и для любого другого момента времени. Тогда матрица переходов для простейшего потока приобретает вид рис. 3.

Состояние в момент $t+dt$

Состояние в момент t	E_0	E_1	E_2	E_3	. . .
E_0	$1-\lambda dt$	λdt			. . .
E_1		$1-\lambda dt$	λdt		. . .
E_2			$1-\lambda dt$	λdt	. . .
E_3				$1-\lambda dt$	. . .
. . .	. . .	. . .	. . .	. . .	. . .

Рис. 3.

Соответствующий полученной матрице переходов граф изображен на рис. 4.

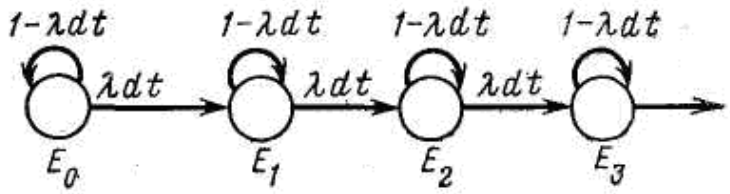


Рис. 4.

8.3. Нестационарный пуассоновский поток

Если вероятность попадания на интервал фиксированной длины τ заданного числа событий зависит не только от длины этого интервала,

но и от его расположения на временной оси, то входящий поток является нестационарным. Естественно, что в этом случае ***плотность потока*** λ — ***среднее число требований****, поступающих в систему в единицу времени,* уже не является постоянной величиной. В связи с этим характеристикой нестационарного потока является ***мгновенная плотность*** $\lambda(t)$. Мгновенной плотностью потока называется предел отношения среднего числа событий, приходящихся на элементарный интервал времени $(t, t+\Delta t)$, к длине интервала, когда последняя стремится к нулю, т. е.

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{m(t+\Delta t) - m(t)}{\Delta t},$$

где $m(t)$ - математическое ожидание числа событий на интервале $(0, t)$.

Пусть в систему распознавания поступает ординарный и без последействия, но нестационарный поток однородных событий с переменной плотностью $\lambda(t)$. Такой поток называется ***нестационарным пуассоновским потоком.***

Показано, что для такого потока число событий, попадающих на временной интервал длиной τ, начинающийся в момент t_0, распределено по закону Пуассона:

$$P_m(\tau, t_0) = \frac{\mu^m}{m!} e^{-\mu(t_0, \tau)}, \quad m = 0, 1, 2, \ldots, \tag{1}$$

где $\mu(t_0, \tau)$ —математическое ожидание числа событий на интервале $(t_0, t_0+\tau)$, равное

$$\mu(t_0, \tau) = \int_{t_0}^{t_0+\tau} \lambda(t)\, dt.$$

Здесь величина $\mu(t_0, \tau)$ зависит и от длины интервала и от его положения на оси $0t$.

Аналогично тому, как это было сделано для простейшего потока, найдем закон распределения промежутка времени T между соседними событиями. Предположим, что первое из двух последовательных событий появилось в момент t_0. При этом условии закон распределения времени T между поступившим и следующим за ним событиями $F_{t_0}(t)$ запишем в виде

$$F_{t_0}(t) = P(T < t) = 1 - P(T \geqslant t).$$

Вероятность того, что на интервале (t_0, t_0+t) не появится ни одного события, равна вероятности выполнения неравенства $T \geq t$ и вычисляется по формуле

$$P(T \geqslant t) = P_{0, t_0}(t) = \exp[-\mu(t_0, t)] = \exp\left[-\int_{t_0}^{t_0+t} \lambda(t)\, dt\right],$$

откуда

$$F_{t_0}(t) = 1 - \exp\left[-\int_{t_0}^{t_0+t} \lambda(t)\,dt\right]. \qquad (2)$$

Дифференцируя (2), получаем плотность распределения

$$f_{t_0}(t) = \lambda(t_0 + t)\exp\left[-\int_{t_0}^{t_0+t} \lambda(t)\,dt\right], \quad t > 0. \qquad (3)$$

Этот закон распределения уже не является показательным. Его характер зависит от вида функции $\lambda(t)$ и от момента t_0. Например, при линейном изменении $\lambda(t)$:

$$\lambda(t) = a + bt$$

плотность (3) имеет вид

$$f_{t_0}(t) = [a + b(t_0 + t)]\exp\left[-at - bt_0 t - \frac{bt^2}{2}\right].$$

8.4. Поток с ограниченным последействием (поток Пальма)

Рассмотренный в предыдущем пункте нестационарный пуассоновский поток является естественным обобщением простейшего потока. Обобщением этого простейшего потока в другом направлении является ***поток с ограниченным последействием***.

Рассмотрим ординарный поток однородных событий. Этот поток называется *потоком с ограниченным последействием* (потоком Пальма), если ***промежутки времени*** между последовательными событиями ***независимы***. Ясно, что простейший поток является частным случаем потока Пальма, когда независимые промежутки времени между последовательными событиями распределены по показательному закону.

Заметим, что нестационарный пуассоновский поток не является потоком Пальма, ибо, как уже было указано, для нестационарного пуассоновского потока закон распределения длины интервала между двумя последовательными событиями зависит от положения начала этого интервала на оси t.

Поскольку начало этого интервала совпадает с концом предыдущего интервала, между ними в силу вышесказанного имеется зависимость. Поток требований, покидающих какую-либо систему распознавания, может быть разбит на два: поток распознаных и поток

нераспознаных требований Относительно характера потока нераспознаных требований справедлива следующая теорема Пальма, которая здесь приводится без доказательства.

Пусть на систему массового обслуживания (распознавания) поступает поток заявок (требований) типа Пальма, причем заявка, заставшая все каналы занятыми, получает отказ (не распознается). Если при этом время обслуживания имеет показательный закон распределения, то поток необслуженных заявок является также потоком типа Пальма.

В частности, если входящий поток является простейшим, то поток нераспознаных требований, уже не являясь простейшим, все же будет иметь ограниченное последействие.

Примером потоков с ограниченным последействием являются так называемые *потоки Эрланга,* образуемые «просеиванием» простейшего потока. Пусть на вход системы распознавания поступает простейший поток требований. Если теперь исключить из потока каждое второе требование, то оставшиеся требования образуют поток, называемый *потоком Эрланга первого порядка. Поток Эрланга второго порядка* получится, если сохранить в простейшем потоке каждое третье требование. Вообще, *потоком Эрланга k-го порядка называется поток,* получаемый из простейшего, если сохранить каждое (k +1)-е требование, исключив остальные. С точки зрения этого определения простейший поток представляет собой поток Эрланга *нулевого порядка*.

Введем случайную величину T_k, равную интервалу между соседними событиями в потоке Эрланга k -го порядка. Показано, что плотность распределения $f_k(t)$ величины T_k имеет вид

$$f_k(t) = \frac{\lambda (\lambda t)^k}{k!} e^{-\lambda t}, \quad t > 0. \tag{1}$$

Закон распределения с плотностью (1) называется законом Эрланга k -го порядка. Очевидно, что при k =0 он превращается в показательный

$$f_0(t) = \lambda e^{-\lambda t}, \quad t > 0. \tag{2}$$

Поток Эрланга любого порядка представляет собой поток Пальма, поскольку из независимости двух соседних промежутков между требованиями в простейшем потоке с неизбежностью следует независимость сумм этих промежутков для любого числа слагаемых в потоке Эрланга соответствующего порядка.

8.5. Время обслуживания

Как уже отмечалось, эффективность системы распознавания зависит не только от характеристик входящего потока требований, но и от производительности самой системы распознавания, т. е. от числа каналов и быстродействия каждого из них. В связи с этим *время распознавания одной проблемы* $T_{об}$ является важной характеристикой системы распознавания. В силу самых различных причин время распознавания в реальных системах распознавания может меняться от одного требования к другомк. Поэтому в общем случае следует считать время распознавания случайной величиной.

Введем закон распределения времени распознавания

$$G(t)=P(T_{об}<t), \quad t>0$$

и плотность его распределения

$$g(t)=G'(t).$$

Для практики распознавания особый интерес представляет случай, когда продолжительность времени распознавания имеет ***показательный закон распределения***, т. е.

$$G(t)=1-e^{-\mu t} \qquad (1)$$

Параметр μ, входящий в (1), имеет простой физический смысл. Величина, обратная 1, равна математическому ожиданию времени распознавания. Действительно,

$$M[T_{об}]=\int_0^\infty t dG(t)=te^{-\mu t}\Big|_0^\infty+\int_0^\infty e^{-\mu t}dt=\frac{1}{\mu}.$$

Важная роль, которую играет показательный закон времени распознавания связана с уже упоминавшимся свойством этого закона. Применительно к данному случаю оно формулируется следующим образом: *если в какой-то момент происходит распознавание требования, то закон распределения оставшегося времени распознавания не зависит от того, сколько времени распознавание уже продолжалось.*

Таким образом, процесс распознавания требований не обладает последействием и поэтому для его анализа может быть использован аппарат теории марковских процессов.

Показательный закон распределения времени распознавания имеет место во многих задачах распознавания, когда распознавание сводится к последовательности попыток, каждая из которых приводит к необходимому результату с некоторой вероятностью.

Примером такого обслуживания является обстрел цели, заканчивающийся после поражения цели. Предположим, что

последовательность выстрелов, каждый из которых поражает цель с вероятностью *p,* образует простейший поток с плотностью λ.

Из этого потока выделим поток успешных выстрелов (выстрел будем называть успешным, если имеет место попадание в цель). Поскольку каждый из выстрелов независимо от других может оказаться успешным, поток успешных выстрелов так же, как и исходный, будет простейшим с плотностью $\Lambda = \lambda p$.

Закон распределения интервала времени между попаданиями имеет вид

$$G(t) = P(T_{об} < t) = 1 - e^{-\Delta t},$$

откуда плотность распределения времени обслуживания

$$g(t) = \Lambda e^{-\Delta t},$$

что соответствует показательному закону с параметром Λ.

Количество примеров реальных распознаваемых объектов, в которых распознавание сводится к последовательности попыток, можно значительно увеличить. К такому типу можно отнести формирование рекомендаций на обслуживание по устранению неисправностей технических устройств, когда распознавание неисправного элемента ведется путем использования ряда тестов. Совершенно аналогичной является задача распознавания, заключающаяся в формировании рекомендаций по обнаружению воздушной цели радиолокатором, многократно зондирующим исследуемое пространство, причем цель может с некоторой вероятностью обнаруживаться в каждом из циклов обзора.

Поскольку показательный закон распределения вполне приемлемым образом соответствует большому количеству реальных распознаваемых объектов обслуживания, а также в связи с тем, что основные характеристики распознаваемых объектов обслуживания зависят, главным образом, не от вида закона распределения, а от среднего значения времени распознавания, в практических распознавания обычно используется допущение о показательности закона распределения времени распознавания. Важно также, что эта гипотеза позволяет существенно упростить математический аппарат, применяемый для распознавания объектов, использующих математический аппарат теории систем массового обслуживания.

8.6. Основные типы систем массового обслуживания и показатели эффективности их функционирования

Важным признаком классификации систем массового обслуживания (систем распознавания) является поведение поступившего в систему требования в ситуации, когда все распознающие аппараты заняты. При этом в одних случаях требование не может ждать момента освобождения системы распознавания и покидает ее нераспознанным. Требование, поступившее в систему распознавания и получившее отказ, потеряно для системы. Поэтому такие системы распознавания будем называть *системами с отказами* или *системами с потерями.*

В других случаях требование может более или менее долго ожидать начала распознавания, т. е. момента освобождения одного из распознающих аппаратов системы. Совокупность таких требований образует очередь. Если при этом время ожидания для каждого из требований не ограничено, система распознавания называется *чистой системой с ожиданием* или *системой без потерь.* В противном случае, когда это время ограничено какими-либо условиями, систему называют *системой распознавания смешанного типа.* Характер ограничений в системах смешанного типа может быть различным. Во многих случаях ограничение накладывается на *продолжительность ожидания в очереди,* т. е. каждое из поступивших требований покидает систему, если распознаваниене началось до определенного момента времени, однако начатое распознавание доводится до конца. В других случаях более естественным является наложить ограничение сверху *на общее время пребывания требования в системе.* Наконец, ограничение может быть наложено на *длину очереди,* т. е. требование становится в очередь и ожидает распознавания только в том случае, если длина очереди (число ожидающих требований) не слишком велика.

Естественным критерием эффективности системы распознавания с отказами является вероятность отказа в распознавании (вероятность потери требования). Так как отказ происходит только в том случае, когда все распознающие аппараты заняты, соответствующие вероятности равны между собой.

Степень загрузки системы распознавания с отказами *характеризует закон распределения числа занятых аппаратов.* Во многих случаях для характеристики эффективности системы распознавания с отказами достаточно указать *среднее число занятых аппаратов.*

В системе распознавания без потерь требование находится до тех пор, пока не будет закончено его распознавание. Исходя из этого, могут быть сформулированы основные критерии эффективности функционирования таких систем. Это, прежде всего, *длина очереди.* Поскольку число требований, ожидающих начала распознавания в очереди, случайно, наиболее полной характеристикой этой величины является закон ее распределения. Знание этого закона позволяет рассчитать среднее число требований, ожидающих распознавания, вероятность того, что длина очереди превысит заданную и т.д. Другим важным критерием для оценки эффективности таких систем является *время ожидания начала распознавания,* наиболее полно характеризуемое своим законом распределения. С использованием этого закона может быть вычислено среднее значение времени ожидания, вероятность того, что распознавание будет начато в течение некоторого заданного интервала времени и т. п. Наконец, характеристикой таких систем является закон распределения числа аппаратов, занятых распознаванием, позволяющий рассчитать среднее число занятых аппаратов, вероятность занятости числа аппаратов, превышающее заданное, и т. п.

Для оценки эффективности систем распознавания смешанного типа могут быть использованы все перечисленные выше критерии. Кроме них, используются и некоторые специфические критерии. Например, для системы, в которой ограничено общее время пребывания требования в системе, определенный интерес представляет расчет времени, затраченного на распознавание требований, которые покидают систему *до момента окончания их распознавания.* Если частичное распознавание не обеспечивает решения задачи распознавания, то имеют место непроизводительные потери, учет которых характеризует эффективность системы распознавания.

Все перечисленные критерии в той или иной степени информативно характеризуют приспособленность рассматриваемой системы распознавания для выполнения поставленных перед ней задач. Анализ численных значений критериев позволяет сделать выводы относительно реальной эффективности системы распознавания и выработать рекомендации по ее повышению.

8.7. Система массового обслуживания с отказами

Пусть имеется *n*-канальная система массового обслуживания с отказами. Представим ее в виде некоторой системы распознавания с конечным множеством состояний:

E_0— свободны все каналы,

E_1 — занят ровно один канал,

….....

E_k — занято ровно *k* каналов,

……

E_n — заняты все *n* каналов.

Проведем оценку эффективности такой системы при следующих допущениях:

1) поток требований — простейший, с плотностью λ*;*
2) время распознавания $T_{об}$ — показательное, с параметром

$$\mu = \frac{1}{M[T_{об}]}.$$

Понятно, что параметры λ и μ по своему смыслу аналогичны. Действительно, если λ есть среднее число требований, поступающих в систему в единицу времени, то μ — среднее число требований, которое система в единицу времени в состоянии распознать.

Важно отметить, что при выполнении принятых допущений процесс перехода рассматриваемой системы из одного состояния в другое является *марковским.*

Действительно, пусть система находится в состоянии E_s (в некоторый момент времени *t* занято *s* аппаратов). Моменты поступления новых требований не зависят от того, что было до момента *t,* так как поток требований, по предположению, простейший. Моменты освобождения занятых аппаратов также не зависят от прошлого системы до момента *t* в силу показательности времени распознавания. Таким образом, моменты переходов системы в новое состояние зависят только от текущего состояния системы, но не от того, как система пришла в это состояние, т. е. система обладает марковским свойством.

В связи с этим для ***оценки эффективности*** такой системы распознавания может быть использован ***аппарат теории марковских процессов.*** Заметим, что все характеристики эффективности системы распознавания (вероятность отказа, вероятность распознавания, среднее число занятых каналов и т. д.) так или иначе определяются при использовании закона распределения вероятностей состояний

системы в установившемся стационарном режиме. Вместе с тем показано, что для любой системы распознавания такой режим устанавливается (т. е. система обладает эргодическим свойством) в том и только в том случае, если выполняется следующее условие:

$$\lambda/\mu<n. \qquad (1)$$

При выполнении условия (1) предельный вектор существует и может быть рассчитан с использованием элементов стохастической матрицы системы **W**. Следует обратить внимание на то, что вопрос о наличии предельного вектора не возникает при анализе системы с отказами, так как число возможных состояний в этой системе конечно и каждое из них, не являясь периодическим, достижимо из любого другого. Этого достаточно для того, чтобы система была эргодической. В системах распознавания смешанного типа, а также в системах без потерь выполнение условия (1) для существования предельного вектора является необходимым и достаточным.

Рассчитаем предельный вектор системы с отказами. Пусть $\mathbf{P} = (P_0\ P_1\ P_2\ \ldots\ P_n)$—вектор вероятностей различных состояний системы в установившемся режиме. Для отыскания компонент вектора используем векторно-матричное уравнение

$$\mathbf{P} = \mathbf{PW}. \qquad (2)$$

Вычислим элементы матрицы переходов системы **W**. Расчет проведем по столбцам матрицы последовательно, начиная с нулевого. $w_{00}(\Delta t)$ есть вероятность того, что за время Δt система, свободная к моменту начала интервала Δt, не будет занята к концу этого интервала.

Пусть $P_s(\Delta t)$ —вероятность поступления в систему s требований в течение Δt, а $P'_s(\Delta t)$ —вероятность распознавания s требований в течение Δt. Тогда, строго говоря,

$$w_{00}(\Delta t) = P_0(\Delta t) + P_1(\Delta t)P'_1(\Delta t)+P_2(\Delta t)P'_2(\Delta t) + \ldots \qquad (3)$$

Однако в соответствии с (1 п.8.2)

$$P_1(\Delta t) = (\lambda \Delta t)\, e^{-\lambda \Delta t} = \\ = (\lambda \Delta t)\left[1 - \lambda \Delta t + \frac{1}{2}(\lambda \Delta t)^2 - \ldots\right] = \lambda \Delta t + o(\Delta t) \qquad (4)$$

и, с другой стороны, в силу (1 п.8.5)

$$P'_1(\Delta t) = 1 - e^{-\mu \Delta t} = \\ = 1 - \left[1 - \mu \Delta t + \frac{1}{2}(\mu \Delta t)^2 - \ldots\right] = \mu \Delta t + o(\Delta t). \qquad (5)$$

Поэтому, перемножая (4) и (5), имеем

$$P_1(\Delta t))P'_1(\Delta t)) = o(\Delta t).$$

Одновременно

$$P_s(\Delta t)P'_s(\Delta t) = o(\Delta t), \qquad s = 2, 3.....$$

так как входящий поток является простейшим и обладает ординарностью.

Тогда, пренебрегая членами, имеющими меньший порядок малости по сравнению с Δt, упростим (3) к виду

$$w_{00}(\Delta t) = P_0(\Delta t) = e^{-\lambda \Delta t} \approx 1 - \lambda \Delta t. \tag{6}$$

Далее, $w_{10}(\Delta t)$ есть вероятность того, что система, в котором к началу интервала Δt имеется один занятый канал, освободится к моменту окончания интервала. Повторяя предыдущие рассуждения, получим

$$w_{10}(\Delta t) = P_0(\Delta t)\, P'_1(\Delta t) = (1 - \lambda \Delta t)\, \mu \Delta t \approx \mu \Delta t. \tag{7}$$

Наконец, можно непосредственно убедиться в том, что

$$w_{s0}(\Delta t) = o(\Delta t) \text{ для всех } s=2, 3, \ldots \tag{8}$$

Для произвольного столбца матрицы **W** с номером k $(0<k<n)$ имеем

$$\begin{aligned} w_{k,k}(\Delta t) &= P_0(\Delta t)\, P'_0(\Delta t) + P_1(\Delta t) P'_1(\Delta t) + \ldots = \\ &= P_0(\Delta t)\, P'_0(\Delta t) + o(\Delta t), \\ w_{k-1,k}(\Delta t) &= P_1(\Delta t)\, P'_0(\Delta t) + P_2(\Delta t)\, P'_1(\Delta t) + \ldots = \\ &= P_1(\Delta t)\, P'_0(\Delta t) + o(\Delta t), \\ w_{k+1,k}(\Delta t) &= P_0(\Delta t)\, P'_1(\Delta t) + P_1(\Delta t)\, P'_2(\Delta t) + \ldots = \\ &= P_0(\Delta t)\, P'_1(\Delta t) + o(\Delta t). \end{aligned}$$

Таким образом,

— переходная вероятность $w_{k,k}(\Delta t)$ равна вероятности того, что за интервал Δt не поступит ни одного нового требования и ни один из k ранее занятых каналов не освободится;

— переходная вероятность $w_{k-1,k}(\Delta t)$ равна вероятности того, что за интервал Δt в систему поступит одно новое требование и ни один из k теперь занятых каналов не освободится;

— переходная вероятность $w_{k+1,k}(\Delta t)$ равна вероятности того, что за интервал Δt не поступит ни одного нового требования и какой-либо один из $k+1$ ранее занятых каналов освободится.

Поэтому

$$w_{k,k}(\Delta t) = e^{-\lambda \Delta t} (e^{-\mu \Delta t})^k = e^{-(\lambda + k\mu)\Delta t} \approx 1 - (\lambda + k\mu)\Delta t, \tag{9}$$

$$w_{k-1,k}(\Delta t) = (\lambda \Delta t)\, e^{-\lambda \Delta t} (e^{-\mu \Delta t})^k = \lambda \Delta t e^{-(\lambda + k\mu)\Delta t} \approx \lambda \Delta t, \tag{10}$$

$$w_{k+1,k}(\Delta t) = e^{-\lambda \Delta t} (1 - e^{-\mu \Delta t})(k+1) = (k+1)\mu \Delta t. \tag{11}$$

Кроме того, как легко убедиться,

$$w_{sk}(\Delta t) = o(\Delta t), \tag{12}$$

если $|s-k| \geq 2$, $s = 0, 1, 2, \ldots, n$, $k = 0, 1, 2, \ldots, n$.

Совершенно аналогично рассчитаем элементы последнего *n*-го столбца матрицы переходов. Имеем

$$w_{n-1,n}(\Delta t) = P_1(\Delta t)P_0(\Delta t) + o(\Delta t) \approx \lambda \Delta t, \tag{13}$$

$$w_{n,n}(\Delta t) = P'_0(\Delta t) + o(\Delta t) \approx 1 - n\mu\Delta t \tag{14}$$

и, как это следует из (12),

$$w_{sk}(\Delta t) = o(\Delta t), \text{ для } s = 0, 1, 2, \ldots. n—2. \tag{15}$$

Заметим, что при вычислении $w_{n,n}(\Delta t)$ в отличие от предыдущего случая (когда $k \neq n$) не учитывается вероятность поступления в систему новых требований, так как это не может изменить состояния системы с отказами, если все каналы уже заняты.

Объединяя (6) —(8) и (11) —(15), получаем матрицу переходов системы (рис. 1).

	E_0	E_1	. . .	E_{k-1}	E_k	E_{k+1}	. . .	E_{n-1}	E_n
E_0	$1-\lambda\Delta t$	$\lambda\Delta t$	. . .	0	0	0	. . .	0	0
E_1	$\mu\Delta t$	$1-(\lambda+\mu)\Delta t$	. . .	0	0	0	. . .	0	0
	. .	. .	. . .	. .	. .	. .	. . .	. .	. .
E_{k-1}	0	0	. . .	$1-(\lambda+$ $+\mu(k-1))\Delta t$	$\lambda\Delta t$	0	. . .	0	0
E_k	0	0	. . .	$k\mu\Delta t$	$1-(\mu k+\lambda)\Delta t$	$\lambda\Delta t$	. . .	0	0
E_{k+1}	0	0	. . .	0	$(k+1)\mu\Delta t$	$1-(\lambda+$ $+\mu(k+1))\Delta t$	. . .	0	0
	0	0	. . .	. .	. .	. .	. . .	. .	. .
E_{n-1}	0	0	. . .	. .	0	0	. . .	$1-(\lambda+$ $+\mu(n-1))\Delta t$	$\lambda\Delta t$
E_n	0	0	. . .	. .	0	0	. . .	$n\mu\Delta t$	$1-n\mu\Delta t$

Рис. 1.

Соответствующий этой матрице граф изображен на рис. 2.

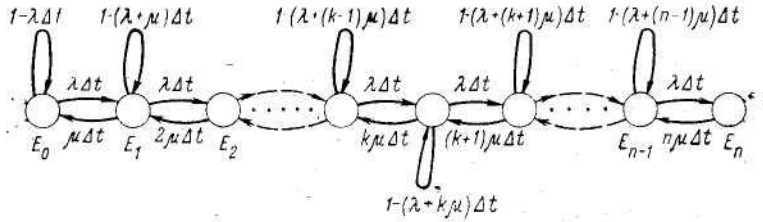


Рис. 2.

8.8. Формулы Эрланга

Подставляя полученную матрицу переходов в векторно-матричное уравнение (2 п.8.7) и преобразуя это уравнение в систему алгебраических уравнений, имеем:

$$\begin{gathered} P_0 = P_0(1-\lambda\Delta t) + P_1\mu\Delta t, \\ P_k = P_{k-1}\lambda\Delta t + P_k[1-(\lambda+\mu k)\Delta t] + P_{k+1}(k+ \\ +1)\mu\Delta t \quad (1 \leqslant k \leqslant n-1), \\ P_n = P_{n-1}\lambda\Delta t + P_n(1-n\mu\Delta t). \end{gathered} \tag{1}$$

После приведения подобных членов и сокращения на Δt получим

$$\begin{gathered} -\lambda P_0 + \mu P_1 = 0, \\ \lambda P_{k-1} - (\lambda + k\mu)P_k + (k+1)\mu P_{k+1} = 0, \quad 0 < k < n, \\ \lambda P_{n-1} - n\mu P_n = 0. \end{gathered} \tag{2}$$

Решим эту систему уравнений относительно $P_0, P_1, \ldots, P_n$, добавив условие нормировки

$$\sum_{k=0}^{n} P_k = 1. \tag{3}$$

Введем $z_k = \lambda P_{k-1} - k\mu P_k$, $k=1, 2, \ldots, n$. Тогда система уравнений (2) очевидным образом преобразуется к виду

$$\begin{gathered} z_1 = 0, \\ z_k - z_{k+1} = 0, \quad 1 \leqslant k \leqslant n-1, \\ z_n = 0 \end{gathered} \tag{4}$$

и имеет решение $z_1=z_2=\ldots = z_{n-1}= z_n = 0$. Отсюда

$$P_k = \frac{\lambda}{k\mu} P_{k-1}, \quad k = 1, 2, \ldots, n.$$

Тогда

$$\begin{gathered} P_1 = \frac{\lambda}{\mu} P_0, \\ P_2 = \frac{\lambda}{2\mu} P_1 = \frac{\lambda^2}{2\mu^2} P_0, \\ P_3 = \frac{\lambda}{3\mu} P_2 = \frac{\lambda^3}{1\cdot 2\cdot 3\mu^3} P_0. \end{gathered} \tag{5}$$

Для любого k (k= 1, 2, ..., n), таким образом, имеем

$$P_k = \frac{P_0}{k!}\left(\frac{\lambda}{\mu}\right)^k. \tag{6}$$

Для определения P_0 используем (3). Подставляя (6) в (3), получаем

$$P_0 \sum_{k=0}^{n} \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k = 1 \quad \text{и} \quad P_0 = \frac{1}{\sum_{k=0}^{n} \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k}. \tag{7}$$

Теперь, подставляя (7) в (6), имеем

$$P_k = \frac{\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k}{\sum_{l=0}^{n} \frac{1}{l!}\left(\frac{\lambda}{\mu}\right)^l}, \quad k = 0, 1, 2, \ldots, n. \tag{8}$$

Введем параметр

$$\alpha = \lambda/\mu, \tag{9}$$

который назовем *приведенной плотностью потока требований.*

Так как $1/\mu = M[T_{об}]$, то $\alpha = \lambda M[T_{об}]$ есть среднее число требований, приходящееся на среднее время распознавания одного требования.

С учетом (9) формулы (8) преобразуются к окончательному виду

$$P_k = \frac{\frac{\alpha^k}{k!}}{\sum_{l=0}^{n} \frac{\alpha^l}{l!}}, \quad k = 0, 1, 2, \ldots, n. \tag{10}$$

Соотношения (10) называются *формулами Эрланга.*

Эти формулы позволяют установить предельный закон распределения числа занятых каналов в зависимости от плотности потока требований и производительности системы распознавания.

Полагая в формуле (9) $k = n$, получим *вероятность отказа:*

$$P_{отк} = P_n = \frac{\frac{\alpha^n}{n!}}{\sum_{l=0}^{n} \frac{\alpha^l}{l!}}. \tag{11}$$

В частности, для одноканальной системы распознавания (n=1)

$$P_{\text{отк}} = \frac{\alpha}{(\alpha+1)}. \tag{12}$$

Среднее число занятых каналов распознавания равно

$$M[k] = \sum_{k=0}^{n} kP_k = \sum_{k=1}^{n} \frac{k\frac{\alpha^k}{k!}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}} = \frac{\sum_{k=1}^{n}\frac{\alpha^k}{(k-1)!}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}} =$$

$$= \alpha\frac{\sum_{k=1}^{n}\frac{\alpha^{k-1}}{(k-1)!}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}} = \alpha\frac{\sum_{l=0}^{n-1}\frac{\alpha^l}{l!}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}} = \alpha\frac{\sum_{l=0}^{n}\frac{\alpha^l}{l!} - \frac{\alpha^n}{n!}}{\sum_{l=0}^{n}\frac{\alpha^l}{l!}} =$$

$$= \alpha\left(1 - \frac{\frac{\alpha^n}{n!}}{\sum_{k=0}^{n}\frac{\alpha^l}{l!}}\right) = \alpha(1 - P_{\text{отк}}). \tag{13}$$

Заметим, что хотя формулы Эрлаига (10) выведены в предположении о показательности закона распределения времени распознавания, они верны, как это показано Б. А. Севастьяновым, и при произвольном законе распределения времени распознавания.

Пример. В четырехканальную систему распознавания с отказами поступает простейший поток требований с плотностью $\lambda=0{,}2\ c^{-1}$. Время распознавания $T_{\text{об}}$ распределено по показательному закону и $M[T_{\text{об}}]=10$ с. Рассчитать вероятность отказа и среднее число занятых каналов в системе.

Решение. Вычислим значение параметра α

$$\alpha = \lambda/\mu = \lambda M[T_{\text{об}}] = 0{,}2 \cdot 10 = 2.$$

Используя соотношение (11), рассчитаем вероятность отказа

$$P_{\text{отк}} = \frac{\frac{\alpha^n}{n!}}{\sum_{l=0}^{n}\frac{\alpha^l}{l!}} = \frac{2^4}{4!\left(1+2+\frac{2^2}{2!}+\frac{2^3}{3!}+\frac{2^4}{4!}\right)} \approx 0{,}1.$$

Подставляя рассчитанное значение для вероятности отказа в (13), вычислим среднее число занятых каналов

$$M[k] = \alpha(1 - P_{\text{отк}}) = 2(1-0{,}1) = 1{,}8 \ \text{(каналов)}.$$

8.9. Система массового обслуживании с ожиданием

Как уже отмечалось, система массового обслуживания называется системой с ожиданием, если заявка, заставшая все каналы занятыми, становится в очередь. В таких системах важную роль играет так называемая «дисциплина очереди». Ожидающие в очереди заявки могут поступать на обслуживание как в порядке очереди, так и в случайном порядке. Существуют системы массового обслуживания с приоритетом, когда некоторые выделяемые по какому-либо признаку заявки обслуживаются в первую очередь.

Каждый тип системы с ожиданием имеет свои особенности и свою математическую теорию. Здесь будет рассмотрен один из самых простых вариантов смешанной системы распознавания, часто встречающийся при распознавании различных объектов.

Пусть на вход n-канальной системы распознавания поступает простейший поток требований с плотностью λ. Время распознавания каждого из требований $T_{об}$ распределено по показательному закону с параметром $\mu=1/M[T_{об}]$. Требование, заставшее все каналы системы занятыми, становится в очередь и ожидает распознавания. Время ожидания $T_{ож}$ будем считать случайным и распределенным по показательному закону

$$H(t)=P(T_{ож}<t)=1-e^{-\nu t} \qquad (1)$$

где параметр ν — величина, обратная среднему времени распознавания, т. е. $\nu=1/M[T_{ож}]$.

Благодаря допущениям о том, что входящий поток является простейшим, а распределения времени распознавания и времени ожидания — показательные, процесс функционирования системы является марковским.

Перечислим состояния системы. Будем нумеровать их не по по числу занятых каналов, как это сделано ранее, а по числу требований, связанных с системой. При этом будем требование называть связанным с системы, если оно либо распознается, либо ожидает в очереди. Возможны состояния системы:

E_0 — свободны все каналы, очереди нет,

E_1 — занят ровно один канал, очереди нет,

……..

E_k — занято ровно k каналов, очереди нет,

……..

E_n— заняты все n каналов, очереди нет,

E_{n+1} — заняты все n каналов, одно требование стоит в очереди,

……..

E_{n+s} — заняты все n каналов, s требований — в очереди.

Поскольку число требований s, ожидающих распознавания в очереди, может быть сколь угодно большим, система имеет бесконечное (хотя и счетное) число состояний.

Анализ системы с ожиданием проведем аналогично тому, как это было сделано для системы с отказами. Вычислим элементы матрицы переходов системы. Ясно, что элементы первых n столбцов матрицы, начиная с нулевого и кончая $(n-1)$-м, не будут отличаться от соответствующих элементов матрицы переходов для системы с отказами. Однако элементы уже n-го столбца имеют некоторые отличия. Дело в том, что система с отказами, как было показано, может оказаться в состоянии E_n, если предыдущим состоянием было либо E_{n-1} либо E_n. В системе с ожиданием появляется еще одна возможность — переход в E_n из E_{n+1}.

С учетом сказанного получим формулы для расчета элементов n-го столбца. В этом столбце:

$w_{n-1,n}(\Delta t)$ —вероятность того, что за интервал Δt в систему поступит одно новое требование и ни один из $(n-1)$-го ранее занятых каналов не освободится;

$w_{n,n}(\Delta t)$ —вероятность того, что за интервал Δt не поступит ни одного нового требования и ни один из n ранее занятых каналов не освободится;

$w_{n+1,n}(\Delta t)$ —вероятность того, что за интервал Δt не поступит ни одного нового требования и либо освободится один из n ранее занятых каналов (при этом единственное ожидающее в очереди требование начнет распознаваться), либо стоящее в очереди требование покинет систему в связи с окончанием времени ожидания.

Используя рассуждения, аналогичные проведенным в предыдущем пункте, имеем

$$\begin{aligned} w_{n-1,n}(\Delta t) &= \lambda \Delta t e^{-\lambda \Delta t}(e^{-\mu \Delta t})^n \approx \lambda \Delta t, \\ w_{n,n}(\Delta t) &= e^{-\lambda \Delta t}(e^{-\mu \Delta t})^n = e^{-(\lambda + n\mu)\Delta t} \approx 1 - (\lambda + n\mu)\Delta t, \\ w_{n+1,n}(\Delta t) &= e^{-\lambda \Delta t}[(1 - e^{-\mu \Delta t})n + (1 - e^{-\nu \Delta t})] \approx \\ &\approx (1 - \lambda \Delta t)(n\mu + \nu)\Delta t \approx (n\mu + \nu)\Delta t. \end{aligned} \qquad (2)$$

Наконец, для произвольного $(n+s)$-го столбца:

$w_{n+s-1,n+s}(\Delta t)$—вероятность того, что за интервал Δt в систему поступит одно требование и ни один из ранее занятых каналов не освободится и ни одно из $(s-1)$ ранее ожидающих в очереди требование не покинет очереди;

$w_{n+s,n+s}(\Delta t)$—вероятность того, что за интервал Δt в систему не поступит ни одного требования и ни один из ранее занятых каналов не

освободится и ни одно из s ранее ожидающих в очереди требований не покинет очереди;

$w_{n+s+1,n+s}(\Delta t)$—вероятность того, что за интервал Δt в систему не поступит ни одного требования и либо освободится один из n ранее занятых каналов (при этом одно из $(s+1)$ ранее ожидающих в очереди требование начнет распознаваться), либо одно из $(s+1)$-й ожидающих распознавания требований покинет систему в связи с окончанием времени ожидания.

В соответствии с этим

$$w_{n+s-1,\ n+s}(\Delta t) = \lambda \Delta t e^{-\lambda \Delta t} (e^{-\mu \Delta t})^n (e^{-\nu \Delta t})^{s-1} \approx \lambda \Delta t,$$

$$w_{n+s,\ n+s}(\Delta t) = e^{-\lambda \Delta t} (e^{-\mu \Delta t})^n (e^{-\nu \Delta t})^s = e^{-(\lambda + n\mu + s\nu)\Delta t} \approx$$
$$\approx 1 - (\lambda + n\mu + s\nu)\Delta t,$$

$$w_{n+s+1,\ n+s}(\Delta t) = e^{-\lambda \Delta t} [(1 - e^{-\mu \Delta t})\, n + (1 - e^{-\nu \Delta t})(s+1)] =$$
$$= (1 - \lambda \Delta t)\,[n\mu + (s+1)\nu]\,\Delta t \approx [n\mu + (s+1)\nu]\,\Delta t. \qquad (3)$$

Используем (2) и (3) для формирования матрицы переходов. Рассчитаем теперь предельный вектор системы. Соответствующая система алгебраических уравнений после естественных упрощений имеет вид:

$$\begin{aligned}
&-\lambda P_0 + \mu P_1 = 0,\\
&\lambda P_0 - (\lambda + \mu) P_1 + 2\mu P_2 = 0,\\
&\lambda P_{k-1} - (\lambda + k\mu) P_k + (k+1)\mu P_{k+1} = 0, \quad 1 \leqslant k \leqslant n-1,\\
&\lambda P_{n-1} - (\lambda + n\mu) P_n + (n\mu + \nu) P_{n+1} = 0,\\
&\lambda P_{n+s-1} - (\lambda + n\mu + s\nu) P_{n+s} + [n\mu + (s+1)\nu] P_{n+s+1} = 0.
\end{aligned} \qquad (4)$$

К полученной системе уравнений необходимо добавить еще одно

$$\sum_{k=0}^{\infty} P_k = 1. \qquad (5)$$

Применим для решения этой системы алгебраических уравнений уже использованный ранее прием. Введем

$$z_k = \lambda P_{k-1} - k\mu P_k, \quad k = 1, 2, \ldots, n,$$

$$z_{n+s} = \lambda P_{n+s-1} - (n\mu + s\nu) P_{n+s}, \quad s = 1, 2, 3, \ldots$$

При этом система уравнений (4) перепишется в виде

$$
\begin{aligned}
&z_1 = 0,\\
&z_1 - z_2 = 0,\\
&\cdots\cdots\cdots\\
&z_k - z_{k+1} = 0 \quad (1 \leqslant k \leqslant n-1),\\
&\cdots\cdots\cdots\\
&z_n - z_{n+1} = 0,\\
&\cdots\cdots\cdots\\
&z_{n+s} - z_{n+s+1} = 0.\\
&\cdots\cdots\cdots
\end{aligned}
$$

Отсюда

$$z_1 = z_2 = \ldots = z_n = z_{n+1} = \ldots = z_{n+s} = z_{n+s+1} = \ldots = 0.$$

Следовательно,

$$
\begin{aligned}
&P_k = \frac{\lambda}{k\mu} P_{k-1}, \quad k = 1, 2, \ldots, n,\\
&P_{n+s} = \frac{\lambda}{n\mu + s\nu} P_{n+s-1}, \quad s = 1, 2, 3, \ldots
\end{aligned}
\tag{6}
$$

Тогда

$$
\begin{aligned}
&P_1 = \frac{\lambda}{\mu} P_0,\\
&P_2 = \frac{\lambda}{2\mu} P_1 = \frac{1}{2!} \cdot \frac{\lambda^2}{\mu^2} P_0,\\
&\cdots\cdots\cdots\cdots\\
&P_k = \frac{\lambda^k}{k!\mu^k} P_0, \quad k \leqslant n,\\
&\cdots\cdots\cdots\cdots\\
&P_n = \frac{\lambda^n}{n!\mu^n} P_0,
\end{aligned}
\tag{7}
$$

$$P_{n+1} = \frac{\lambda}{n\mu + \nu} P_n = \frac{\lambda^{n+1} P_0}{n!\mu^n (\mu n + \nu)},$$

$$P_{n+s} = \frac{\lambda^{n+s} P_0}{n!\mu^n \prod_{x=1}^{s} (n\mu + x\nu)}, \quad s = 1, 2, 3, \ldots \tag{8}$$

Заметим, что первые n формул (7) совпадают с формулами (6 п.8.8) для системы с отказами.

В формулы (7) и (8) в качестве сомножителя входит вероятность P_0. Определим ее из (5). Подставляя (7) и (8) в (5), получаем

$$P_0 = \left\{ \sum_{k=0}^{n} \frac{\lambda^k}{k!\,\mu^k} + \sum_{s=1}^{\infty} \frac{\lambda^{n+s}}{n!\mu^n \prod_{\varkappa=1}^{s} (n\mu + \varkappa\nu)} \right\} = 1,$$

Откуда

$$P_0 = \frac{1}{\sum_{k=0}^{n} \frac{\lambda^k}{k!\mu^k} + \sum_{s=1}^{\infty} \frac{\lambda^{n+s}}{n!\mu^n \prod_{\varkappa=1}^{s} (n\mu + \varkappa\nu)}}. \tag{9}$$

Преобразуем (7)— (9), введя приведенные плотности

$$\lambda/\mu = \lambda M[T_{об}] = \alpha, \qquad \nu/\mu = \nu M[T_{об}] = \beta. \tag{10}$$

Параметры α и β выражают соответственно среднее число требований и среднее число уходов требований, стоящих в очереди, приходящиеся на среднее время распознавания одного требования. В новых обозначениях формулы (7)— (9) примут вид

$$P_k = \frac{\alpha^k}{k!} P_0, \qquad 0 \leqslant k \leqslant n, \tag{11}$$

$$P_{n+s} = \frac{\frac{\alpha^{n+s}}{n!} P_0}{\prod_{\varkappa=1}^{s} (n + \varkappa\beta)}, \qquad s \geqslant 1, \tag{12}$$

$$P_0 = \frac{1}{\sum_{k=0}^{n} \frac{\alpha^k}{k!} + \frac{\alpha^n}{n!} \sum_{s=1}^{\infty} \frac{\alpha^s}{\prod_{\varkappa=1}^{s} (n + \varkappa\beta)}}. \tag{13}$$

Подставляя (13) в (11) и (13), получаем окончательные выражения для вероятностей состояний системы:

$$P_k = \frac{\frac{\alpha^k}{k!}}{\sum_{l=0}^{n} \frac{\alpha^l}{l!} + \frac{\alpha^n}{n!} \sum_{s=1}^{\infty} \frac{\alpha^s}{\prod_{x=1}^{s} (n + x\beta)}}, \quad 0 \leqslant k \leqslant n, \tag{14}$$

$$P_{n+s} = \frac{\frac{\alpha^n}{n!} \frac{\alpha^s}{\prod_{x=1}^{s} (n + x\beta)}}{\sum_{l=0}^{n} \frac{\alpha^l}{l!} + \frac{\alpha^n}{n!} \sum_{s=1}^{\infty} \frac{\alpha^s}{\prod_{x=1}^{s} (n + x\beta)}}, \quad s \geqslant 1. \tag{15}$$

Зная закон распределения вероятностей состояний системы, легко теперь рассчитать другие вероятностные характеристики системы.

Найдем математическое ожидание m_s числа требований, находящихся в очереди:

$$m_s = M[s] = \frac{\frac{\alpha^n}{n!} \sum_{s=1}^{\infty} \frac{s\alpha^s}{\prod_{x=1}^{s} (n + x\beta)}}{\sum_{k=0}^{n} \frac{\alpha^k}{k!} + \frac{\alpha^n}{n!} \sum_{s=1}^{\infty} \frac{\alpha^s}{\prod_{x=1}^{s} (n + x\beta)}}. \tag{16}$$

Теперь легко определить вероятность P_n того, что требование покинет систему нераспознанным. В самом деле, в установившемся режиме эта вероятность равна отношению среднего числа требований, покидающих очередь нераспознанными в единицу времени, к среднему числу требований, поступающих в систему в единицу времени. При этом среднее число требований, поступающих в систему в единицу времени, равно λ, — параметру входящего потока требований. Среднее же число требований $m_н$, покидающих систему нераспознанными,

можно рассчитать, зная среднее число требований, ожидающих в очереди, m_s и плотность «потока уходов» стоящих в очереди заявок ν:

$$m_н = \nu m_s = \nu \frac{\frac{\alpha^n}{n!}\sum_{s=1}^{\infty}\frac{s\alpha^s}{\prod_{x=1}^{s}(n+x\beta)}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}+\frac{\alpha^n}{n!}\sum_{s=1}^{\infty}\frac{\alpha^s}{\prod_{x=1}^{s}(n+x\beta)}}.$$

Отсюда

$$P_x = \frac{m_н}{\lambda} = \frac{\nu}{\lambda}\frac{\frac{\alpha^n}{n!}\sum_{s=1}^{\infty}\frac{s\alpha^s}{\prod_{x=1}^{\infty}(n+x\beta)}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}+\frac{\alpha^n}{n!}\sum_{s=1}^{\infty}\frac{\alpha^s}{\prod_{x=1}^{s}(n+x\beta)}} =$$

$$= \frac{\beta}{\alpha}\cdot\frac{\frac{\alpha^n}{n!}\sum_{s=1}^{\infty}\frac{s\alpha^s}{\prod_{x=1}^{s}(n+x\beta)}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}+\frac{\alpha^n}{n!}\sum_{s=1}^{\infty}\frac{\alpha^s}{\prod_{x}(n+x\beta)}}. \qquad (17)$$

Непосредственное использование формул (14)—(17) затруднено тем, что в них входят бесконечные суммы. Однако члены этих сумм быстро убывают (если $\alpha<n$). Заметим, что когда параметр $\beta\to\infty$, рассматривавмая система превращается в систему с отказами (требование мгновенно уходит из очереди). Формулы (14) при этом преобразуются в формулы Эрланга, а формулы (15) дают нули.

Приведенные выше формулы позволяют получить количественные оценки системы распознавания и для другого

крайнего случая, когда время ожидания в очереди неограниченно велико (чистая система с ожиданием). В такой системе требования вообще не покидают очереди и поэтому P_n=0.

Как уже отмечалось, в чистой системе с ожиданием не всегда имеется предельный стационарный режим. Такой режим существует лишь в случае, если $\alpha<n$, т. е. среднее число поступающих требований, приходящееся на среднее время распознавания одного требования, не превышает возможностей n-канальной системы. В противном случае $(\alpha \geq n)$ число требований, ожидающих распознавания в очереди, будет неограниченно возрастать.

Найдем предельные вероятности P_k состояний чистой системы с ожиданием для $\alpha<n$. Для этого положим в формулах (13) - (15) параметр β =0.

Имеем

$$P_0 = \frac{1}{\sum_{k=0}^{n} \frac{\alpha^k}{k!} + \frac{\alpha^n}{n!} \sum_{s=1}^{\infty} \frac{\alpha^s}{n^s}}. \tag{18}$$

Суммируя бесконечно убывающую геометрическую прогрессию в знаменателе (18), получим

$$P_0 = \frac{1}{\sum_{k=0}^{n} \frac{\alpha^k}{k!} + \frac{\alpha^{n+1}}{n!\,(n-\alpha)}}. \tag{19}$$

Отсюда, используя (14) и (15) , находим

$$P_0 = \frac{1}{\sum_{k=0}^{n} \frac{\alpha^k}{k!} + \frac{\alpha^{n+1}}{n!\,(n-\alpha)}}. \tag{20}$$

$$P_{n+s} = \frac{\frac{\alpha^{n+s}}{n!\,n^s}}{\sum_{k=0}^{n} \frac{\alpha^k}{k!} + \frac{\alpha^{n+1}}{n!\,(n-\alpha)}}, \quad s \geqslant 1. \tag{21}$$

Вычислим среднее число требований, находящихся в очереди. Непосредственная подстановка β=0 в (16) с использованием (19) дает

$$m_s = \frac{\frac{\alpha^n}{n!}\sum_{s=1}^{\infty}\frac{s\alpha^s}{n^s}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}+\frac{\alpha^{n+1}}{n!\,(n-\alpha)}}. \qquad (22)$$

С целью упрощения (22) просуммируем арифметико-геометрическую прогрессию

$$\sum_{s=1}^{\infty} s\alpha^s/n^s.$$

Введя $r = \alpha/n$, имеем

$$\sum_{s=1}^{\infty}\frac{s\alpha^s}{n^s}=\sum_{s=1}^{\infty}sr^s=r\sum_{s=1}^{\infty}sr^{s-1}=r\frac{d}{dr}\left(\sum_{s=1}^{\infty}r^s\right)=$$
$$=r\frac{d}{dr}\left(\frac{r}{1-r}\right)=\frac{r}{(1-r)^2}=\frac{\alpha}{n\left(1-\frac{\alpha}{n}\right)^2}. \qquad (23)$$

Подставляя (23) в (22), получаем

$$m_s = \frac{\frac{\alpha^{n+1}}{n!n\left(1-\frac{\alpha}{n}\right)^2}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!}+\frac{\alpha^{n+1}}{n!\,(n-\alpha)}}. \qquad (24)$$

Получим теперь формулу для расчета среднего времени ожидания требования в очереди.

Если в момент поступления требования хотя бы один из каналов системы свободен, то время ожидания равно нулю. Если требование поступает в момент, когда все каналы системы заняты, но очереди нет, то время ожидания в среднем равно $1/(n\mu)$ (так как поток освобождений в n-канальной системе имеет плотность $n\mu$). Если требование застанет все каналы занятыми и одно требование в очереди, то среднее время ожидания равно $2/(n\mu)$ и т. д. Поэтому, среднее время ожидания $\overline{t}_{\text{ож}}$ начала распознавания равно

$$\overline{t}_{\text{ож}} = \sum_{s=1}^{\infty}\frac{s}{n\mu}P_{n+s-1}.$$

Так как в соответствии с (21)

$$P_{n+s-1}=\frac{n}{\alpha}P_{n+s},$$

то

$$\overline{t}_{\text{ож}}=\sum_{s=1}^{\infty}\frac{s}{\alpha\mu}P_{n+s}=\frac{1}{\lambda}\sum_{s=1}^{\infty}sP_{n+s}=\frac{m_s}{\lambda}. \quad (25)$$

Таким образом, среднее время ожидания начала распознавания равно среднему числу требований, ожидающих в очереди, деленному на плотность потока требований.

8.10. Система смешанного типа с ограничением по длине очереди

В системах распознавания смешанного типа с ограничением по длине очереди требование, заставшее все каналы занятыми, становится в очередь лишь в том случае, если его длина не превышает некоторого q. Если же число требований в очереди уже равно q, то вновь поступившее требование покидает систему нераспознанным.

Рассмотрим такую n-канальную систему распознавания, сохранив прежние допущения о том, что входящий поток требований простейший и время распознавания распределено по показательному закону.

Число возможных состояний такой системы конечно, так как общее число требований, связанных с системой в этом случае, не может превышать $n+q$. Перечислим эти состояния:

E_0 — все каналы свободны, очереди нет.

E_1 — занят ровно один канал, очереди нет,

……

E_k — занято ровно k каналов, очереди нет,

……

E_n — заняты все n каналов, очереди нет,

E_{n+1} — заняты все n каналов, одно требование стоит в очереди,

……

E_{n+s} — заняты все n каналов, s требований стоят в очереди,

……

E_{n+q} — заняты все n каналов, q требований стоят в очереди.

Поскольку число возможных состояний системы конечно и каждое из них достижимо из любого другого, предельный вектор в такой системе существует. Заметим, кроме того, что в такой системе

распознавания требование, занявшая очередь, будет ожидать распознавания неограниченно долго. Это обстоятельство позволяет использовать для описания процесса функционирования такой системы первые $n+q$ уравнений (4 п.8.9), полученных для смешанной системы распознавания с ограничением по длительности (см. п. 8.9), считая при этом параметр $\nu=0$.

Соответствующая совокупность алгебраических уравнений имеет вид

$$\begin{aligned}
&-\lambda P_0+\mu P_1=0,\\
&\lambda P_{k-1}-(\lambda+k\mu)P_k+(k+1)\mu P_{k+1}=0, \quad 1\leqslant k\leqslant n-1,\\
&\lambda P_{n-1}-(\lambda+n\mu)P_n+n\mu P_{n+1}=0,\\
&\lambda P_{n+s-1}-(\lambda+n\mu)P_{n+s}+n\mu P_{n+s+1}=0,\\
&\lambda P_{n+q-1}-n\mu P_{n+q}=0.
\end{aligned} \tag{1}$$

Особенность структуры последнего уравнения связана, во-первых, с тем, что поступление нового требования в момент, когда система находится в состоянии E_{n+q}, не может изменить состояния системы, а, во-вторых, с тем, что состояние E_{n+q} является крайним и поэтому переход из E_{n+q+1} в E_{n+q} невозможен.

Решая так же, как и ранее, эту систему уравнений с привлечением дополнительного условия

$$\sum_{k=0}^{n+q} P_k=1,$$

окончательно получаем

$$P_k=\frac{\frac{\alpha^k}{k!}}{\sum\limits_{l=0}^{n}\frac{\alpha^l}{l!}+\frac{\alpha^n}{n!}\sum\limits_{s=1}^{q}\left(\frac{\alpha}{n}\right)^s}, \tag{2}$$

$$P_{n+s}=\frac{\frac{\alpha^n}{n!}\left(\frac{\alpha}{n}\right)^s}{\sum\limits_{k=0}^{n}\frac{\alpha^k}{k!}+\frac{\alpha^n}{n!}\sum\limits_{s=1}^{q}\left(\frac{\alpha}{n}\right)^s}, \quad 1\leqslant s\leqslant q. \tag{3}$$

Вероятность того, что требование покинет систему нераспознанным, равна вероятности P_{n+q} того, что в очереди уже стоит q требований. Нетрудно заметить, что формулы (2) и (3), как следовало

ожидать, могут быть получены из (14 п.8.9) и (15 п.8.9), если положить в них $\beta = 0$ и ограничить суммирование по s верхней границей q.

Рассчитаем среднее число требований m'_s, ожидающих распознавания в очереди. Для этого достаточно использовать соотношение (22 п.8.9), ограничив суммирование по s верхней границей q. При этом, так как

$$\sum_{s=1}^{q} \frac{s\alpha^s}{n^s} = \sum_{s=1}^{q} s \cdot r^s = r \sum_{s=1}^{q} s \cdot r^{s-1} = r \frac{d}{dr}\left(\sum_{s=1}^{q} r^s\right) =$$

$$= r \frac{d}{dr}\left(\frac{r - r^{q+1}}{1-r}\right) = r \frac{1 - (q+1)r^q + qr^{q+1}}{(1-r)^2},$$

то

$$m'_s = \frac{\dfrac{\alpha^{n+1}[1 - (q+1)r^q + qr^{q+1}]}{n \cdot n!\left(1 - \dfrac{\alpha}{n}\right)^2}}{\sum_{k=0}^{n} \dfrac{\alpha^k}{k!} + \dfrac{\alpha^{n+1}}{n!(n-\alpha)}\left[1 - \left(\dfrac{\alpha}{n}\right)^q\right]}. \qquad (4)$$

Пример. В двухканальную систему массового распознавания поступает поток требований с плотностью λ=2 1/мин. Среднее время распознавания одного требования $M[t_{об}]$=2 мин. Допустимая длина очереди равна 3. Рассчитать вероятность отказа, среднее число требований в очереди, среднее время ожидания в очереди.

Решение. Имеем: n=2, q = 3, λ =2, μ=1/ $M[t_{об}]$=0,5. По формуле (3)находим

$$P_{отк} = P_{n+q} = \frac{\dfrac{\alpha^n}{n!}\left(\dfrac{\alpha}{n}\right)^q}{\sum_{k=0}^{n} \dfrac{\alpha^k}{k!} + \dfrac{\alpha^n}{n!}\sum_{s=1}^{q}\left(\dfrac{\alpha}{n}\right)^s} =$$

$$= \frac{\dfrac{4^2}{2!} \cdot 2^3}{1 + \dfrac{4}{1} + \dfrac{4^2}{2} + \dfrac{4^2}{2}\dfrac{2 - 2^4}{1-2}} = \frac{64}{125} = 0{,}512.$$

Среднее число требований в очереди рассчитаем по формуле (4)

$$m'_s = \frac{4^3}{2 \cdot 2 \cdot 125} \cdot \frac{1 - 4 \cdot 2^3 + 3 \cdot 2^4}{(1-2)^2} = \frac{16 \cdot 17}{125} = 2{,}18.$$

Наконец, используя соотношение (25 п.8.9), находим среднее время ожидания начала распознавания для требования, вставшего в очередь:

$$\bar{t}_{\text{ож}} = \frac{m'_s}{\lambda} = \frac{2,18}{2} = 1,09 \text{ мин.}$$

8.11. Система с ожиданием. Произвольные распределения для входящего потока требований и времени распознавания

Анализ многоканальной системы с ожиданием для случая, когда законы распределения для входящего потока требований и времени распознавания произвольны, весьма затруднителен. Однако задача резко *упрощается,* если система одноканальна.

Рассмотрим одноканальную систему с ожиданием, на вход которой поступает стационарный входящий поток требований с плотностью λ и произвольным законом распределения длин интервалов между ними. Время распознавания требований также распределено по произвольному закону, причем $M[T_{\text{об}}] = 1/\mu$.

Пусть в момент, когда система закончила распознавание очередного требования, длина очереди равна u_0. Предположим, далее, что за время распознавания следующего требования $T_{\text{об}}$ в систему поступило некоторое случайное число требований r. Тогда в момент окончания распознавания очередного требования число требований в системе будет равно

$$u = \begin{cases} u_0 + r - 1, & \text{если } u_0 \neq 0, \\ r, & \text{если } u_0 = 0. \end{cases} \tag{1}$$

Соотношение (1) удобно записать в виде

$$u = u_0 + r - 1 + \delta, \tag{2}$$

где

$$\delta = \begin{cases} 1, & \text{если } u_0 = 0, \\ 0, & \text{если } u_0 \neq 0. \end{cases} \tag{3}$$

Усредняя соотношение (2) по r, имеем

$$\sum_r uP(r) = \sum_r u_0 P(r) + \sum_r rP(r) - 1 + \sum_r \delta P(r). \tag{4}$$

Если теперь учесть, что в силу стационарности входящего потока

$$\sum_r u \cdot P(r) = \sum_r u_0 P(r),$$

то соотношение (4) преобразовывается к виду

$$\sum_r \delta P(r) = 1 - \sum_r rP(r) = 1 - \lambda T_{об}. \tag{5}$$

Усредняя выражение (5) по $T_{об}$, получаем

$$\int_0^\infty \varphi(T_{об}) \sum_r \delta P(r) dT_{об} = 1 - \lambda \int_0^\infty T_{об}\varphi(T_{об}) dT_{об},$$

где $\varphi(T_{об})$ — закон распределения случайного времени распознавания. Так как

$$\int_0^\infty T_{об}\varphi(T_{об}) dT_{об} = M[T_{об}] = \frac{1}{\mu},$$

то окончательно имеем

$$\int_0^\infty \varphi(T_{об}) \sum_r \delta P(r)\, dT_{об} = 1 - \frac{\lambda}{\mu} = 1 - \alpha. \tag{6}$$

Заметим, что в левой части полученного соотношения находится среднее значение параметра δ, которое с учетом (3) может быть вычислено иначе, а именно

$$M[\delta] = 1 \cdot \text{Вер}\{u_0 = 0\} + 0 \cdot \text{Вер}\{u_0 \neq 0\} = \text{Вер}\{u_0 = 0\}.$$

Таким образом,

$$\text{Вер}\{u_0 = 0\} = P_{св} = 1 - \alpha \quad \text{и} \quad \text{Вер}\{u_0 \neq 0\} = P_{зан} = \alpha.$$

Итак, вероятность того, что одиночный канал занят, не зависит от характера законов распределения для входящего потока требований и времени распознавания и равна приведенной плотности требований α.

Для определения среднего значения длины очереди возведем в квадрат обе части соотношения (2):

$$u^2 = u_0^2 + (r-1)^2 + \delta^2 + 2u_0(r-1) + 2u_0\delta + 2\delta(r-1).$$

Найдем математические ожидания обеих частей полученного равенства, имея в виду, что $u_0\delta = 0$ и $\delta^2 = \delta$:

$$M[u^2] = M[u_0^2] + M[(r-1)^2] + M[\delta] + 2M[u_0(r-1)] + 2M[\delta(r-1)]. \tag{7}$$

Из условия стационарности следует, что

$$M[u] = M[u_0] \text{ и } M[u^2] = M[u^2_0].$$

Учтем теперь, что число поступающих в систему требований r не зависит от наличия очереди и ее длины. Тогда (7) перепишется следующим образом:

$$0 = M[(r-1)^2] + M[\delta] + 2M[u_0]M[r-1] + 2M[\delta]M[r-1]. \quad (8)$$

С другой стороны,

$$M[r] = \int_0^\infty \varphi(T_{об}) \sum_r rP(r)dT_{об} = \lambda \int_0^\infty \varphi(T_{об})T_{об}dT_{об} = \frac{\lambda}{\mu} = \alpha. \quad (9)$$

Кроме того,

$$M[\delta] = 1 - \alpha \quad (10)$$

в силу соотношения (6).

Подставляя (9) и (10) в (8), имеем

$$0 = M[r^2] - 2\alpha + 1 + (1-\alpha) + 2M[u](\alpha - 1) + 2(1-\alpha)(\alpha - 1),$$

откуда после упрощения получаем соотношения для определения среднего числа требований в системе

$$M[u] = \alpha + \frac{M[r^2] - \alpha}{2(1-\alpha)}. \quad (11)$$

Теперь легко получить формулу для расчета средней длины очереди. Очевидно, что для одноканальной системы

$$m_s = M(u) - P_{зан} = \frac{M[r^2] - \alpha}{2(1-\alpha)}. \quad (12)$$

Полученное выражение верно для любых законов распределения, длин интервалов между требованиями на входе системы и времени их распознавания, при условии, что они не зависят от длины очереди, не меняются с течением времени, а также если $\alpha<1$. Таким образом, определение значения $M[u]$ сведено к расчету величины $M[r^2]$. Рассмотрим частный случай, представляющий практический интерес. Входящий поток — пуассоновский, распределение времени распознавания — произвольное. Для пуассоновского входящего потока вероятность $P_r(T_{об})$ поступления ровно r требований в течение фиксированного времени продолжительностью $T_{об}$ равна

$$P_r(T_{об}) = \frac{(\lambda T_{об})^r}{r!} e^{-\lambda T_{об}}.$$

Кроме того, для пуассоновского закона, как известно, математическое ожидание $M_{T_{об}}(r)$ и дисперсия $Д_{T_{об}}(r)$ числа требований, поступивших в течение интервала $T_{об}$, равны $\lambda T_{об}$.

Так как

$$Д_{T_{об}}(r) = M_{T_{об}}[(r - M_{T_{об}}(r))^2] = M_{T_{об}}[r^2] - M^2_{T_{об}}[r],$$

то

$$M_{T_{об}}[r^2] = \lambda T_{об} + (\lambda T_{об})^2. \tag{13}$$

Усредняя (13) по $T_{об}$, имеем

$$M[r^2] = \lambda M[T_{об}] + \lambda^2 M[T^2_{об}]. \tag{14}$$

Учитывая, что

$$Д(T_{об}) = M[(T_{об} - M(T_{об}))^2] = M[T^2_{об}] - M^2[T_{об}],$$

$$M(T_{об}) = 1/\mu,$$

преобразуем (14) к виду

$$M[r^2] = \frac{\lambda}{\mu} + \frac{\lambda^2}{\mu^2} + \lambda^2 Д(T_{об}) = \alpha + \alpha^2 + \lambda^2 Д(T_{об}). \tag{15}$$

Подставляя (15) в (12), имеем

$$m_s = \frac{\alpha^2 + \lambda^2 Д(T_{об})}{2(1-\alpha)}. \tag{16}$$

Из выражения (16) следует, что

$$\lim_{\alpha \to 1} m_s = \infty$$

независимо от распределения времени распознавания.

Если время распознавания экспоненциально, то $Д(T_{об}) = 1/\mu^2$. При этом

$$m_s^{\exp} = \alpha^2/(1-\alpha).$$

Легко видеть, что средняя длина очереди принимает минимальное значение, если $Д(T_{об}) = 0$, т. е. если время распознавания постоянно. В этом случае

$$m_s^{\text{const}} = \frac{\alpha^2}{2(1-\alpha)}.$$

Теперь получим формулы для расчета среднего времени ожидания распознавания. Используем для этого соотношение (25 п.8.9), которое верно для любого количества каналов в системе и для произвольных распределений входящего потока и времени распознавания:

$$\bar{t}_{ож}^{\exp} = \frac{\alpha^2}{\lambda(1-\alpha)} \quad \text{и} \quad \bar{t}_{ож}^{\text{const}} = \frac{\alpha^2}{2\lambda(1-\alpha)}.$$

Таким образом, как среднее число требований в очереди, так и среднее время ожидания при строго постоянном времени распознавания вдвое меньше, чем в случае, когда оно распределено по показательному закону.

9. Метод статистических испытаний

Выше отмечалось, что если аналитическое описание процесса функционирования системы распознавания в целом получить невозможно или очень трудно, для анализа стохастических систем используется метод статистического моделирования на ЭВМ, обычно называемый методом статистических испытаний или методом Монте-Карло.

Широкое распространение этого метода связано с тем, что во многих практических случаях, когда построение аналитической модели функционирования системы распознавания в целом трудно осуществимо, удается легко описать поведение отдельных элементов системы распознавания или элементарные акты процесса ее функционирования. С другой стороны, потенциальные возможности метода выявились лишь после появления быстродействующих ЭВМ, способных в обозримое время произвести массовые расчеты, необходимые для реализации метода.

9.1. Сущность метода статистических испытаний

Метод статистических испытаний состоит в следующем. Вместо того, чтобы описывать случайное явление аналитически, производится ее моделирование с помощью некоторой процедуры, дающей случайный результат. При этом важно, чтобы количество различных исходов указанной процедуры и распределение вероятностей исходов, совпадало с соответствующими характеристиками распознаваемого явления.

Рассмотрим простой пример. По некоторой цели производится три независимых выстрела, в каждом из которых вероятность попадания p=0,5. При попадании цель поражается. Рассчитать вероятность W поражения цели. Аналитическое решение задачи элементарно:

$$W = 1 - (1-p)^3 = 1-0{,}5^3 = 0{,}875.$$

Решим теперь эту задачу методом статистических испытаний. В качестве случайного механизма используем, например, следующий. Левая страница любой раскрытой книги имеет четный номер, который

может быть кратен или не кратен 4. Поскольку кратные и некратные 4 страницы чередуются, вероятность того, что наугад раскрытая книга имеет номер левой страницы, кратный 4, равна 0,5. Пусть теперь случайная процедура состоит в трехкратном открывании книги наугад и анализе всякий раз номера левой страницы на краткость 4. Будем считать исход такого случайного эксперимента благоприятным, если хотя бы один из номеров кратен 4. Проведем такой эксперимент большое число раз n и подсчитаем число благоприятных исходов m.

При этом в соответствии с теорией вероятностей частота появления благоприятного исхода почти наверняка будет мало отличаться от вероятности этого события. Поэтому отношение m/n можно использовать в качестве удовлетворительной оценки вероятности появления благоприятного исхода описанного эксперимента. Поскольку вероятность появления кратного 4 номера страницы хотя бы один раз при трехкратном открывании книги наугад равна вероятности попадания в цель хотя бы одного выстрела из трех, это отношение m/n является также и оценкой искомой вероятности поражения цели.

В данном примере аналитическое решение гораздо проще, чем решение методом статистических испытаний. Однако можно привести множество примеров задач распознавания, когда аналитический расчет вероятности случайного события или статистических характеристик исследуемого случайного процесса является столь сложным, что получение их методом статистических испытаний оказывается совершенно оправданным.

Пусть, например, нужно рассчитать вероятность P того, что случайная величина ξ, характеризующая распознаваемый объект, с плотностью вероятности $\varphi(x)$ принимает значения в интервале $[a, b]$. Аналитически эта вероятность может быть вычислена по формуле

$$P = \int_a^b \varphi(x)dx. \qquad (1)$$

Если интеграл (1) не вычисляется в квадратурах, для оценки значения вероятности P целесообразно использовать метод статистических испытаний.

Пусть в нашем распоряжения имеется случайный механизм, формирующий случайные числа ξ с плотностью вероятности $\varphi(x)$ (способы получения такого механизма будут рассмотрены ниже). Тогда можно организовать случайную процедуру, состоящую в многократном формировании случайных чисел ξ с проверкой всякий раз выполнения неравенства $a \le \xi \le b$. Отношение числа благоприятных исходов к общему числу экспериментов дает оценку искомой

вероятности P тем лучшую, чем больше экспериментов было проведено.

Очень эффективным является использование метода статистических испытаний при моделировании процессов распознавания, эволюция которых может быть представлена в виде некоторой последовательности элементарных актов. Каждый из этих актов состоит в переходе системы с определенной вероятностью из одного состояния в другое. Если эти вероятности определяются только текущим состоянием системы и не зависят от того, как система оказалась в этом состоянии, система обладает марковским свойством. При этом для ее анализа могут быть использованы методы теории марковских цепей. Однако для многих реальных систем ее эволюция в каждый момент времени определяется не только текущим состоянием, но и предысторией. В таких случаях аналитическое исследование системы оказывается весьма затруднительным, в то время как метод статистических испытаний с этими трудностями легко справляется. При этом производится последовательное моделирование каждого элементарного акта. Пусть, например, в процессе распознавания система в некоторый момент времени t_i оказалась в состоянии E_i. В соответствии с траекторией движения системы (последовательностью прохождения состояний) до момента t_i вычислены вероятности переходов $\{P_{ij}\}$ $(j \in \mathcal{B}$, $\mathcal{B}$ — множество возможных состояний системы) из состояния E_i в другое состояние. Моделирование элементарного акта перехода из E_i осуществляется путем использования датчика случайных чисел, распределенных равномерно в интервале [0; 1]. Поскольку

$$\sum_{j \in \mathcal{B}} P_{ij} = 1,$$

интервал [0; 1] может быть разбит на совокупность подынтервалов $[\alpha^i_{k-1}, \alpha^i_k]$, число которых должно быть равно количеству возможных состояний системы. При этом правая граница каждого подынтервала вычисляется из условия

$$\alpha^i_k = \sum_{j=1}^{k} P_{ij}, \quad k \in \mathcal{B}. \tag{2}$$

Пусть теперь датчик случайных чисел, распределенных по равномерному закону в интервале [0; 1], выдает случайное число ξ. Поскольку вероятность попадания ξ внутрь каждого из подынтервалов равна его длине и в соответствии с (2)

$$\alpha^i{}_k - \alpha^i{}_{k-1} = P_{ik},$$

то закон распределения вероятностей $\{P_{ij}\}$ переходов системы из E_i в другие состояния совпадает с законом распределения вероятностей попадания ξ в подынтервалы отрезка [0; 1]. Теперь для того, чтобы определить номер состояния, в которое переходит система в данный момент времени t_i, достаточно узнать, внутри какого из подынтервалов оказалось случайное число ξ. Искомый номер подынтервала легко находится последовательной проверкой неравенств

$$\alpha^i{}_{k-1} \leqslant \xi \leqslant \alpha^i{}_k, \quad k \in \mathcal{B}.$$

Для решения этой задачи понадобился датчик случайных чисел, распределенных равномерно в интервале [0; 1]. Из дальнейшего станет ясно, что такой датчик нужен и при решении других задач методом статистических испытаний.

Поэтому рассмотрим кратко способы формирования равномерно распределенных случайных величин.

9.2. Формирование равномерно распределенных случайных величин

Проблема получения на ЭВМ случайных чисел, распределенных равномерно, может быть решена различными способами.

Простейший способ состоит в использовании хранящейся в памяти машины таблицы равномерно распределенных случайных чисел, подготовленной заранее. В случае необходимости получения случайного числа с помощью специальной команды обращаются к этой таблице и считывают очередную ее строку. Этот способ не нашел широкого распространения, главным образом, потому, что при использовании метода статистических испытаний в процессе решения задачи необходимо получение очень большого количества случайных чисел. Кроме этого, определенные трудности имеет предварительная подготовка достаточно большого (сотни тысяч или даже миллионы) количества равномерно распределенных случайных чисел.

Поэтому будем рассматривать лишь те способы построения генераторов случайных чисел, которые обеспечивают формирование этих чисел непосредственно в процессе работы машины. При этом задача состоит в получении последовательности величин, обладающих статистическими свойствами, аналогичными системе случайных чисел с заданным законом распределения. Такие последовательности принято называть псевдослучайными. Если при этом формируемая последовательность величин обладает статистическими характеристи-

ками системы равномерно распределенных случайных чисел, ее называют квазиравномерной.

ЭВМ работают, как правило, с числами, представленными в двоичной системе счисления, т. е. в виде

$$\xi = \sum_{i=1}^{n} \varepsilon_i 2^{-i}, \tag{1}$$

где

$$\varepsilon_i = \begin{cases} 1, & \text{если } 2^{-i} \text{ присутствует в двоичном разложении} \\ & \text{числа } \xi, \\ 0 & \text{в противном случае.} \end{cases}$$

Поэтому понятно, что получение равномерно распределенных n-разрядных случайных чисел сводится к отысканию способа формирования чисел в виде (1), обеспечивающего выполнение требования

$$\varepsilon_i = \begin{cases} 1 \text{ с вероятностью } 1/2, \\ 0 \text{ с вероятностью } 1/2, \; i = 1, 2, \ldots, n. \end{cases}$$

На практике используется способ формирования квазиравномерных распределений случайных чисел программным методом.

Методы программного получения псевдослучайных чисел, как правило, основаны на использовании некоторого рекуррентного вычислительного процесса, определяемого в общем случае соотношением $\xi_n = f(\xi_{n-1}, \xi_{n-2}, \ldots, \xi_{n-k})$.

Программа вычисления последовательности равномерно распределенных случайных чисел должна удовлетворять следующим требованиям:

1) полученная в результате работы программы совокупность псевдослучайных чисел должна отвечать установленным критериям проверки «случайности»;

2) случайные числа полученной совокупности должны быть весьма слабо коррелированы между собой;

3) распределение полученной совокупности псевдослучайных чисел должно достаточно хорошо аппроксимировать равномерное распределение.

Вполне удовлетворительные для практики результаты дает следующая рекуррентная процедура получения псевдослучайных чисел.

Выбирается пара чисел

$$\alpha_0 = \varepsilon_{10} \cdot 2^{-1} + \varepsilon_{20} \cdot 2^{-2} + \ldots + \varepsilon_{n0} \cdot 2^{-n},$$

$$\alpha_1 = \varepsilon_{11} \cdot 2^{-1} + \varepsilon_{21} \cdot 2^{-2} + \ldots + \varepsilon_{n1} \cdot 2^{-n}.$$

Образуем произведение этих двух чисел

$$\alpha_0 \cdot \alpha_1 = \delta_1 \cdot 2^{-1} + \delta_2 \cdot 2^{-2} + \ldots + \delta_{2n} \cdot 2^{-2n}$$

и выберем средние цифры полученного числа в качестве α_2, т. е. считая n четным, имеем

$$\alpha_2 = \delta_{n/2+1} \cdot 2^{-1} + \delta_{n/2+2} \cdot 2^{-2} + \ldots + \delta_{n/2+n} \cdot 2^{-n} =$$

$$= \varepsilon_{12} \cdot 2^{-1} + \varepsilon_{22} \cdot 2^{-2} + \ldots + \varepsilon_{n2} \cdot 2^{-n}.$$

Процесс повторяется для α_1 и α_2 с получением тем же способом α_3 и т. д.

Рассмотрим сущность программных методов формирования последовательности псевдослучайных чисел, основанных на имитации хаотического перемешивания содержимого разрядов мантиссы псевдослучайных чисел.

Типовая программа такой процедуры имеет следующую структуру:

1) изображение начального числа α_0 сдвигается на некоторое количество разрядов в сторону младших разрядов;

2) полученное при этом число складывается с α_0;

3) вычисляется абсолютная величина суммы. Полученный результат и представляет собой α_1. Далее процедура повторяется нужное число раз.

Недостатком всех программных способов получения псевдослучайных чисел является то, что получаемая при их реализации последовательность оказывается периодической. Поэтому очень длинные последовательности уже не будут случайными. Правда, при удачном выборе процедуры период может быть весьма большим (несколько миллионов). Другой недостаток состоит в трудности теоретической оценки статистических свойств получаемой последовательности.

Важным достоинством программных методов является простота практической реализации. Кроме того, при использовании этих методов возможен контроль работы машины в процессе решения задачи (возможен двойной просчет).

9.3. Формирование случайных величин с заданным законом распределения

Для формирования последовательности случайных чисел, имеющих заданный закон распределения, может быть использована последовательность равномерно распределенных случайных чисел.

Пусть η —случайное число с равномерной плотностью распределения в интервале [0; 1]. Введем случайное число ξ с помощью соотношения

$$\xi = \varphi(\eta). \tag{1}$$

причем $\varphi(\eta)$-монотонная функция. Тогда однозначно может быть получена обратная функция

$$\eta = \varphi^{-1}(\xi).$$

Запишем функцию распределения $P_{\xi}(x)$ случайной величины ξ. По определению

$$F_{\xi}(x) = p(\xi < x) = \int_{-\infty}^{x} f(x)dx, \tag{2}$$

где $f(x)$—плотность распределения случайной величины ξ.

Используя (1), имеем

$$P(\xi < x) = P[\varphi(\eta) < x]. \tag{3}$$

Так как $\varphi(\eta)$ монотонна, то

$$P[\varphi(\eta) < x] = P[\eta < \varphi^{-1}(x)]. \tag{4}$$

С другой стороны, поскольку η — равномерно распределенная случайная величина, то

$$P[\eta < \varphi^{-1}(x)] = \int_{0}^{\varphi^{-1}(x)} dy = \varphi^{-1}(x). \tag{5}$$

Объединяя (2) —(5), имеем

$$\int_{-\infty}^{x} f(x)dx = \varphi^{-1}(x), \text{ или } \int_{-\infty}^{\xi} f(x)dx = \varphi^{-1}(\xi) = \eta. \tag{6}$$

Соотношение (6) позволяет рассчитать последовательность чисел $\xi_1, \xi_2, \ldots, \xi_n, \ldots$ с заданным законом распределения $f(x)$ с использованием последовательности $\eta_1, \eta_2, \ldots, \eta_n, \ldots$ равномерно распределенных случайных чисел.

Пусть, например, требуется получить случайные числа с показательным законом распределения

$$f(x) = \lambda \cdot e^{-\lambda x}, \quad x > 0.$$

Используя (6), получаем

$$\lambda \int_0^{\xi_i} e^{-\lambda x} dx = \eta_i, \tag{7}$$

или после вычисления интеграла (7)

$$1 - e^{-\lambda \xi_i} = \eta_i.$$

Решая это уравнение относительно ξ_i, будем иметь

$$\xi_i = -\frac{1}{\lambda} \ln (1 - \eta_i). \tag{8}$$

Если в нашем распоряжении имеется последовательность η_1, η_2, ..., η_n,... чисел, распределенных равномерно, то последовательность чисел ξ_1, ξ_2, ..., ξ_n,... будет распределена по показательному закону.

При решении многих задач распознавания непосредственное использование соотношения (6) оказывается неудобным (например, когда интервал (6) не вычисляется в квадратурах). Поэтому на практике широкое распространение имеют различные приближенные приемы преобразования случайных чисел.

Рассмотрим один такой прием.

Пусть требуется получить последовательность случайных величии с функцией плотности *f(x)*. Если область определения соответствующей случайной величины ξ не ограничена, перейдем к усеченному распределению на интервале [*c; d*]. Разобьем [*c; d*] на *n* интервалов. Тогда случайная величина ξ_j может быть представлена в виде суммы

$$\xi_j = a_j + \zeta_j, \tag{9}$$

где a_i — абсцисса левой границы *j*-го интервала; ξ_j - случайная величина, возможные значения которой располагаются внутри интервала $[a_i; a_{i+1}]$.

Процедура получения случайного числа ξ_j сводится теперь к следующему:

1) случайный выбор интервала (определение значения a_i),
2) случайный выбор ξ_j из интервала $[a_i; a_{i+1}]$.
3) формирование ξ_j в соответствии с соотношением (9).

Таким образцом, для формирования одного случайного числа из их последовательности с заданным законом распределения необходимо дважды использовать датчик случайных чисел.

Понятно, что при выборе интервала на первом шаге указанной процедуры должна учитываться заданная плотность распределения *f(x).* С этой целью, используя разбиение [*c*; *d*] на *n* интервалов, кусочно-линейно аппроксимируем *f(x)* отрезками прямых, параллельных оси абсцисс (рис. 1).

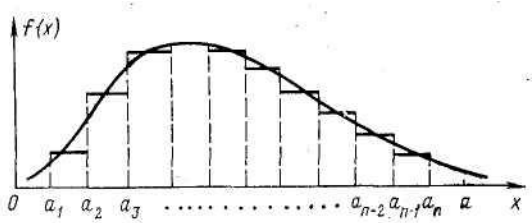


Рис.1.

Вычислим площадь каждого из прямоугольников на которые разбивается фигура, ограниченная осью абсцисс и аппроксимирующей *f(x)* ломаной.

Для приближенного вычисления площади *k*-го прямоугольника можно использовать формулу

$$S_k \approx \frac{d-c}{n} \cdot \frac{f(a_k)+f(a_{k+1})}{2}.$$

Пусть в нашем распоряжении имеется датчик случайных чисел η_1, η_2, ..., распределенных равномерно в интервале [0; 1]. Разобьем интервал [0; 1] на *n* подынтервалов (рис. 2): [α_1; α_2], [α_2; α_3], [α_3; α_4], ..., [α_n; 1] таким образом, чтобы выполнялись соотношения

$$\alpha_k = \sum_{i=1}^{k-1} S_i, \quad k = 2, 3, \ldots, n, \quad \alpha_1 = 0.$$

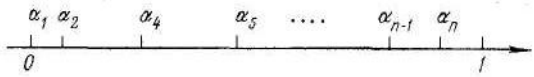


Рис. 2.

Теперь понятно, что вероятность попадания случайного числа ξ с плотностью распределения *f(x)* внутрь интервала [a_i; a_{i+1}] равна вероятности попадания равномерно распределенного случайного числа η внутрь интервала [α_i; α_{i+1}]. Поэтому при получении очередного случайного числа ξ_j из последовательности ξ_1, ξ_2, ξ_3,… номер интервала *j* отыскивается путем проверки системы неравенств

$$a_k \leqslant \eta_{j1} \leqslant a_{k+1},\ k=1,\ 2,\ 3,\ \ldots,\ n, \tag{10}$$

после чего по номеру k_j удовлетворившегося неравенства рассчитывается значение a_j.

$$a_j = c + \frac{d-c}{n}(k_j - 1). \tag{11}$$

Количество интервалов n для разбиения области определения случайной величины ξ обычно выбирают достаточно большим, чтобы распределение этой случайной величины внутри каждого из интервалов можно было бы приближенно считать равномерным. При этом случайная добавка ξ_j к величине a_j вычисляется по формуле

$$\zeta_j = \frac{d-c}{n}\eta_{j2}. \tag{12}$$

В соотношениях (11) и (12) (η_{j1}, η_{j2}) — пара случайных чисел, распределенных равномерно, используемых для формирования ξ_j.

Достоинством описанной процедуры является простота и возможность применения для формирования случайных чисел с как угодно сложным законом распределения. Недостаток состоит в необходимости проведения некоторой подготовительной работы перед непосредственным применением процедуры (разбиение области определения ξ на интервалы, подсчет величин $\{S_k\}$ и $\{\alpha_k\}$).

9.4. Применение метода статистических испытаний для анализа систем распознавания

Выше отмечалось, что задачи анализа широкого класса реально функционирующих систем распознавания могут быть решены методами теории массового обслуживания. Однако следует заметить, что результаты, имеющиеся в литературе по этому вопросу, получены главным образом для сравнительно простых случаев. Это относится прежде всего к описанию входного потока требований, а также структуры и свойств самих систем распознавания. Вместе с тем реальные входящие потоки требований по своим свойствам далеко не всегда соответствуют простейшему потоку требований, время распознавания часто не распределено по показательному закону и, наконец, логика, устанавливающая дисциплину распознавания, может быть достаточно сложной. По этим причинам аналитическое описание многих реальных систем распознавания оказывается затруднительным. В связи с этим для анализа таких систем распознавания целесообразно использовать метод статистических испытаний, справляющийся с перечисленными трудностями.

Метод статистических испытаний позволяет более полно по сравнению с аналитическими методами характеризовать зависимость эффективности систем распознавания от параметров потока требований и самой системы распознавания. Так, например, метод статистических испытаний позволяет оценить не только простейшие характеристики эффективности системы распознавания (вероятность отказа, вероятность распознавания, среднее значение доли отказов), но и значения многих других важных показателей системы распознавания (значение дисперсии доли отказов; вероятность того, что значение доли отказов будет не ниже заданного и т.п.).

Метод статистических испытаний позволяет осуществить оценку эффективности многофазных систем распознавания, аналитическое исследование которых ограничивается трудностями описания потока требований, покинувших очередную фазу распознавания и поступающих на вход следующей.

Сущность метода статистических испытаний применительно к анализу систем распознавания состоит в следующем. С помощью специальных алгоритмов формируются реализации потока требований с заданным законом распределения интервалов между требованиями. Далее моделируется процесс функционирования системы распознавания. Все показатели работы системы распознавания, интересующие исследователя, фиксируются. Общий алгоритм модели многократно воспроизводит случайные реализации процесса функционирования системы распознавания при некоторых заранее заданных условиях (характер и параметры входящего потока, параметры системы и т. д.). Накопленная в результате информация статистически обрабатывается.

Рассмотрим кратко общие принципы построения таких моделей.

Входящий поток требований однозначно задается последовательностью моментов времени поступления требований в систему t_1, t_2, ..., t_i. Удобно вместо величин t_1, t_2, ..., t_i, ..., рассматривать случайные величины ξ_1, ξ_2, ..., ξ_i, ..., определяющие длину интервалов между моментами поступления требований. При этом

$$\begin{aligned} t_1 &= \xi_1, \\ t_2 &= \xi_1 + \xi_2, \\ &\cdots\cdots\cdots \\ t_i &= \xi_1 + \xi_2 + \ldots + \xi_i. \end{aligned} \tag{1}$$

Теперь понятно, что для задания входящего потока достаточно получить последовательность случайных величин ξ_1, ξ_2, ..., ξ_i с заданным законом распределения. Методы получения такой

последовательности с использованием датчика равномерно распределенных случайных чисел были рассмотрены выше.

Пусть, например, на вход системы поступает простейший поток. Плотность распределения длин интервалов между требованиями для такого потока имеет вид

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0. \tag{2}$$

Тогда, как было показано, для получения последовательности чисел $\xi_1, \xi_2, \ldots, \xi_i, \ldots$ используется соотношение (8 п.9.3):

$$\xi_i = -\frac{1}{\lambda} \ln(1 - \eta_i),$$

где η_i — i-е случайное число из последовательности с равномерным распределением.

Предположим теперь, что на вход поступает стационарный поток с ограниченным последействием и равномерным распределением длин интервалов между требованиями. Функция плотности для такого потока имеет вид

$$f(x) = \frac{1}{b}, \quad 0 \leqslant x \leqslant b.$$

Используя соотношение (6 п.9.3), получим

$$\int_0^{\xi_i} \frac{1}{b}\, dx = \frac{\xi_i}{b} = \eta_i,$$

Откуда

$$\xi_i = b\eta_i, \quad i = 1, 2, 3, \ldots$$

Аналогично может быть получена последовательность $\xi_1, \xi_2, \ldots, \xi_i, \ldots$ и для каких-либо других законов распределения длин интервалов между требованиями.

Те же приемы используются для формирования случайных значений времени распознавания.

Принцип построения алгоритма, моделирующего процесс функционирования системы распознавания, рассмотрим на примере простейшей системы распознавания с отказами. На рис. 1 представлена логическая граф-схема алгоритма, описывающего работу такой системы распознавания.

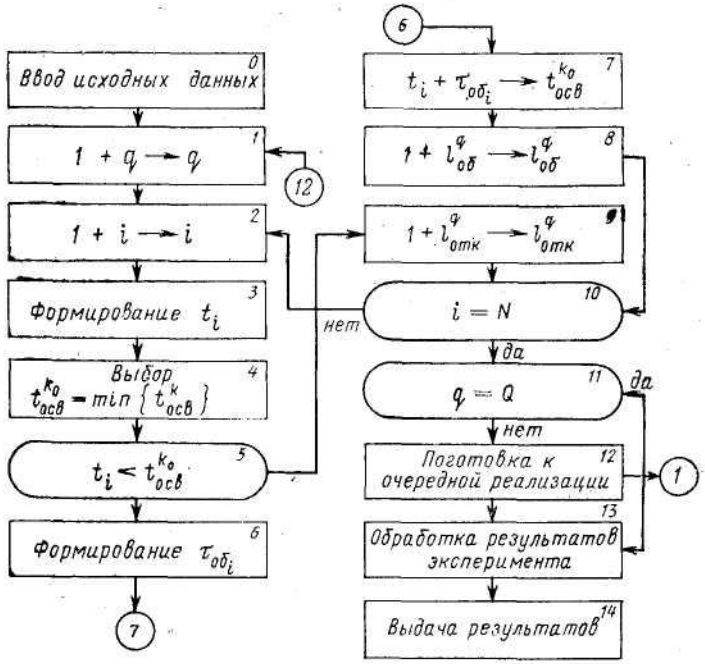


Рис. 1.

Оператор 0 осуществляет ввод исходной информации (число каналов системы, параметры входящего потока и закон распределения времени распознавания).

Оператор 1 представляет собой счетчик числа реализаций процесса функционирования системы распознавания.

Оператор 2 фиксирует номер очередного требования.

Оператор 3 формирует значения момента поступления очередного требования в соответствии с методикой, изложенной выше.

Оператор 4 осуществляет сравнение между собой моментов освобождения каналов системы распознавания и выбирает из них наиболее ранний. Пусть номер соответствующего канала равен k_0.

Оператор 5 производит сравнение величины t_i (момент поступления очередного требования) с величиной $t^{k_0}_{осв}$ (момент освобождения канала k_0). Если $t_i < t^{k_0}_{осв}$, управление передается оператору *9*, в противном случае — оператору *6*.

Оператор 9 работает в случае, если неравенство $t_i < t^{k_0}_{осв}$ выполняется, что соответствует поступлению очередного требования до освобождения какого-либо из каналов системы распознавания. При этом требование получает отказ на распознавания и значение счетчика числа отказов $l^q_{отк}$ увеличивается на единицу.

Оператор 6 работает в случае, если неравенство $t_i < t^{k_0}_{осв}$ не выполняется. При этом канал k_0, раньше всех освободившийся, начинает распознавание очередного требования. В соответствии с изложенной выше методикой формируется случайное значение времени распознавания этого требования.

Оператор 7 вычисляет время освобождения канала k_0 от распознавания очередного требования, начавшегося в момент t_i и продолжающегося в течение интервала $\tau_{об\,i}$.

Оператор 8 представляет собой счетчик числа распознанных требований.

Оператор 10 осуществляет сравнение номера очередного требования с общим числом требований N, поступающих в систему. Если равенство $i=N$ не выполняется, осуществляется переход к оператору *2*, в противном случае — к оператору *11*.

Оператор 11 проверяет, выполнено ли запланированное количество распознаваний. Если число проведенных распознаваний q еще не равно запланированному Q, управление передается оператору *12*, в противном случае — оператору *13*.

Оператор 12 осуществляет подготовку к очередной итерации. При этом очищаются рабочие ячейки памяти, хранящие значения i, $\{t^k_{осв}\}^n_1$, а содержимое ячеек $l^q_{отк}$ и $l^q_{об}$ пересылается в специальный массив для последующей статистической обработки.

Оператор 13 осуществляет статистическую обработку наборов $\{l^q_{отк}\}^Q_1$ и $\{l^q_{об}\}^Q_1$. При этом могут быть вычислены оценки для параметров, характеризующих эффективность системы.

Средняя доля распознаванных требований из общего числа поступивших требований рассчитывается по формуле

$$M[r] = \frac{\sum_{q=1}^{Q} l^q_{об}}{NQ}.$$

Дисперсия случайного значения доли распознанных требований рассчитывается по формуле

$$Д[r] = \frac{\sum_{q=1}^{Q} \left(\frac{l_{об}^{q}}{N} - M[r] \right)^2}{Q-1}.$$

Вероятность отказа на распознавание может быть оценена через частоту отказов по формуле

$$P_{отк} = \frac{\sum_{q=1}^{Q} l_{отк}^{q}}{NQ}.$$

Вероятность того, что доля распознанных требований будет не ниже заданной, может быть оценена следующим образом.

Пусть r_3 — заданная доля распознанных требований; Q_1 — количество реализаций, для каждой из которых $r = l_{об}/N \geq r_3$. Тогда

$$P(r \geq r_3) = Q_1/Q.$$

С использованием достаточно полной статистической модели системы можно оценить и многие другие показатели ее эффективности.

Сделаем несколько замечаний относительно оценки точности метода. Пусть моделируется процесс распознавания, в котором результатом каждого из N независимых распознаваний является случайная величина ξ_i (i — номер распознавания). Предположим, что эта величина обладает конечными математическим ожиданием $M[\xi_i]=a$ и дисперсией $Д[\xi_i]=\sigma^2$. Тогда среднее арифметическое

$$\bar{\xi} = \frac{1}{N} \sum_{i=1}^{N} \xi_i$$

является приближенным значением математического ожидания a. Ошибка оценки может быть установлена с помощью неравенства Чебышева:

$$P\{|\bar{\xi} - a| < \varepsilon\} > 1 - \delta; \quad N > N_0 = \sigma^2/\varepsilon^2\delta. \tag{3}$$

Соотношение (3) позволяет рассчитать число распознаваний N, которые необходимо провести для получения оценки математического ожидания с заданными точностью ε и достоверностью 1—δ.

Обычно трудно заранее оценить дисперсию σ^2. Поэтому при решении конкретных задач распознавания вместо теоретического значения σ^2 используют статистическую оценку дисперсии:

$$\Delta = \frac{\sum_{i=1}^{N} (\xi_i - \overline{\xi})^2}{N-1}.$$

Пусть теперь моделируется событие A, вероятность появления которого равна p. Введем величину

$$\xi_i = \begin{cases} 1, & \text{если событие } A \text{ произошло в } i\text{-м испытании,} \\ 0 & \text{в противном случае.} \end{cases}$$

Тогда количество испытаний, в каждом из которых событие A произошло, равно

$$L = \sum_{i=1}^{N} \xi_i,$$

где N— общее число испытаний.

Частота появления события A равна L/N и является случайной величиной, имеющей математическое ожидание

$$M\left[\frac{L}{N}\right] = \frac{1}{N} M[L] = \frac{1}{N} \sum_{i=1}^{N} M[\xi_i] = p$$

и дисперсию

$$Д\left[\frac{L}{N}\right] = \frac{1}{N^2} Д[L] = \frac{1}{N^2} \sum_{i=1}^{N} Д[\xi_i] =$$

$$= \frac{1}{N^2} \sum_{i=1}^{N} M[(\xi_i - p)]^2 =$$

$$= \frac{1}{N^2} \sum_{i=1}^{N} (M[\xi^2 i] - p^2) = \frac{p(1-p)}{N}.$$

В соответствии с законом больших чисел (теорема Бернулли) частота появления события A, равная L/N, при достаточном числе испытаний как угодно мало отличается от соответствующей вероятности p,

Необходимое число испытаний N_0, как и ранее, можно оцепить с помощью неравенства Чебышева

$$N > N_0 = \frac{\sigma^2}{\varepsilon^2 \delta} = \frac{p(1-p)}{\varepsilon^2 \delta},$$

так как

$$p^2 = Д[\xi_i] = M[(\xi_i - p)^2] = M[\xi^2 i] - p^2 = p(1-p).$$

10. Элементы теории алгоритмов

10.1. Основные определения

Приведенное раньше определение алгоритма как конечного упорядоченного набора точных правил, указывающих, какие действия и в каком порядке необходимо выполнить, чтобы после конечного числа шагов получить результат, является общеупотребительным, хотя и не обладает достаточной строгостью. Более строгое определение можно получить, исходя из следующего. Введем несколько понятий. Будем называть *алфавитом* конечное множество объектов, называемых символами (буквами). В качестве символов алфавита можно рассматривать объекты различной природы: буквы алфавита русского или любого иностранного языка или их сочетания, цифры, знаки, рисунки и т. п.

Словом в данном алфавите называется конечная упорядоченная совокупность букв. Например, в алфавите *(a, b, c)* словами будут *abc, abbbca, aaaa* и т. д. Длина слова измеряется числом входящих в него букв. Употребляется также и пустое слово, не содержащее ни одной буквы, оно обозначается знаком Ø.

Введенное понятие слова в алфавите существенно отличается от понятия слова в обычном языке, даже если алфавиты совпадают. Различие состоит в том, что словом в алфавите русского языка мы называем теперь не только все фактически существующие слова русского языка, но также и любые бессмысленные буквосочетания. Если расширить обычный алфавит русского языка, добавив к нему знаки раздела и знаки препинания, то словами в этом алфавите можно считать фразы, абзацы или даже целые книги.

Между словами в одном и том же или различных алфавитах могут быть установлены соответствия. *Алфавитным оператором* (или *алфавитным отображением)* называется соответствие, сопоставляющее словам в данном алфавите слова в этом же или каком-либо другом алфавите. В последнем случае различают входной и выходной алфавиты оператора и соответственно входные и выходные слова.

Алфавитный оператор называется однозначным, если он каждому входному слову ставит в соответствие одно вполне определенное выходное слово, если же оператор не сопоставляет никакого выходного слова, то говорят, что оператор не определен на этом входном слове. Совокупность всех слов, на которых оператор определен, называется областью его определения. Два алфавитных

оператора считаются равными, если они имеют одну и ту же область определения и любому слову из этой области ставят в соответствие одно и то же выходное слово.

Простейший способ задания алфавитного оператора состоит в установлении таблицы соответствия. В такой таблице для каждого входного слова зафиксировано соответствующее ему выходное. Примеры таких таблиц: русско-английский словарь, таблица неопределенных интегралов, таблица выигрышей денежно-вещевой лотереи, таблица условных обозначений для топографических карт и т. д. Однако такой способ задания алфавитных операторов не является универсальным, так как не может быть в принципе использован, если, например, область определения оператора включает бесконечное число слов. В таких случаях алфавитный оператор может быть задан с помощью какой-либо системы правил, позволяющих установить для любого входного слова соответствующее ему выходное слово. ***Алфавитные операторы, задаваемые с помощью конечной системы правил, называются алгоритмами.***

Два алгоритма считаются равными, если равны реализуемые ими алфавитные операторы, а также совпадают системы правил, задающие действия этих операторов на входные слова. Алгоритмы, у которых совпадают только реализуемые ими алфавитные операторы, но не совпадают способы их задания, называются эквивалентными.

Совокупность правил, образующих алгоритм, должна обладать следующими свойствами:

— полнота — в процессе решения задачи с помощью алгоритма не может возникнуть ситуация, для которой отсутствует указание относительно дальнейших действий;

— недвусмысленность — каждое из предписаний, образующих алгоритм, можно понимать лишь единственным образом;

— непротиворечивость — никакая пара предписаний алгоритма, относящихся к одной и той же ситуации, возникшей при выполнении алгоритма, не может содержать противоречащие друг другу или взаимоисключающие указания;

— массовость — возможность использования алгоритма на всем множестве возможных численных значений исходных величин;

— результативность (направленность) — способность для любых допустимых численных значений исходных данных приводить через конечное число шагов к получению требуемого результата (другими словами, результативность означает, что правильность получаемого в итоге работы алгоритма результата определяется не тем, удачно или неудачно выбраны исходные данные, а только тем, насколько верно составлен сам алгоритм).

Алгоритм, в соответствии с которым решение поставленной задачи сводится только к арифметическим действиям (сложению, вычитанию, умножению, делению), называется численным (арифметическим). Если предписания, содержащиеся в алгоритме, определяют какие-либо логические действия над исходными данными, алгоритм называют логическим. Как правило, реальные алгоритмы представляют собой совокупность арифметических и логических действий, реализуемых в определенной последовательности.

В АСУ алгоритмы классифицируются на информационно-поисковые, учета, оперативно-производственного, календарного и технико-экономического планирования, контроля, управления и т. п.

Под алгоритмом распознавания будем понимать совокупность предписаний, определяющих характер воздействия органов управления системы распознавания на органы распознавания для выполнения ими заданного алгоритма распознавания. Иными словами, алгоритмом распознавания называется совокупность предписаний, ведущих к результативному выполнению процесса распознавания в каком-либо устройстве распознавания или их совокупности (в системе распознавания).

10.2. Запись алгоритмов. Операторные схемы. Граф-схемы алгоритмов

Любой алгоритм представляет собой совокупность операторов, определяющих сущность и последовательность операций, которые надлежит выполнить для получения результата. Широко распространенным является способ записи алгоритмов в виде так называемых операторных схем, содержащих пронумерованную последовательность операторов, каждый из которых изображает группу элементарных операций. Такая запись алгоритма хотя и не содержит развернутых схем счета отдельных промежуточных величин, но тем не менее позволяет достаточно свободно ориентироваться в общей идее построения алгоритма и хорошо отражает его структуру.

Выбор системы операторов для представления алгоритма играет важную роль, так как определяет наглядность изображения алгоритма и степень удобства его использования. Обычно к системе операторов, используемых для составления алгоритма решения рассматриваемого класса задач, предъявляют два основных требования: 1) желательно, чтобы каждый используемый оператор имел ясный смысл, связанный с природой распознаваемого объекта (процесса); 2) должна быть полная

уверенность, что каждый из операторов может быть реализован с помощью последовательности элементарных операций.

Вся совокупность операторов, составляющих алгоритм, может быть разделена на три группы: основные, вспомогательные, служебные. К основным будем относить операторы, используемые для описания отдельных элементарных актов процесса распознавания и взаимодействия между ними. Иначе говоря, основные операторы реализуют соотношения математической модели, описывающие процессы распознавания элементов реальной системы с учеюм воздействия внешней среды.

В отличие от них, вспомогательные операторы осуществляют вычисление параметров и характеристик, которые необходимы для работы основных операторов.

Служебные операторы не связаны непосредственно с соотношениями математической модели. Они обеспечивают взаимодействие основных и вспомогательных операторов, синхронизацию работы алгоритма и выполняют некоторые второстепенные функции: в частности, осуществляют фиксацию величин, являющихся результатами работы алгоритма, их обработку и т. п.

Для составления операторных схем используются два типа основных операторов: арифметические и логические. Под арифметическим оператором понимают совокупность операций, реализующих какое-нибудь соотношение или систему соотношений между величинами. Арифметические операторы обычно обозначаются заглавными полужирными буквами латинского алфавита. Например, запись $\mathbf{A}_{J2}$ означает, что имеется в виду арифметический оператор № 12.

Принципиальным свойством любого арифметического оператора является его однонаправленность, т. е. после выполнения изображаемых им операций независимо от результатов расчета производится передача управления (осуществляется переход) какому-либо одному вполне определенному оператору. Другими словами, после выполнения операций, предписываемых арифметическим операторам, алгоритмический процесс может быть продолжен лишь по одному пути, независимо от результатов, выдаваемых оператором.

Передача управления от арифметических операторов обозначается приписыванием к символу, обозначающему данный оператор, справа вверху номера того оператора, которому передается управление. Например, запись $\mathbf{A}_{12}{}^{16}$ означает, что после выполнения действий, предусмотренных оператором № 12, управление передается оператору № 16.

Логические операторы предназначены для проверки справедливости заданных условий и выработки признаков, обеспечивающих результат

проверки. В наиболее простом случае проверка заданного условия сводится к сравнению по величине двух чисел. Символически операция сравнения изображается в виде равенства (например, $a = b$) или неравенства ($a>b$ или $a\geq b$). Сравнивая величины a и b, будем полагать признак ω, обозначающий результат сравнения, равным единице, если проверяемое соотношение выполняется, и равным нулю в противном случае.

Характерное свойство логических операторов состоит в том, что после реализации логического оператора управление передается одному из двух операторов, в зависимости от значения признака, вырабатываемого данным логическим оператором. Иначе говоря, в отличие от арифметических операторов, направление продолжения процесса после выполнения действий логического оператора зависит от результатов вычислений. В операторных схемах алгоритмов логические операторы обычно обозначаются буквой **P** с указанием номера оператора, например $\mathbf{P}_2$. Для изображения передачи управления от логических операторов используются специальные обозначения. В операторных схемах символ логического оператора, от которого передается управление, снабжается стрелками с номерами тех операторов, которым передается управление. При этом номер оператора, которому передается управление, если проверяемое данным логическим оператором соотношение выполнено, приписывается к стрелке, направленной вверх, в противном случае — к стрелке, направленной вниз.

Например, запись $\mathbf{P}_2^{6\uparrow 24\downarrow}$ означает, что от логического оператора № 2 управление передается оператору № 6, если условие, проверяемое $\mathbf{P}_2$, выполнено, или же оператору № 24, если оно не выполнено.

Для операторов всех классов (как арифметических, так и логических) в операторных схемах стрелка опускается, если от данного оператора управление передается непосредственно следующему за ним оператору.

Например, запись $\mathbf{A}_4\mathbf{P}_5^{8\downarrow}\mathbf{A}_6$ означает, что после выполнения арифметического оператора $\mathbf{A}_4$ управление передается логическому оператору $\mathbf{P}_5$. Затем, если выполнено проверяемое $\mathbf{P}_5$ логическое условие, управление передается непосредственно следующему за ним оператору $\mathbf{A}_6$, в противном случае — оператору №8.

Наконец, передача управления данному оператору обозначается номером того оператора, от которого передается управление, записываемым слева вверху от символа данного оператора. Например, запись $^{4,7}\mathbf{A}_{12}$ означает, что оператору №12 управление передается от операторов №4 и №7. Передача управления данному оператору от

предыдущего изображается лишь в тех случаях, когда данному оператору передается управление от нескольких операторов.

Помимо арифметических и логических операторов алгоритмы, как правило, содержат другие специфические операторы (например, оператор ввода исходных данных в ЭВМ, оператор вывода полученного числового материала, оператор останова и т. п.).

Для некоторых наиболее употребительных из них приняты стандартные обозначения, которые сведены в таблицу.

Наименование оператора	Выполняемая работа	Обозначение
Перенос	Перенос чисел из одного запоминающего устройства в другое; ввод исходных данных в ЭВМ	П
Переадресация (по параметру i)	Переадресация команд	F (i)
Засылка	Вынесение величин в стандартные ячейки	З
Обращение	Обращение к группе операторов с номерами m, $m+1$, ..., n	Е (m; n)
Формирование	Формирование новых команд	Ф
Останов	Останов машины	Я
Нестандартный	Любой оператор, отличный от вышеперечисленных	Н

Способ записи алгоритмов в виде операторных схем удобен при непосредственном программировании алгоритма. Однако при анализе алгоритмов его использование затруднительно, так как операторная схема не обладает достаточной наглядностью, особенно если алгоритм имеет сложную логическую структуру. В связи с этим на практике часто используются так называемые граф-схемы алгоритмов.

Арифметические операторы графически изображаются в виде прямоугольников, внутри которых записано аналитическое соотношение, реализуемое оператором. Передача управления от арифметического оператора изображается стрелкой, выходящей из прямоугольника, обозначающего оператор, от которого передается управление, и направленной к изображению оператора, которому передается управление.

Логические операторы графически изображаются в виде ромбов (иногда овалов), внутри которых словами или символически записано проверяемое оператором условие.

Передача управления от логических операторов изображается в виде двух стрелок, одна из которых отмечается единицей (соответствует передаче управления при выполнении проверяемого условия), а вторая — нулем (соответствует передаче управления при невыполнении проверяемого условия).

Пример. Составить алгоритм выбора максимального по величине члена из последовательности чисел { a_1, a_2, …, a_n}. Записать алгоритм в виде операторной схемы и граф-схемы.

Решение. В соответствии со структурой задачи предусмотрим в алгоритме наличие следующих операторов:

— оператор ввода исходных д.шных $\mathbf{П}_0$;

— арифметический оператор $\mathbf{А}_1$ присвоения ячейке, содержащей номер i очередного члена последовательности, значения i+1;

— логический оператор $\mathbf{Р}_2$ сравнения содержимого ячейки i с содержимым некоторой рабочей ячейки $ря_0 (a_i \geqslant ря_0)$;

— арифметический оператор $\mathbf{А}_3$ присвоения рабочей ячейке $ря_0$ содержимот ячейки i;

— арифметический оператор $\mathbf{А}_4$ присвоений рабочей ячейке $ря_1$. значения i, соответствующего номеру очередного анализируемого члена последовательности;

— логический оператор $\mathbf{Р}_1$ сравнения номера очередного члена i с общим числом членов последовательности $n(i=n)$;

— нестандартный оператор $\mathbf{Н}_6$ печати содержимого рабочих ячеек $ря_0$ и $ря_1$;

— оператор останова $\mathbf{Я}_7$.

Операторная схема алгоритма имеет вид:

$$\mathbf{П}_0^5\mathbf{А}_1\mathbf{Р}_2^{5\downarrow}\mathbf{А}_3\mathbf{А}_4^2\mathbf{Р}_5^{1\downarrow}\mathbf{Н}_6\mathbf{Я}_7.$$

Граф-схема алгоритма изображена на рис. 1.

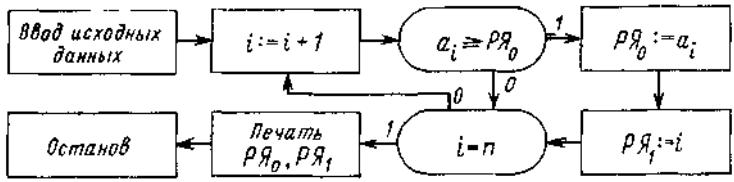


Рис. 1.

10.3. Построение алгоритмов

Построение алгоритма является таким этапом задачи распознавания объекта (процесса), когда формализованная модель распознаваемого объекта уже получена и решены все принципиальные вопросы, связанные с выбором математического аппарата распознавания.

При построении алгоритма прежде всего намечаются основные операторы. Они должны быть увязаны между собой в соответствии с формализованной моделью распознаваемого объекта. Далее необходимо установить, значения каких параметров нужно распознать для обеспечения работы основных операторов. Эти сведения служат исходным материалом для введения в операторную схему вспомогательных операторов. После того, как достаточно отработана главная часть алгоритма, можно переходить к следующему шагу — введению служебных операторов. Для этого необходимо рассмотреть динамику функционирования элементов распознаваемого объекта, выявить величины, подлежащие фиксации, и проанализировать процесс их обработки. Задачи построения алгоритмов принадлежат к творческим и трудно формализуются.

Создано большое количество формальных методов синтеза алгоритмов. Разработан так называемый канонический метод составления алгоритмов. Этот метод удобен при составлении достаточно простых алгоритмов. Однако сложность его практической реализации быстро возрастает с увеличением числа операторов. Тем не менее важным достоинством метода является возможность его формализации, что в принципе позволяет использовать для решения задач синтеза алгоритмов компьютерную технику. Опишем модифицированный канонический метод построения граф-схем алгоритмов. Практическое построение граф-схемы алгоритма требует решения следующих вопросов:

1) выбор определенной последовательности логических операторов $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n$, в которой их взаимное расположение наилучшим образом отражает естественное течение алгоритмизируемого процесса;

2) построение по последовательности $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n$ таблицы всех 2^n наборов возможных значений логических переменных;

3) определение для каждого из наборов соответствующей последовательности выполнения арифметических операторов;

4) объединение всех последовательностей в одну граф-схему;

5) выполнение эквивалентных преобразований полученной граф-схемы с целью ее минимизации.

Упорядочение массива логических операторов $\{\mathbf{P}_j\}$ в последовательность, удовлетворяющую требованиям п. 1, осуществляется на основе анализа условий задачи, подлежащей алгоритмизации. Пусть такая последовательность выбрана. Построение таблицы наборов значений логических переменных и определение соответствующих последовательностей выполнения арифметических операторов осуществляется с помощью так называемой характеристической таблицы.

Характеристическая таблица содержит 2^n строк и две группы столбцов, причем первая группа состоит из n столбцов, обозначаемых $\mathbf{P}_1$, $\mathbf{P}_2$, ..., $\mathbf{P}_n$, а вторая — из $n+1$ столбцов, обозначаемых соответственно N_0, N_1, ..., N_n.

Строки первой группы столбцов заполняются нулями и единицами таким образом, чтобы их расположение соответствовало записи числа, равного номеру соответствующей строки в двоичной системе счисления. Нумерация строк при этом осуществляется по порядку, начиная с нуля. Поэтому в верхней строке таблицы записываем набор 0, 0, ..., 0, в следующей строке — набор 0, 0, ..., 0, 1 и так далее до последней строки, которой соответствует набор 1, 1, ..., 1 (рис. 1).

$\mathbf{P}_1$	$\mathbf{P}_2$	. . .	$\mathbf{P}_{n-1}$	$\mathbf{P}_n$	N_0	N_1	N_2	. . .	N_n
0	0	. . .	0	0				. . .	
0	0	. . .	0	1				. . .	
.	.	.	.	.	.	.	.	.	.
1	1	. . .	1	0				. . .	
1	1	. . .	1	1				. . .	

Рис. 1

Столбцы второй группы заполняются последовательно, начиная с N_0-го. Очередной столбец N_i заполняется построчно. *В* каждую строку этого столбца записываются номера тех арифметических операторов, которые могут быть выполнены при указанных в строке значениях логических операторов $\mathbf{P}_1$, $\mathbf{P}_2$, ..., $\mathbf{P}_i$. Кроме того, в ту же строку вписываются номера тех арифметических операторов, которые должны быть выполнены, чтобы оказалось возможным реализовать проверку логического уеловия $\mathbf{P}_{i+1}$. Если указанные арифметические

операторы отсутствуют или если для данного набора значений логических переменных $\mathbf{P_1}, \mathbf{P_2}, \ldots, \mathbf{P_i}$ логическая переменная $\mathbf{P_{i+1}}$ теряет смысл, то в соответствующую строку столбца N_i заносится пустой оператор **Z.**

После заполнения характеристической таблицы осуществляется построение дерева алгоритма. Дерево алгоритма представляет собой граф, состоящий из вершин, соединенных направленными дугами (стрелками) и строится в соответствии со следующим формальным правилом.

Правило 1 (построение дерева).

1. Из характеристической таблицы выписываются группы арифметических операторов, содержащиеся в столбцах $N_0, N_1, \ldots, N_n$.
2. Каждая группа операторов соответствует вершине дерева алгоритма.
3. Каждая вершина, расположенная в i-м столбце, соединяется стрелками с парой соответствующих вершин (i+l)-го столбца. Понятно, что строкам, содержащим одинаковые значения логических переменных, соответствуют одинаковые группы арифметических операторов.

В столбцах $\mathbf{P_1}, \mathbf{P_2}, \ldots, \mathbf{P_i}$ имеется 2^i групп строк, содержащих одинаковые наборы значений логических переменных. Всем строкам каждой из таких групп соответствует один и тот же набор арифметических операторов, помещаемый в N_i. Таким образом, в столбце N_i может содержаться не более чем 2^i различных наборов арифметических операторов. Поэтому при построении дерева в i-м столбце образуется 2^i вершин.

После выполнения действий, предписываемых сформулированным правилом, дерево алгоритма построено. Это дерево может быть существенно упрощено эквивалентными преобразованиями. Переходим к описанию эквивалентных преобразований дерева алгоритма. Введем несколько определений.

Кустом дерева называется подграф, содержащий все вершины, связанные с начальной вершиной куста. Куст обозпачастся ломером начальной вершины.

Вершиной i-го ранга называется совокупность арифметических операторов, выполняемых непосредственно после проверки i-го логического условия. Арифметические операторы, образующие вершины i-го ранга, находятся в i-м столбце характеристической таблицы.

Рассмотрим подграф, состоящий из вершины некоторого i-го ранга, содержащей множество $B_0^{(i)}$ арифметических операторов, и двух

связанных с ней вершин старшего на единицу ранга, содержащих соответственно множества $B_1^{(i+1)}$, $B_2^{(i+1)}$ операторов (рис. 2).

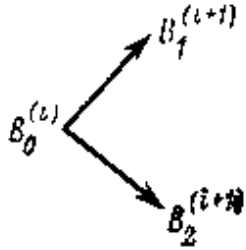
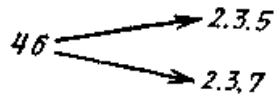


Рис. 2

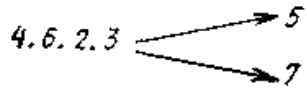
Введем понятие *B*-пересечения. *B*-пересечение B_1 и B_2 не пусто и содержит ровно *k* операторов, если *первые по порядку k* операторов из B_1 и B_2 совпадают.

Правило 2 (правило перемещения операторов). Если *B*-паресечение B_1 и B_2 не пусто, то соответствующее этому пересечению подмножество операторов приписывается справа к B_0 и вычеркивается из B_1 и B_2. Правило перемещения применяется последовательно ко всем вершинам равного ранга, начиная с (*n*—1)-го. Указанное правило не распространяется на оператор **Z**. Если в результате применения правила B_1 или B_2 оказывается пусто, то на этом месте ставится оператор **Z.**

Так, например, если подграф (B_0, B_1, B_2) имеет вид



то после применения правила 2 он преобразуется к виду



Введем понятие *обобщенный номер* вершины, представляющий собой дробь (A/a), в числителе которой стоит набор выполняемых арифметических операторов, а в знаменателе — номер вершины.

Правило 3 (правило нумерации вершин).

1. В последнем *n*-м столбце вершины нумеруются сверху вниз, начиная с единицы, причем равные группы арифметических операторов номеруются одинаково.

2. Нумерация вершин во всех остальных столбцах осуществляется для вершин равного ранга последовательно, начиная с (n—1)-го. Пусть вершины рангов $i+1, i+2, \ldots, n$ пронумерованы. Номер каждой вершины i-го ранга определяется парой связанных с ней обобщенных номеров, соответствующих вершинам (i +1) -го ранга. При этом:

а) если пара состоит из равных обобщенных номеров, то номер вершины полагается равным номеру их знаменателя;

б) если пара состоит из различных обобщенных номеров, то вершине присваивается очередной по порядку номер;

в) если очередная пара обобщенных номеров уже встречалась в рассматриваемом столбце, то соответствующей вершине присваивается тот же номер, что и ранее.

Правило 4 (минимизация дерева). Правила минимизации выполняются последовательно для вершин равного ранга, начиная с вершины первого ранга. При этом рассматриваются все возможные пары вершин равного ранга.

Пусть просмотрены все вершины до (i — 1)-го ранга включительно. При рассмотрении произвольной пары $(\mathbf{A}_1^{(i)}/a_1, \mathbf{A}_2^{(i)}/a_2)$, обобщенных номеров, соответствующих вершинам i-го ранга a_1 и a_2, могут возникнуть следующие ситуации:

а) $a_1 = a_2, \quad \mathbf{A}_1 = \mathbf{A}_2;$

б) $a_1 = a_2, \quad \mathbf{A}_1 \neq \mathbf{A}_2;$

в) $a_1 \neq a_2.$

В ситуации а) куст с начальной вершиной a_2 исключается из дерева вместе с вершиной a_2, а стрелка, соединяющая вершину a_2 с вершиной предыдущего ранга, направляется в начальную вершину a_1 оставшегося куста. В ситуации б) вершина a_2 соединяется стрелкой с вершиной a_1, после чего куст с начальной вершиной a_2 исключается из дерева с сохранением начальной вершины. В ситуации в) следует перейти к рассмотрению очередной пары.

На рис. 3 и 4 иллюстрируется применение правила 4.

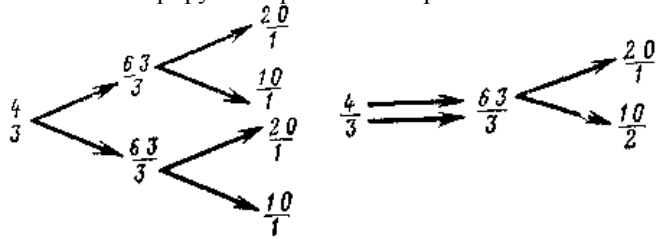


Рис. 3

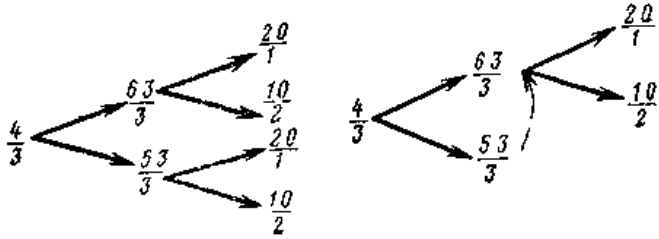


Рис. 4

Правило 5.

1. Все номера вершины стереть.
2. Если две вершины соседних рангов соединены двумя стрелками, то эти стрелки следует объединить.

После выполнения всех описанных правил может появиться возможность упрощения схемы за счет объединения арифметических операторов. Введем понятие *B'*-пересечеиия. *B*'-пересечение множеств B_1 и B_2 арифметических операторов не пусто и содержит ровно k операторов, если *последние по порядку* k операторов из B_1 и B_2 совпадают Возможность упрощения возникает, если в дереве алгоритма существует несколько групп арифметических операторов, для которых выходящие из них стрелки сходятся к одной вершине. Такие группы арифметических операторов назовем связанными.

Правило 6 (объединение операторов). Если *B*'-пересечение связанных групп арифметических операторов не пусто, то арифметические операторы, входящие в *B'*-пересечение, записываются левее общей для них вершины и одновременно исключаются из указанных групп.

На рис. 5 и 6 приведены примеры применения правила 6.

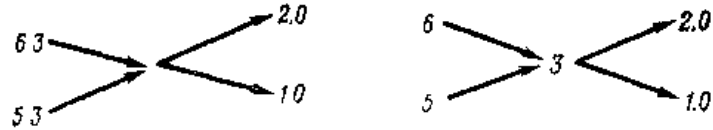


Рис.5

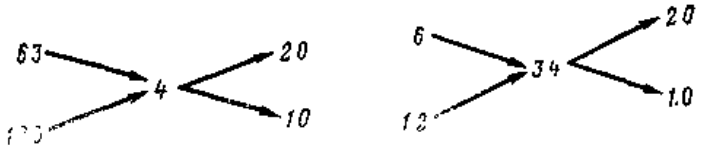


Рис. 6

Правило 7 (преобразование дерева в граф-схему алгоритма). Преобразование осуществляется для всех вершин равного ранга, начиная с первого.

1. Вершины, из которых выходят две стрелки, обозначаются кружками и внутри их записываются номера логических операторов, соответствующие рангу вершины. При этом верхняя стрелка помечается нулем (соответствует невыполнению проверяемого логического условия), а нижняя — единицей (соответствует выполнению проверяемого логического условия).
2. Пустые операторы, не имеющие выходов, последовательно уничтожаются вместе с входящими в них стрелками.
3. Анализируются образовавшиеся цепочки стрелок и операторов, соединяющие вершины младшего ранга с вершинами старших рангов. Каждая такая цепочка заменяется одной стрелкой, в середине которой в прямоугольнике записываются все непустые арифметические операторы цепочки.
4. Рассматриваются вершины, не имеющие выходов. Такие вершины назовем висящими. Пусть множество арифметических операторов, соответствующих висящей вершине, оканчивается арифметическим оператором $\mathbf{A}_i$. Если этот оператор уже встречался в дереве алгоритма, то в висящей вершине его следует уничтожить, а оставшиеся арифметические операторы стрелкой соединить с $\mathbf{A}_i$ в дереве алгоритма. Таким образом, в алгоритме возникает цикл.

Если же этот оператор не встречался в дереве алгоритма, то он означает останов. Группы арифметических операторов, заканчивающиеся одним и тем же оператором останова, соединяются стрелками в одну вершину, после чего к ним применяется правило 6 (рис. 7).

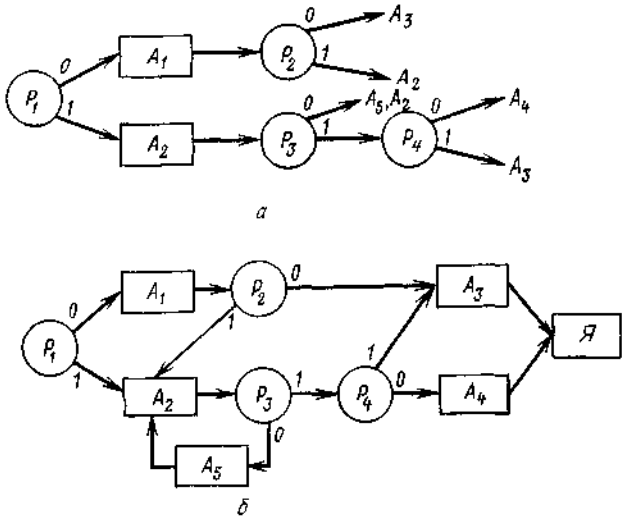


Рис. 7.

На этом построение граф-схемы алгоритма заканчивается.

Пример. В одноканальную систему распознавания поступает случайный поток требований с известным законом распределения длины интервалов времени между соседними требованиями. Время распознавани очередного требования также является случайной величиной с заданным законом распределения. Требования распознаются в порядке поступления Если очередное поступившее требование застает канал занятым, то оно покидает систему распознавания. Необходимо составить алгоритм, позволяющий получить статистические характеристики качества распознавания: вероятность отказа, долю распознаванных требований, среднее время занятости канала

Решение. Процесс функционирования системы будем рассматривать в интервале времени $[0, T]$ Сформируем последовательность логических операторов

$\mathbf{P}_1$—логический оператор, осуществляющий проверку попадания момента поступления очередного требования в интервал $[0, T]$ Этот оператор реализуется с помощью проверки выполнения неравенства $t_j < T$, где t_j—момент поступления j-й требования;

$\mathbf{P}_2$ — логический оператор, осуществляющий проверку возможности начала распознавания очередного требования. Оператор реализуется с помощью проверки неравенства

$$t_j > t_{j-1}^{\text{осв}},$$

где $t_{j-1}^{\text{осв}}$ — момент освобождения канала от распознавания $(j-1)$-го требования;

$\mathbf{P}_3$ — оператор, осуществляющий проверку выполнения заданного количества реализации M Оператор $\mathbf{P}_3$ реализуется с помощью проверки неравенства $m<M$.

Заполним столбцы $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ характеристической таблицы (рис. 8)

Номер строки	P_1	P_2	P_3	N_0	N_1	N_2	N_3
1	0	0	0	0	1	Z	2.Я
2	0	0	1	0	1	Z	0
3	0	1	0	0	1	Z	2.Я
4	0	1	1	0	1	Z	0
5	1	0	0	0	3	5.0	Z
6	1	0	1	0	3	5.0	Z
7	1	1	0	0	3	4.6.0	Z
8	1	1	1	0	3	4.6.0	Z

Рис 8.

Арифметические операторы будем формировать по мере надобности при заполнении таблицы В соответствии с правилами построения характеристической таблицы при заполнении каждого i-го столбца N_i необходимо ответить на два вопроса:

а) что можно сделать, когда известны фиксированные значения логических операторов $\mathbf{P}_1$, $\mathbf{P}_2$, . . ., $\mathbf{P}_i$;

б) чго нужно сделать, чтобы осуществить проверку выполнения оператора $\mathbf{P}_{i+1}$.

Для проверки выполнения оператора $\mathbf{P}_1$ необходимо для каждого требования вычислить момент его поступления t_3. В соответствии с этим введем арифметический оператор $\mathbf{A}_0$ вычисления величины t_i по заданному закону ее распределения. Так как оператор $\mathbf{A}_0$ предшествует $\mathbf{P}_1$, то его индекс 0 записывается во все строки столбца N_0.

В первых четырех строках характеристической таблицы $\mathbf{P_1 = 0}$. Это означает, что для очередного j-го требования условие $t_3 < T$ не выполняется. Следовательно, оно не будет выполняться и для всех последующих требований. При этом процесс распознавания для очередной реализации закончен. Тогда следует прибавить единицу к числу проделанных реализаций m и в зависимости от выполнения условия $\mathbf{P}_3$ прекратить или продолжить счет. Для этого введем оператор $\mathbf{A}_1$. Индекс 1 оператора $\mathbf{A}_i$ проставим в первых четырех строках столбца N_1 характеристической таблицы. Если $\mathbf{P}_1$=0, то проверка выполнения условия $\mathbf{P}_2$ теряет смысл. Поэтому в первых четырех строках столбца N_2 проставим символ пустого оператора **Z.**

Знания величины m достаточно для проверки выполнения условия $\mathbf{P}_3$ и в столбце N_3 можно указать арифметические операторы, которые следует выполнить в зависимости от значения $\mathbf{P}_3$. Если $\mathbf{P_3 = 0}$, то следует осуществить статистическую обработку результатов и выдать их на печать Выполнение этих функций возложим на оператор $\mathbf{A}_2$ После выполнения оператора $\mathbf{A}_3$ следует останов, осуществляемый оператором **Я**. Индекс 2 и символ **Я** проставим в первой и третьей строках столбца N_3, где $\mathbf{P_3 = 0}$.

Если $\mathbf{P_3 = 1}$, то следует начать очередную реализацию процесса распознавания. Для этого управление должно передаваться оператору $\mathbf{A}_0$. Его индекс 0 проставим во второй и четвертой строках столбца N_3. Перейдем к рассмотрению строк 5—8 таблицы, для которых $\mathbf{P_1 = 1}$.

Для проверки выполнения условий $\mathbf{P}_2$ необходимо для каждого требования знать, свободна ли система распознавания в момент поступления требования. Введем оператор $\mathbf{A}_3$, вычисляющий случайное время распознавания требования в соответствии с заданным законом распределения и момент $t_{j-1}^{\text{осв}}$ освобождения системы от распознавания предыдущего требования. Индекс 3 оператора $\mathbf{A}_3$ проставим в строках 5—8 столбца N_1.

Оператор $\mathbf{P}_2$ разветвляет процесс на два: в момент поступления очередного требования система свободна $(\mathbf{P_2 = 1})$ и занята $(\mathbf{P_2 = 0})$. В первом случае требование начинает распознаваться, во втором — получаем отказ. Введем операторы:

$\mathbf{A}_4$ — вычисления количества распознанных требований;

$\mathbf{A}_5$ — вычисления количества нераспознанных требований;

$\mathbf{A}_6$ — вычисления суммарного времени занятости системы.

Индексы 4 и 6 операторов $\mathbf{A}_4$ и $\mathbf{A}_6$ поставим в строках 7 и 8 столбца N_2, а индекс 5 оператора $\mathbf{A}_5$ — в строках 5 и 6 того же столбца. Кроме этого, в строках 5—8 необходимо записать индекс 0 оператора $\mathbf{A}_0$,

чтобы обеспечить передачу управления на продолжение данной реализации.

Очевидно, что если $\mathbf{P}_1 = 1$, то проверка выполнения условия $\mathbf{P}_3$ не должна производиться, так как реализация еще не закончена. Поэтому в строках 5—8 столбца N_3 проставим символ пустого оператора **Z.** Построим дерево алгоритма (рис. 9).

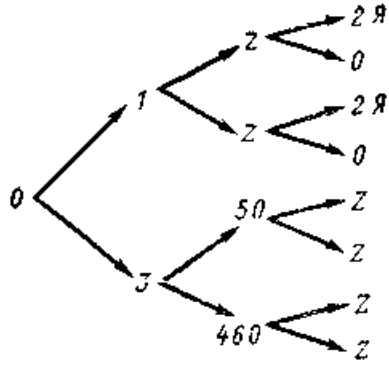


Рис. 9.

Поскольку *B*-пересечение для всех множеств операторов пусто, то правило 2 перемещения операторов неприменимо. Перенумеруем вершины дерева в соответствии с правилом 3 (рис. 10).

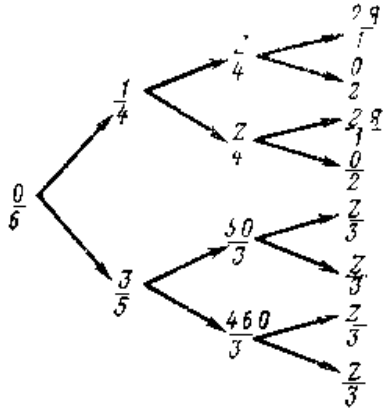


Рис. 10.

Осуществим эквивалентное преобразование полученного дерева. Результатом применения правила 4 является усеченное дерево, изображенное на рис. 11.

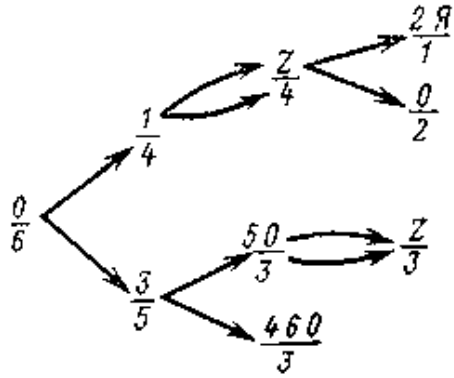


Рис. 11.

Применяя правило 5, получаем дерево, изображенное на рис. 12.

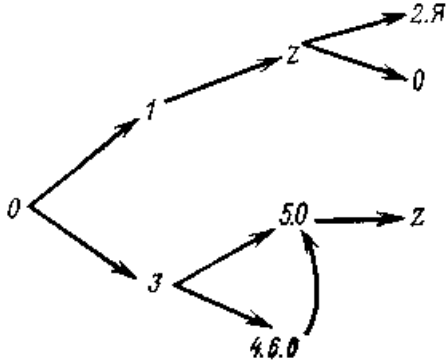


Рис. 12.

Поскольку в дереве отсутствуют связанные группы арифметических операторов, то правило 6 неприменимо. Используем правило 7 для преобразования минимизированного дерева в граф-схему алгоритма (рис. 13).

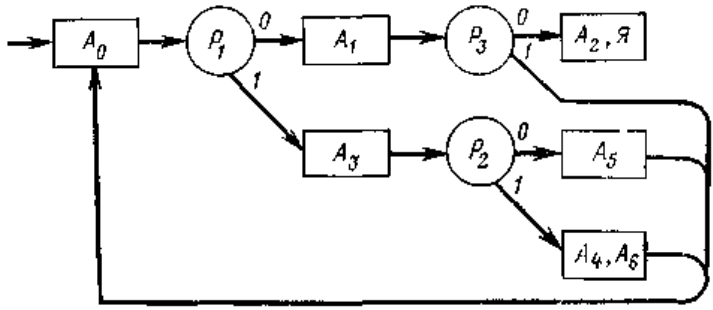


Рис. 13.

Составление граф-схемы алгоритма закончено.

10.4. Нечеткие алгоритмы

10.4.1. Проблема выполнения нечетких алгоритмов.

Новые для классической математики элементы: нечеткое множество, нечеткое отношение, нечеткая функция, нечеткая и лингвистическая переменные — приводят к чрезвычайно важному для процессов распознавания с нечетким описанием объекта распознавания понятию нечеткого алгоритма. Известно, что для распознавания того или иного объекта на ЭВМ требуется построить алгоритм его распознавания. Определим такой алгоритм как последовательность операторов, выполнение которых в соответствии с их семантикой и исходными данными приводит к получению некоторой информации о распознаваемом объекте (к решению той или иной задачи распознавания).

Любой оператор, содержащий в своей формулировке по крайней мере одну нечеткую или лингвистическую переменную, нечеткую функцию или нечеткое отношение, будем называть нечетким оператором. Дадим содержательное определение нечеткого алгоритма.

Последовательность выполняемых в соответствии с их семантикой нечетких операторов, приводящую к неполностью определенному (нечеткому) решению поставленной задачи распознавания, назовем нечетким алгоритмом. Впервые понятие нечеткого алгоритма было введено Заде. Нечеткие алгоритмы позволяют: применять лингвистическое описание для моделирования сложных процессов

распознавания; описывать одни нечеткие понятия другими, уже определенными; устанавливать нечеткие отношения между понятиями; прогнозировать поведение объекта управления; формировать множество альтернатив и производить формальное описание нечетких правил распознавания.

Непосредственное выполнение нечетких алгоритмов на ЭВМ невозможно, в связи с чем и возникает проблема разработки методов выполнения арифметических и логических нечетких операторов, процедур вычисления итога выполнения нечеткого алгоритма при ограниченном времени выполнения, процедур лингвистической аппроксимации.

10.4.2. Нечеткая и лингвистическая логики.

Нечетким логическим выражением называется формула, в состав которой входят нечеткие предикаты. Нечетким предикатом назовем отображение $P^F : X^n \to [0, 1]$, где $X = \{x\}$, n — любое натуральное число. Принадлежащее отрезку $[0, 1]$ число, которое предикат ставит в соответствие конкретному набору $(x_{k_1}, x_{k_2}, \ldots x_{k_n})$, где $k_i \in \overline{1, n}$, будем называть степенью принадлежностиописываемого данным набором высказывания к множеству истинных высказываний или коротко — степенью истинности. Интерпретация степени истинности, как и для функции принадлежности, может быть следующей: степень истинности — это вероятность того, что ЛРО назовет высказывание истинным.

Нечеткие логические выражения (или нечеткие формулы) отличаются от обычных наличием в их формулировках лингвистических и нечетких переменных и нечетких отношений (предикатов).

Приведем примеры.

1. Нечеткий предикат примерного равенства $AE(x, y)$: $x \approx y$, где $x, y \in R^1$.
2. Нечеткий предикат порядка $GT(C, H)$: $C > H$, где C, H — нечеткие числа.

Пусть μ_1 и μ_2 — степени истинности высказываний P_1^F и P_2^F (в которые превращаются нечеткие предикаты P_1^F и P_2^F после подстановки вместо переменных $x_{k_1} - x_{k_n}$ элементов множества X). Тогда степень истинности сложного высказывания, образованного из P_1^F и P_2^F с помощью операций дизъюнкции, конъюнкции и отрицания, может быть определена следующим образом:

$$\mu(P^F{}_1 \vee P^F{}_2) = \oplus(\mu_1, \mu_2); \qquad (1)$$

$$\mu(P_1^F \wedge P_2^F) = \odot(\mu_1, \mu_2); \quad (2)$$

$$\mu(\neg P_1^F) = 1 - \mu_1. \quad (3)$$

Здесь операции $\oplus$ и $\odot$ соответствуют операциям объединения и пересечения нечетких множеств. При минимаксной интерпретации функции принадлежности

$$\oplus(\mu_1, \mu_2) = \max\{\mu_1, \mu_2\}, \quad (4)$$

$$\odot(\mu_1, \mu_2) = \min\{\mu_1, \mu_2\}; \quad (5)$$

при вероятностной интерпретации

$$\oplus(\mu_1, \mu_2) = \mu_1 + \mu_2 - \mu_1\mu_2,$$

$$\odot(\mu_1, \mu_2) = \mu_1\mu_2.$$

Нечеткой называется логика, в которой степень истинности высказываний определяется выражениями (1) — (5).

Степень истинности более сложных высказываний можно определить, последовательно сворачивая их с учетом старшинства операций и применяя формулы (1)—(3). Задание нечетких предикатов производится путем специального опроса ЛРО или с помощью нечетких алгоритмов. В рамках нечеткой логики обобщен известный метод резолюций.

Рассмотрим условный нечеткий оператор общего вида

если У **то** Ф **иначе** Е, (6)

где У — некоторое нечеткое логическое выражение (условие); Ф и Е — группы нечетких операторов (в частности, в эти группы могут входить и обычные четкие операторы). Результат выполнения условного оператора (6) определим выражением

$$V(\textbf{если } У \textbf{ то } Ф \textbf{ иначе } Е) = \{\mu_У / V(Ф), (1-\mu_У)/V(Е)\}, \quad (7)$$

где $V(\xi)$ — результат выполнения оператора — ξ; $\mu_У$ степень истинности условия У

Таким образом, результатом выполнения условного нечеткого оператора является нечеткое множество результатов выполнения соответствующих групп нечетких операторов. Содержательно определение (7) означает, что начинают выполняться обе группы нечетких операторов Ф и Е. Однако каждая группа помечается своей меткой — степенью истинности.

При необходимости однозначно определить группу операторов продолжения можно воспользоваться двумя способами:

1. Задать порог степени истинности $\gamma_0 \in (0, 1)$. Вычислить $\mu_У$. Тогда

$$V(\textbf{если } У \textbf{ то } Ф \textbf{ иначе } Е) = \begin{cases} V(Ф), & \text{если } \mu_У \geqslant \gamma_0; \\ V(Е), & \text{если } \mu_У < \gamma_0. \end{cases} \quad (8)$$

Здесь следует обратить внимание на значение $\gamma_0 = 0{,}5$. Оно

относится к случаю, когда переход к выполнению группы нечетких операторов Ф осуществляется, если условие У более истинно, чем ложно. Увеличение γ_0 свыше 0,5 означает повышение требований к уровню определенности заключения об истинности У.

2. Вычислить μ_Y. Разыграть равномерную распределенную на интервале (0,1) случайную величину. Пусть полученное значение есть γ_0. Тогда искомый результат определяется выражением (8). Здесь μ_Y рассматривается как вероятность истинности условия У.

В общем случае степень истинности оказывается не числом из отрезка $[0, 1]$, а нечетким числом. Логика, в которой степени истинности являются нечеткими числами, называется лингвистической. Заметим, что иногда нечеткую логику называют многозначной (учитывая при этом именно определения (1) — (5)), а лингвистическую логику — нечеткой.

Лингвистическая степень истинности (ее значения — нечеткие числа) появляется, в частности, при оценке истинности одних нечетких высказываний относительно других. Пусть имеются высказывания $W : \langle X \text{ есть } F \rangle$ и $Q : \langle X \text{ есть } G \rangle$, где F и G — нечеткие подмножества U. Тогда истинность Q относительно W вычисляется как степень соответствия G и F:

$$T(W, Q) = \bigcup_{\tau \in [0, 1]} \mu_T(\tau)/\tau,$$

где

$$\mu_T(\tau) = \sup_{u : \tau = \mu_G(u)} \mu_F(u).$$

Одним из элементов лингвистической логики является правило истинностной модификации утверждения, которое заключается в следующем. Пусть известно, что лингвистическая степень истинности высказывания $Q : \langle X \text{ есть } G \rangle$ равна T. Тогда справедливо высказывание $W : \langle X \text{ есть } F \rangle$, где

$$F = \bigcup_{u \in U} \mu_F(u)/u, \qquad (9)$$

$$u = \mu_G^{-1}(\tau), \ \mu_F(u) = \mu_T(\mu_G(u)).$$

В лингвистической логике вводятся операции над лингвистическими истинностями, определяемые на основе (1) — (5) по принципу обобщения. Операции позволяют вычислять лингвистическую степень истинности составных логических выражений.

Отдельное направление работ в лингвистической логике связано с изучением построения выводов из нечетких посылок, включающих нечеткие кванторы типа «редко», «очень часто» и т. п. Предложено

решение задачи нахождения квантора φ_B по известным кванторам ρ_A и ρ_B в схеме вывода типа modus ponens

$$\frac{\rho_A A;\quad A \Rightarrow \rho_B B}{\varphi_B B}$$

и в более общих схемах вывода.

10.4.3. Выполнение нечетких алгоритмов.

Итог выполнения нечеткого алгоритма определим как нечеткое множество результатов

$$V = \bigcup_{i=1}^{I} \mu_i / V_i, \tag{10}$$

где V_i—*i*-й элемент итога (*i*-й результат); μ_i—степень истинности результата V_i. Элемент итога состоит из результатов проделанных операций и значений переменных, образовавшихся к моменту достижения конечной точки нечеткого алгоритма при определенном «прослеживании» текста последнего.

Опишем процедуру построения графа полного выполнения нечеткого алгоритма, где полным считается выполнение, при котором *Ш* достигает максимально возможного значения Участки нечеткого алгоритма, не содержащие условных нечетких операторов, будем изображать дугами, а условные нечеткие операторы— узлами графа. Начальной точке алгоритма соответствует узел, из которого выходит одна дуга и в который ни одна дуга не входит. Из (7) видно, что каждый условный нечеткий оператор образует при выполнении в качестве результата две ветви. Эти ветви представляют собой те участки нечеткого алгоритма, которые получают управление от данного нечеткого оператора или получат его впоследствии. Каждая из двух ветвей, выходящих из узлов, ограничивается либо новым условным оператором, который получит управление от последнего оператора ветви, либо концом алгоритма.

Если считать, что разметка дуг и узлов графа осуществляется текстами соответствующих участков (в частности, операторов) нечеткого алгоритма, то из изложенного ясно, что некоторые дуги графа полного выполнения нечеткого алгоритма, а также некоторые его узлы оказываются помеченными одинаково. Ясно также, что граф Г полного выполнения нечеткого алгоритма (рис. 1) является выходящим деревом.

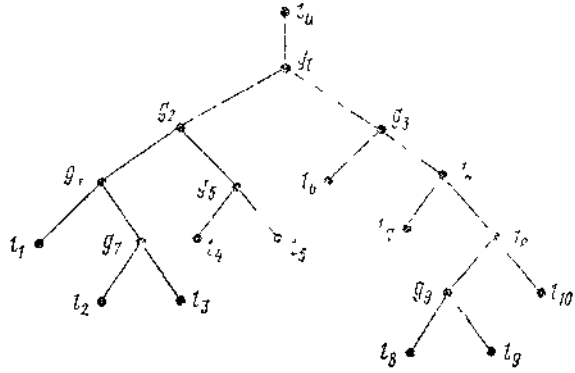


Рис. 1. Граф полного пыполнения нечеткого алгоритма

Корню графа l_0 соответствует начальная точка нечеткого алгоритма, множеству листьев $L=\{l_i\}_{i=1}$ — множество результатов выполнения нечеткого алгоритма, множеству узлов $G=\{g_k\}$ — множество условных операторов, множеству дуг $Y=\{y_j\}$ — множество участков нечеткого алгоритма, в которых отсутствуют нечеткие условные операторы.

Формирование всего множества V требует существенных затрат времени. При ограниченном времени может иметь место неполное (в смысле, определенном выше) выполнение нечеткого алгоритма. В связи с этим возникает задача разработки таких методов выполнения нечеткого алгоритма, которые позволяли бы последовательно отыскивать элементы множества V со все меньшей степенью истинности. Тогда, прекратив по истечении отведенного времени выполнение нечеткого алгоритма, получим нечеткое множество $V^* \subseteq V$, состоящее из результатов с большими степенями истинности типо сравнению со степенями истинности элементов множества $V \setminus V^*$.

Произведем новую разметку дуг графа Г. Дуге, выходящей из l_0 поставим в соответствие число 1, а каждым двум дугам, выходящим из узлов g_i: $(g_i g_{i_1})$ и $(g_i g_{i_2})$, — числа μ'_{g_i} и $(1-\mu'_{g_i})$, где μ'_{g_i} — степень истинности условия, входящего в тот условный нечеткий оператор, который соответствует узлу g_i.

Из корня l_0 в лист l_i ведет единственный путь, так как граф Г — выходящее дерево. Обозначим этот путь через D_i. Тогда

степень истинности листа l_i графа Г (а поскольку между множествами результатов $V_0 = \{V_i\}^{I}_{i=1}$ и листьев L имеет место взаимно однозначное соответствие, то и степень истинности μ_i результата V_i) определим следующим образом:

$$\mu_i = \mu_{l_i} = \bigodot_{g_t \in D_i} \mu_{g_t}, \tag{11}$$

где μ_{g_t} — метка дуги $(g_i g_j) \in D_i$.

Построим теперь процедуры выполнения нечетких алгоритмов в соответствии с введенным выше критерием для обеих интерпретаций операции $\odot$.

Минимаксная интерпретация операции $\odot$ $(\odot = \min)$. Пусть для некоторого узла g_ξ

$$\begin{aligned} \mu^{\max}_{g_\xi} &= \max\left\{\mu'_{g_\xi},\ 1-\mu'_{g_\xi}\right\}, \\ \mu^{\min}_{g_\xi} &= \min\left\{\mu'_{g_\xi},\ 1-\mu'_{g_\xi}\right\}. \end{aligned} \tag{12}$$

Опишем сначала процедуру нахождения результата выполнения нечеткого алгоритма с наибольшей степенью истинности, а затем процедуру поиска решений с меньшими степенями истинности.

Алгоритм 1. Поиск результата с наибольшей степенью истинности. При выполнении каждого условного нечеткого оператора необходимо, чтобы управление получала ветвь с большей степенью истинности. Результат будет обладать наибольшей (по сравнению с другими результатами) степенью истинности.

Алгоритм 2. Последовательный поиск результатов со все меньшими степенями истинности.

Назовем максимальным путем из узла g_i в лист путь $D^{\max}_{g_i} = g_i g_r \ldots l_j$, такой, что $\mu_{(g_i\, g_r)} = \mu^{\min}_{g_i}$, а для всех остальных дуг путии

$$(g_s g_t) \in D^{\max}_{g_i}$$

меет место

$$\mu_{(g_s\, g_t)} = \mu^{\max}_{g_s},$$

если только $g_s \notin L$.

Шаг 1. Образовать пустой список Ф. По алгоритму 1 найти путь D_a (приводящий из l_0 в лист l_a с максимальной степенью истинности) и получть соответствующий листу результат. Занести в Ф все узлы

$g_i \in D_a$ и те инцидентные им дуги, которые оказались помеченными величинами $\mu_{g_i}^{min}$.

Шаг 2. Упорядочить элементы списка Ф так, чтобы узлы и инцидентные им дуги оказались расположенными в ряд по убыванию значений меток дуг.

Шаг 3. Получить очередной (по убыванию степени истинности) результат, соответствующий листу l_g графа Г. Лист l_g лежит на максимальном пути $D_{g_j}^{max}$ из узла g_j, расположенного первым в списке Ф. Занести в список Ф все узлы $g_r \in D_{g_j}^{max}$ и те инцидентные им дуги, которые оказались помеченными величинами $\mu_{g_r}^{min}$. Удалить из списка Ф узел g_j и инцидентную ему дугу.

Шаг 4. Если список Ф пуст, то все результаты найдены. Если же он не пуст, то перейти к шагу 2.

Степень истинности результатов выполнения нечеткого алгоритма вычисляется по (11).

Вероятностная интерпретация операции $\odot (\odot = *)$. Пусть $D_i = l_0 g_1 g_2 \ldots g_k g_i$ — путь, ведущий из корня l_0 в узел g_i. Обозначим

$$\Psi_i = \prod_{g_i \in D_i} \mu_{g_i}. \qquad (13)$$

Пусть также из узла g_i выходят дуги $(g_i g_{i_1})$ и $(g_i g_{i_2})$, имеющие соответственно метки $\mu_1 = \mu'_{g_i}$ и $\mu_2 = 1 - \mu'_{g_i}$; L_{g_i} — множество листьев графа Г, достижимыхиз узла g_i. При выбранной интерпретации операции $\odot$ имеет место неравенство

$$(\forall l_v \in L_{g_i})\ \mu_{l_v} \leqslant \max \{\Psi_i \cdot \mu_1,\ \Psi_i \cdot \mu_2\}. \qquad (14)$$

Алгоритм 3. Последовательный поиск результатов с меньшими степенями истинности.

Шаг 1. Образовать пустой список Ф. Выполнить все действия по нечеткому алгоритму до достижения первого узла. Перейти к шагу 5.

Шаг 2. Вычислить степень истинности условия, соответствующего узлу. Занести в список Ф образовавшиеся дуги вместе с верхними оценками степени истинности листьев, достижимых при последующем продвижении из узла по каждой из дуг. Оценки вычисляются согласно (14).

Шаг 3. Упорядочить элементы списка Ф так, чтобы дуги и соответствующие им оценки оказались расположенными в ряд по убыванию значений оценок.

Шаг 4. Выполнить действия по нечеткому алгоритму, которые соответствуют дуге, оказавшейся первой в списке Ф, до достижения

очередного узла. Первый элемент списка Ф (дугу и оценку) из списка удалить.

Шаг 5. Вычислить Ψ_i для образовавшегося узла по формуле (13). Если этот узел — лист, то положить $\mu_{i_i} = \Psi_i$ и перейти к шагу 6. В противном случае перейти к шагу 2.

Шаг 6. Если список Ф пуст, то все результаты найдены. Если список Ф не пуст, то перейти к шагу 3.

10.4.4. Лингвистическая аппроксимация.

Пусть $\langle \beta, T, X, G, M \rangle$ — лингвистическая переменная, $T = \{T_i\}$ — ее базовое терм-множество, |T|=*n*. По определению лингвистической переменной каждое ее значение T_i является наименованием нечеткой переменной $<T_i, X, C_i>$, где $C_i = \bigcup_{x \in X} \mu^i_x / x$.

Обозначим носитель нечеткого множества C_s через S_i.

Пусть $<\rho, X, C_0>$ — нечеткая переменная, семантика которой определена нечетким множеством C_0, вычисленным в результате выполнения некоторого нечеткого алгоритма. Пусть также известно, что данная нечеткая переменная является значением лингвистической переменной β, а наименование нечеткой переменной неизвестно. Задача поиска наименования нечеткой переменной ρ в расширенном терм-множестве *T** лингвистической переменной β называется задачей лингвистической аппроксимации. Лингвистическая аппроксимация позволяет осуществить вербальное представление результатов работы лингвистических моделей: решений по управлению или ответов на вопросы ЛРО.

Рассмотрим более детально процесс образования расширенного терм-множества *T** с помощью процедур *G* и *M*. Предложено рассматривать элементы множества *T** как составные строки символов $\tau = \tau_1 \tau_2 \ldots \tau_i \ldots$, где элементы строк τ_i могут принадлежать к одному из четырех множеств: базовому терм-множеству *T,* множеству соединителей {и, или}, множеству модификаторов {не, очень, примерно,...}, множеству разделителей $\{\_, (,)\}$, где $\_$ — символ пробела.

Анализ вопросов применения составных термов для описания элементов задач распознавания объектов показывает, что, во-первых, набор модификаторов, позволяющих с помощью процедуры *G* описывать имеющие смысл в задачах распознавания расширения множества *T* до *T*,* не должен быть слишком большим. Во-вторых, длина цепочек τ (количество элементов в них) также не должна быть

большой. Наконец, мощность множества T, согласно известной гипотезе о емкости памяти ЛРО, должна удовлетворять неравенству $n \leq 7 \pm 2$. Достаточным является представление процедуры G в виде контекстно-свободной грамматики:

$$G = \langle V_N, V_T, U, \Pi \rangle, \tag{15}$$

где

$$V_N = \{A, B, C, D, E, F, H, U\};$$

$$V_T = \{\lambda, \_\_ \text{ и } \_\_, \_\_ \text{ или } \_\_, \text{ не } \_\_, \text{ очень } \_\_\} \cup T;$$

$$\Pi: U \to ABEHCDF, \quad U \to ABE;$$

$$H \to \_\_ \text{ и } \_\_, \qquad H \to \_\_ \text{ или } \_\_,$$

$$A \to \text{не } \_\_, \qquad A \to \lambda;$$

$$B \to \text{очень } \_\_, \qquad B \to \lambda;$$

$$C \to A, \qquad D \to B;$$

$$F \to E, \qquad E \to T_1;$$

$$E \to T_2, \qquad E \to T_3;$$

$$\cdots\cdots\cdots \qquad \cdots$$

$$E \to T_n.$$

При описании грамматики G (15) выбраны следующие обозначения: V_T—множество терминальных символов; V_N — множество нетерминальных символов; U — начальный символ; Π —множество правил подстановки; λ — пустой символ. Тогда $T^* = L(G)$, где $L(G)$— язык, порождаемый грамматикой (15). Язык $L(G)$ состоит из цепочек двух типов: η_1 и η_2. Цепочки типа η_1 имеют вид $m_1 m_2 T$, цепочки типа η_2 — вид $\eta'_1 \_\_$ и $\_\_ \eta''_1$, либо $\eta'_1 \_\_$ или $\_\_ \eta''_1$.

Здесь

$$m_1 \in \{\text{не } \_\_, \lambda\}, \quad T_i \in T, \quad m_2 \in \{\text{очень } \_\_, \lambda\},$$

η'_1 и η''_1 — цепочки типа η_1.

Семантика каждого из термов $T_i \in T^*$ задается процедурой M, которую в соответствии с указанными выше работами можно описать правилами (16), обозначив через T_i элемент множества T:

$$M(T_i) = C_i; \; M(\text{не } \_\_ \; T_i) = \bar{C}_i;$$

$$M(\text{очень } \_\_ \; T_i) = (C_i)^2; \; M(\text{не } \_\_ \text{ очень } \_\_ \; T_i) = \overline{(C_i)^2};$$

$$M(\eta'_1 \_\_ \text{ или } \_\_ \eta''_1) = M(\eta'_1) \cap M(\eta''_1); \tag{16}$$

$$M(\eta'_1 \_\_ \text{ или } \_\_ \eta''_1) = M(\eta'_1) \cup M(\eta''_1).$$

Построение нечеткого множества C_i для терма T_i, описываемое первой строкой системы (16), может осуществляться ЛРО с помощью диалоговой процедуры. Остальные уравнения описывают аналитические преобразования семантики базовых термов, задаваемой нечеткими множествами C_i.

Пусть T_i, T_j — два терма лингвистической переменной β; C_i, $C_j \in \tilde{C}$ — соответствующие им нечеткие множества; S_i, S_j — носители нечетких множеств C_i и C_j. Введем нечеткое отношение сходства S между нечеткими множествами, описывающими значения термов лингвистической переменной, с помощью функции принадлежности $\mu_S : C \times C \to [0,1]$. Отношение S должно удовлетворять следующим свойствам:

1) $\mu_S(C_i, C_i) = 1$ — рефлексивность;
2) $\mu_S(C_i, C_j) = \mu_S(C_j, C_i)$ — симметричность;
3) $\mu_S(C_i, \varnothing) = 0$ — отсутствие сходства между пустым и непустым множествами;
4) $S_i \cap S_j = \varnothing \Rightarrow \mu_S(C_i, C_j) = 0$ — отсутствие сходства между непересекающимися множествами.

Вид функции μ_S определяется соображениями, вкладываемыми в понятие сходства нечетких множеств. В частности, показана приемлемость следующего выражения:

$$\mu_S(C_i, C_j) = \frac{\int\limits_{S_i \cap S_j} \min\left\{\mu_x^i, \mu_x^j\right\} dx}{\int\limits_{S_i \cup S_j} \max\left\{\mu_x^i, \mu_x^j\right\} dx}, \tag{17}$$

или

$$\mu_S(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|},$$

где $|A|$ — мощность нечеткого множества A.

Заметим, что при таком определении справедливо высказывание

$$A = B \Leftrightarrow \mu_S(A, B) = 1.$$

Покажем это. Пусть A, B — нечеткие подмножества некоторого универсального множества X. Имеет место равенство

$$\mu_S(A, B) = 1 \Leftrightarrow |A \cap B| = |A \cup B|,$$

что возможно лишь при $A \cap B = A \cup B$, так как в остальных случаях $|A \cap B| < |A \cup B|$. По определению равенства нечетких множеств

$(\forall x \in X)\, A \cap B = A \cup B \Leftrightarrow \mu_{A \cap B}(x) = \mu_{A \cup B}(x)$. При минимаксной интерпретации операций пересечения и объединения нечетких множеств

$$\mu_{A \cap B}(x) = \min\{\mu_A(x),\ \mu_B(x)\},$$
$$\mu_{A \cup B}(x) = \max\{\mu_A(x),\ \mu_B(x)\}.$$

Следовательно, в этом случае высказывание $(\forall x \in X)\, \mu_{A \cap B}(x) = \mu_{A \cup B}(x)$ справедливо лишь при условии, что $(\forall x \in X)\, \mu_A(x) = \mu_B(x)$. Это по определению и означает $A = B$.

Поскольку множество всех возможных решений задачи лингвистической аппроксимации определяется множеством T и процедурой G, решением задачи назовем элемент $T_k \in T^*$, удовлетворяющий соотношению

$$C_k = \arg \max_{C_j \in C^*} \mu_S(C_0,\ C_j), \qquad (18)$$

где C^* — система нечетких множеств, соответствующих термам $T_j \in T^*$.

Таким образом, процедура лингвистической аппроксимации нечеткого множества включает следующие шаги:

Шаг 1. Образовать расширенное терм-множество той лингвистической переменной, значением которой является аппроксимируемое нечеткое множество, с помощью процедуры G (15).

Шаг 2. Определить семантику термов из расширенного терм-множества с помощью процедуры M (16).

Шаг 3. Решить задачу (18).

Примеры использования процедуры приведены на рис. 2.

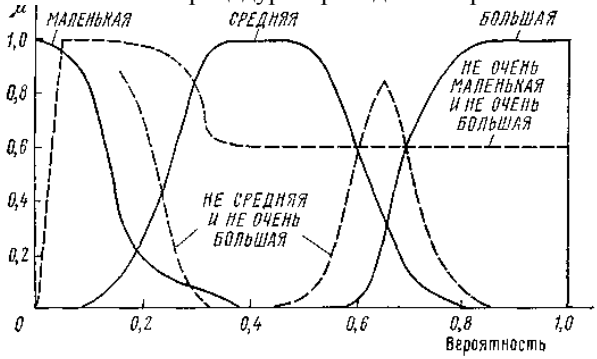


Рис. 2. Примеры лингвистической аппроксимации нечетких множеств: ——— базовые термы лингвистической переменной ВЕРОЯТНОСТЬ: – – – вычисленные нечеткие числа

Задача лингвистической аппроксимации может быть сформулирована и в следующей частной постановке. Пусть X_0 — четкое значение лингвистической переменной β. Требуется определить подходящее лингвистическое значение этой переменной. Поскольку множество T^* возможных значений лингвистической переменной β конечно, а число его элементов невелико, сформулированную задачу можно решить простым перебором по следующей схеме. Вычислить степени принадлежности μ_l четкого значения X_0 ко всем термам T_i из множества T^*. Лингвистическое значение переменной β будет иметь вид

$$\bigcup_{i \in \overline{1, |T^*|}} \mu_i / T_i.$$

При необходимости от нечеткого лингвистического значения переменной β можно перейти к четкому. Для этого достаточно решить задачу о выборе.

11. Методы оценки параметров распознаваемых объектов

Задача распознавания объекта может быть сформулирована как задача отыскания поверхности отклика объекта. В простейшем случае эта задача сводится к отысканию численных значений параметров объекта, характеризующих состояние объекта в прошлом, настоящем или будущем.

Исходным информационным материалом для решения этой задачи служат измерения параметров объекта или его реакций на различные входные воздействия. Заметим, что на практике нетривиальная по сути своей задача распознавания объекта осложняется дополнительно в силу ряда привходящих обстоятельств.

Прежде всего отметим, что часто с помощью непосредственных измерений отдельные переменные состояния точно определить нельзя. Обычно оказывается, что измерения, которые можно осуществить, представляют собой некоторые функции параметров объекта. Нужно также иметь в виду, что эти измерения содержат случайные ошибки; кроме того, сам объект может быть подвержен воздействию случайных возмущений.

При распознавании многих реальных объектов может быть использовано априорное представление о динамике функционирования этого объекта, позволяющее сформировать правдоподобную гипотезу о характере соотношений, связывающих параметры объекта (т. е. гипотезу о модели объекта). Так, например, часто можно считать, что

поведение объекта хорошо описывается системой дифференциальных уравнений определенной структуры или системой разностных уравнений, или, наконец, системой полиномов, параметры которых нужно определить.

В других случаях динамика объекта известна, но наличие случайных возмущений, воздействующих на объект, и случайных ошибок измерений не позволяют точно рассчитать ее параметры.

В любом из этих случаев возникает задача распознавания оптимальных в некотором смысле оценок параметров объекта с использованием совокупности сделанных измерений В зависимости от того, на какой момент времени: прошлый, настоящий или будущий — производится оценивание, соответствующую задачу называют сглаживанием, фильтрацией или прогнозированием.

Рассмотрим методы решения задачи оценки параметров. Эти методы нашли наиболее широкое применение в практических задачах распознавания. Для применения этих методов необходимо иметь в качестве априорной информации уравнение модели объекта, в котором могут быть неизвестны только параметры (некоторые числа). В зависимости от критерия распознаваня или применяемого вычислительного алгоритма эти методы получили различные наименования. В табл. 1 приведены краткие сведения о различных методах параметрического распознавания, рассмотренных в этом разделе.

Таблица 1

Наименование метода	Модель объекта, основные обозначения, предпосылки	Критерий	Необходимая априорная информация	Примечания
Метод наименьших квадратов (МНК)	Модель $z_n = \varphi(\mathbf{x}_n, \mathbf{A}, \xi_n)$, уравнение оценки состояния $\hat{z}_n = f(\mathbf{x}_n, \hat{\mathbf{A}})$	$\sum_{n=1}^{N} (z_n - \hat{z}_n)^2 \to \min_{\mathbf{A}}$	Уравнение оценки состояния	1. Для линейной модели с аддитивным шумом существует аналитическое решение 2. Возможно использование в рекуррентном и ретроспективном режимах оценивания параметров
Обобщенный метод наименьших квадратов (ОМНК)	Применяется, если аддитивная помеха ξ коррелирована во времени. Модель $z_n = \varphi(\mathbf{x}_n, \mathbf{A}) + \xi_n$, $\xi_n = \sum_{j=1}^{l} c^{(j)} \xi_{n-j} + \xi_n$, уравнение оценки состояния $\hat{z}_n = f(\mathbf{x}_n, \hat{\mathbf{A}}, \mathbf{C})$	$\sum_{n=1}^{N} (z_n - \hat{z}_n)^2 \to \min_{\mathbf{A},\mathbf{C}}$	1. Уравнение оценки состояния 2. Коэффициенты в уравнении шума (ковариационная матрица шума)	1. Существует аналитическое решение при известных коэффициентах $c^{(j)}$ ($j = 1, \ldots, l$) 2. При неизвестных коэффициентах $c^{(j)}$ аналитическое решение в общем случае отсутствует
Метод трансформации переменных (МТП)	Применяется, если неизвестны коэффициенты $c^{(j)}$. МТП – итерационная процедура последовательного решения двух систем линейных уравнений, в одной из которых	$\sum_{n=1}^{N} (z_n - \hat{z}_n)^2 \to \min_{\mathbf{A},\mathbf{C}}$	Уравнение оценки состояния	Используется для моделей, приведенных в ОМНК, и является итеративной процедурой решения системы

	принимаются известными коэффициенты $c^{(j)}$ и находятся $a^{(j)}$, в другой находятся $c^{(j)}$ при известных из предыдущей итерации $a^{(j)}$ и т.д.			нелинейных уравнений
Метод преобразования модели (МПМ)	В уравнении модели ненаблюдаемые переменные шума выражаются через прошлые наблюдения. Вводятся новые коэффициенты, представляющие собой нелинейную комбинацию старых. В результате получается линейная система с независимым шумом. Новые коэффициенты B (A, C) определяются аналитически МНК	$\sum_{n=1}^{N} (z_n - \hat{z}_n)^2 \to \min_{\mathbf{B}}$	Уравнение оценки состояния в форме $\hat{z}_n = f_1(\mathbf{x}_n, \mathbf{B})$	Используется для моделей, приведенных в ОМНК, и является процедурой замены системы нелинейных уравнений линейными путем замены переменных
Метод максимального правдоподобия (ММП)	Модель объекта $z_n = \varphi(\mathbf{x}_n, \mathbf{A}, \xi_n)$. Находится аналитическое выражение функции совместной плотности вероятности всех наблюдаемых переменных. В точке максимума этой функции по параметрам находятся искомые оценки	$p(xy) \to \max_{\mathbf{A}}$, где $p(x, y)$ – совместная плотность вероятности всех наблюдаемых переменных	Уравнение совместной плотности распределения вероятностей наблюдаемых переменных	Обычно максимизируется вместо функции плотности вероятности натуральный логарифм этой функции
Байесовские оценки (БО)	Уравнение модели задается в виде $\boldsymbol{\xi}_{n+1} = f_1(\boldsymbol{\theta}_n, \boldsymbol{\xi}_n, \boldsymbol{\epsilon}_{n+1})$; $\boldsymbol{\theta}_{n+1} = f_2(\boldsymbol{\theta}_n, \boldsymbol{\xi}_n, \boldsymbol{\epsilon}_{n+1})$, где $\boldsymbol{\theta}_n$ – ненаблюдаемые переменные	Минимум функции риска $R(c) = \int_{-\infty}^{\infty} c(\boldsymbol{\theta}) p(\boldsymbol{\theta}/\boldsymbol{\xi}) d\boldsymbol{\theta}$, где $c(\boldsymbol{\theta}) = \psi(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ – функция	1. Уравнение модели 2. Уравнение совместной плотности $p(\boldsymbol{\xi}/\boldsymbol{\theta})$.	1. Оцениваемый параметр $\boldsymbol{\theta}$ при БО представляется как случайный вектор $\boldsymbol{\theta}$, имеющий плотность распределения $p(\boldsymbol{\theta})$

Таблица 1 (продолжение)

Наименование метода	Модель объекта, основные обозначения, предпосылки	Критерий	Необходимая априорная информация	Примечания
	состояния (для задач идентификации оценки параметров); $\xi_n = \xi(x_n, \epsilon_n)$ – наблюдаемые переменные; ϵ_n – ненаблюдаемые шумы. БО строятся по рекуррентному алгоритму вида $\hat{\theta}_{n+1} = \varphi[\hat{\theta}_n, p_{n+1}(\theta/\xi_{n+1})]$, где $p_{n+1}(\theta/\xi_{n+1})$ – апостериорная условная плотность распределения, которая пересчитывается на каждом шаге по формулам Байеса через условную априорную плотность $p(\theta/\xi_n)$ и заданные безусловные плотности распределения	потерь. При квадратичной функции потерь вида $c(\theta) = (\theta - \hat{\theta})^T(\theta - \hat{\theta})$ минимизация $R(c)$ приводит к оценкам вида $\hat{\theta} = \int_{-\infty}^{\infty} \theta p(\theta/\xi)\, d\theta$, т.е. к условному математическому ожиданию θ	Значение $p(\xi/\theta)$ вычисляется по уравнению модели, если задано распределение шума ϵ 3. Плотность распределения $p(\theta)$	и условную плотность распределения $p(\theta/x, \epsilon)$ 2. Применение БО целесообразно при рекуррентной идентификации
Оценивание параметров с помощью фильтра Калмана (ФК)	Уравнение модели $\xi_{n+1} = A_0(\xi_n) + A_1(\xi_n)\theta_n + B\epsilon_{n+1}$; $\theta_{n+1} = \theta_n$ где θ – вектор неизвестных параметров; $\epsilon \sim N(0, 1)$. Формулы фильтра Калмана для этой модели позволяют рекуррентно пересчитывать на каждом шаге оптимальные оценки параметров θ, представляющих собой условные математические ожидания параметров, т.е. $\theta = M(\theta/\epsilon)$	Минимум функции риска при квадратичной функции потерь, как и для БО	1. Уравнение модели 2. Дисперсионная матрица шума B, вектор A_0 и матрица A_1	1. A_0, A_1, B могут зависеть от наблюдений, но эти зависимости должны быть известны. Применение целесообразно при рекуррентной идентификации 2. ФК является частным случаем БО для линейной модели

Метод инструментальной (вспомогательной) переменной (МИП)	**Уравнение модели** $z_n = \mathbf{X}_n\mathbf{A} + \epsilon_n$; $\operatorname{cov}(x_n^{(i)}, \epsilon_n) \neq 0$. Генерируются или выбираются инструментальная переменная Y, такая, что $\operatorname{cov}(\mathbf{Y}_n, \boldsymbol{\epsilon}_n) = 0$, $\operatorname{cov}(\mathbf{Y}_n, \mathbf{X}_n) \neq 0$. Оценки вычисляются из уравнения $\hat{\mathbf{A}} = (\mathbf{Y}_n\mathbf{X}_n)^{-1}(\mathbf{Y}_n\mathbf{Z}_n)$	Определяется условием $\operatorname{cov}(\mathbf{Y}_n, \epsilon_n) \to \min_{\mathbf{A}}$	1. Уравнение модели 2. Наличие инструментальной переменной	1. МИП при определенном способе выбора инструментальной переменной Y получил название метода сдвига или метода копирования 2. Возможно использование при рекуррентном и ретроспективном оценивании параметров
Оценивание параметров методом стохастической аппроксимации (СА)	Уравнение модели $z_n = f(\boldsymbol{\theta}, x_n) + \epsilon_n$ Условие возможности применения метода $M(\epsilon_n) = 0$, алгоритм идентификации $\hat{\boldsymbol{\theta}}_{n+1} = \hat{\boldsymbol{\theta}}_n + \gamma_{n+1} S_{n+1}$ S_n – функция невязки, γ_n – коэффициент усиления, $S_n = \frac{d}{d\boldsymbol{\theta}} \varphi[z_n - f(\boldsymbol{\theta}, x_n)] \Big\|_{\hat{\boldsymbol{\theta}}_{n-1}}$, где φ – функция потерь	Критерий произвольный. Обычно применяется $[z_n - f(\boldsymbol{\theta}, x_n)]^2 \to \min_{\boldsymbol{\theta}}$	1. Уравнение модели 2. Начальное значение оценки $\hat{\boldsymbol{\theta}}_0$	1. Только рекуррентные оценки 2. Размерность вектора S должна быть равна размерности $\hat{\boldsymbol{\theta}}$

Таблица 1 (окончание)

Наименование метода	Модель объекта, основные обозначения, предпосылки	Критерий	Необходимая априорная информация	Примечания
Метод осредненных невязок (МОН)	Уравнение модели $z_n = \varphi(x_n, \boldsymbol{\theta}^*) + \epsilon_n$. Уравнение оценки состояния $\hat{z}_n = \varphi(x_n, \hat{\boldsymbol{\theta}}_{n-1})$. Рекуррентное оценивание параметров по алгоритму $\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + \gamma_n S_n$. По сравнению с рекуррентным МНК и СА изменен способ формирования вектора невязки S_n. Вектор невязки построен на рекуррентно определяемых вторых моментах наблюдаемых переменных, члены матрицы γ_n, в отличие от СА и рекуррентного МНК, не стремятся к нулю с ростом выборки	$\sum_{n=1}^{N} (z_n - \hat{z})^2 \to \min_{\theta}$	Уравнение оценки состояния	Возможно использование при рекуррентном оценивании параметров. Метод разработан для линейных и нелинейных относительно оцениваемых параметров функций φ

11.1. Метод наименьших квадратов (МНК)

Метод наименьших квадратов (МНК) широко применяется для распознавания линейных и нелинейных систем.

Критерий МНК — минимизация суммы квадратов невязок между наблюдаемой выходной переменной *z* и ее оценкой, записанной в виде зависимости *z* от входной переменной и определяемых параметров.

МНК применяется в детерминированных постановках для аппроксимации экспериментальных результатов аналитическими выражениями. В вероятностной интерпретации для моделей с аддитивным шумом разработано теоретическое обоснование метода.

Пусть контролируемая переменная *y* неизвестным образом зависит от набора неизвестных параметров системы **C** и времени *t*. Пусть имеются основания предположить, что неизвестную зависимость $y = \varphi(\mathbf{C}, t)$ вполне удовлетворительным образом можно аппроксимировать полиномом определенной степени *k*. С этой целью введем

$$\tilde{y}(t) = \sum_{i=0}^{k} a_i t^i, \quad \begin{vmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{vmatrix} = \mathbf{A}. \tag{1}$$

Задача состоит в оценке неизвестных компонент вектора **A** таким образом, чтобы предсказываемое значение функции $\tilde{y}(t)$ в любой момент времени было в некотором смысле близко к истинному $y(t)$. В качестве исходного материала для решения поставленной задачи используется совокупность *n* измерений $Y_n = (y_1, y_2, \ldots, y_n)$, выполненных в некоторые определенные моменты времени $(t_1, t_2, \ldots, t_n)$. При этом

$$\mathbf{Y}_n = \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{vmatrix} = \begin{vmatrix} \varphi(\mathbf{C}, t_1) \\ \varphi(\mathbf{C}, t_2) \\ \vdots \\ \varphi(\mathbf{C}, t_n) \end{vmatrix}.$$

Рассчитаем предсказываемые значения функции отклика $\tilde{y}(t)$, задаваемые соотношением (1), в те же моменты времени. Совокупность этих значений образует вектор

$$\tilde{\mathbf{Y}}_n = \begin{vmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \cdot \\ \cdot \\ \cdot \\ \tilde{y}_n \end{vmatrix} = \begin{vmatrix} \sum_{i=0}^{k} a_i t_1^i \\ \sum_{i=0}^{k} a_i t_2^i \\ \cdot \\ \cdot \\ \cdot \\ \sum_{i=0}^{k} a_i t_n^i \end{vmatrix}. \qquad (2)$$

Качество аппроксимации функции отклика целесообразно оценивать суммой квадратов отклонений предсказываемых значений от истинных. В соответствии с этим введем функционал

$$\mathcal{J} = \frac{1}{2} \sum_{j=1}^{n} (y_j - \tilde{y}_j)^2 = \frac{1}{2} \sum_{j=1}^{n} \left(y_j - \sum_{i=0}^{k} a_i t_j^i \right)^2. \qquad (3)$$

Оптимальные в выбранном смысле оценки компонент вектора **A** отыщем, приравнивая нулю частные производные от функционала (3) по этим компонентам и решая полученную систему уравнений. При этом

$$\frac{\partial \mathcal{J}}{\partial a_i} = \sum_{j=1}^{n} \left(y_j - \sum_{i=0}^{k} a_i t_j^i \right) t_j^i, \quad i = 0, 1, 2, \ldots, k,$$

$$\sum_{j=1}^{n} \left(y_j - \sum_{i=0}^{k} a_i t_j^i \right) t_j^i = 0, \quad i = 0, 1, 2, \ldots, k. \qquad (4)$$

Систему уравнений (4) трансформируем следующим образом:

$$\sum_{j=1}^{n} \left(y_j - \sum_{i=0}^{k} a_i t_j^i \right) t_j^i = \sum_{j=1}^{n} y_j t_j^i - \sum_{i=0}^{k} \sum_{j=1}^{n} a_i t_j^{i+i} = 0,$$

$$i = 0, 1, 2, \ldots, k,$$

откуда

$$
\begin{aligned}
&a_0\sum_{j=1}^{n} t_j^0 + a_1\sum_{j=1}^{n} t_j^1 + \ldots + a_k\sum_{j=1}^{n} t_j^k = \sum_{j=1}^{n} y_j,\\
&a_0\sum_{j=1}^{n} t_j^1 + a_1\sum_{j=1}^{n} t_j^2 + \ldots + a_k\sum_{j=1}^{n} t_j^{k+1} = \sum_{j=1}^{n} y_j t_j,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_0\sum_{j=1}^{n} t_j^k + a_1\sum_{j=1}^{n} t_j^{k+1} + \ldots + a_k\sum_{j=1}^{n} t_j^{2k} = \sum_{j=1}^{n} y_1 t_j^k .
\end{aligned}
\qquad (5)
$$

Полученная система из (k+1)-го уравнения с таким же числом неизвестных $(a_0, a_1, \ldots, a_k)$ называется обычно «нормальной системой уравнений Гаусса». Решение этой системы дает оценки компонент вектора **A**, обеспечивающие минимум (в классе линейных оценок) среднего квадрата отклонения предсказываемых аппроксимирующим полиномом значений функции от наблюдаемых при эксперименте.

Все соотношения метода наименьших квадратов оказываются более компактными при использовании векторно-матричной формы записи. Введем матрицу $\mathbf{H}_n$ линейного преобразования значений параметров в значения функции, предсказываемые аппроксимирующим полиномом в заданные моменты времени $t_1, t_2, \ldots t_n$:

$$
\mathbf{H}_n = \begin{vmatrix}
1 & t_1 & t_1^2 & \ldots & t_1^k\\
1 & t_2 & t_2^2 & \ldots & t_2^k\\
\ldots & \ldots & \ldots & \ldots & \ldots\\
1 & t_n & t_n^2 & \ldots & t_n^k
\end{vmatrix}.
$$

Теперь соотношение (2) можно записать следующим образом:

$$\tilde{\mathbf{Y}}_n = \mathbf{H}_n\mathbf{A},$$

а минимизируемый функционал приобретает вид

$$
\mathcal{J} = \frac{1}{2}(\mathbf{Y}_n - \tilde{\mathbf{Y}}_n)^T(\mathbf{Y}_n - \tilde{\mathbf{Y}}_n) =
$$

$$
= \frac{1}{2}(\mathbf{Y}_n - \mathbf{H}_n\mathbf{A})^T(\mathbf{Y}_n - \mathbf{H}_n\mathbf{A}).
$$

Здесь T — символ транспонирования.
Далее имеем:

$$\frac{\partial\mathcal{J}}{\partial\mathbf{A}^T} = -\mathbf{H}_n^T(\mathbf{Y}_n - \mathbf{H}_n\mathbf{A})$$

и после введения вектора $\hat{\mathbf{A}}$ оптимальных оценок

$$-\mathbf{H}_n^T(\mathbf{Y}_n - \mathbf{H}_n\hat{\mathbf{A}}) = 0.$$

Отсюда

$$\mathbf{H}_n^T\mathbf{H}_n\hat{\mathbf{A}} = \mathbf{H}_n^T\mathbf{Y}_n$$

и, наконец,

$$\hat{\mathbf{A}} = (\mathbf{H}_n^T\mathbf{H}_n)^{-1}\mathbf{H}_n^T\mathbf{Y}_n. \qquad (6)$$

Таким образом, при использовании для оценки параметров системы метода наименьших квадратов необходима информация о значениях функции отклика (контролируемой переменной) на некотором множестве моментов времени и гипотеза о характере соотношений, аппроксимирующих зависимость функции отклика от параметров системы. Типичным примером такой задачи является задача сглаживания параметров некоторой траектории, аппроксимируемой полиномом заданной степени.

Заметим, что качество оценки параметров существенно зависит от количества используемых при обработке измерений. Это обстоятельство проявляется особенно наглядно, когда структура функции отклика априорно известна. Пусть, например, закон изменения контролируемой переменной $\tilde{y}(t)$ имеет вид

$$\tilde{y}(t) = a_0 + a_1 t + \ldots + a_k t^k,$$

так что истинные значения этой переменной на наборе моментов времени $t_1, t_2, \ldots, t_n$ образуют вектор

$$\tilde{\mathbf{Y}}_n = \begin{vmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \cdot \\ \cdot \\ \tilde{y}_n \end{vmatrix}.$$

Будем считать, что в процессе измерения на истинные значения контролируемой переменной аддитивно накладывается стационарный белый гауссов шум $v(t)$ с нулевым математическим ожиданием и дисперсией σ^2. Тогда

$$y(t) = \tilde{y}(t) + v(t),$$

где $v(t)$—случайная ошибка измерения, причем

$$\overline{v(t)} = 0, \qquad \overline{[v(t)]^2} = \sigma^2.$$

Введем вектор $\mathbf{V}_n$ значений случайных ошибок измерения в моменты времени $t_1, t_2, \ldots, t_n$

$$\mathbf{V}_n = \begin{vmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_n \end{vmatrix}.$$

При этом

$$\mathbf{Y}_n = \tilde{\mathbf{Y}}_n + \mathbf{V}_n = \mathbf{H}_n\mathbf{A} + \mathbf{V}_n,$$

откуда

$$\mathbf{H}_n\mathbf{A} = \mathbf{Y}_n - \mathbf{V}_n. \qquad (7)$$

Умножая обе части соотношения (7) на $(\mathbf{H}_n^T\mathbf{H}_n)^{-1}\mathbf{H}_n^T$, получаем

$$(\mathbf{H}_n^T\mathbf{H}_n)^{-1}\ (\mathbf{H}_n^T\mathbf{H}_n)\,\mathbf{A} = (\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\,\mathbf{Y}_n -$$

$$-\,(\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\mathbf{V}_n,$$

или, учитывая (6),

$$\mathbf{A} = \hat{\mathbf{A}} - (\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\mathbf{V}_n =$$

$$= \hat{\mathbf{A}} - (\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\,(\mathbf{Y}_n - \mathbf{H}_n\mathbf{A}).$$

Вектор

$$\hat{\mathbf{A}} - \mathbf{A} = (\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\,\mathbf{V}_n$$

имеет смысл вектора ошибок оценки параметров. Ковариационная матрица ошибок оценок параметров имеет вид

$$\Psi_n = \overline{[(\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\mathbf{V}_n]\,[(\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\mathbf{V}_n]^T} =$$

$$= \overline{(\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\,\mathbf{V}_n\mathbf{V}_n^T\mathbf{H}_n\,(\mathbf{H}_n^T\mathbf{H}_n)^{-1}} =$$

$$= (\mathbf{H}_n^T\mathbf{H}_n)^{-1}\,\mathbf{H}_n^T\,\mathbf{R}_n\mathbf{H}_n\,(\mathbf{H}_n^T\,\mathbf{H}_n)^{-1} =$$

$$= \mathbf{H}_n^{-1}\,(\mathbf{H}_n^T)^{-1}\,\mathbf{H}_n^T\,\mathbf{R}_n\mathbf{H}_n\mathbf{H}_n^{-1}\,(\mathbf{H}_n^T)^{-1} =$$

$$= \mathbf{H}_n^{-1}\mathbf{R}_n(\mathbf{H}_n^T)^{-1} = (\mathbf{H}_n^T\,\mathbf{H}_n)^{-1}\,\mathbf{R}_n. \qquad (8)$$

Здесь $\mathbf{R}_n = \overline{\mathbf{V}_n\mathbf{V}^T}_n$—корреляционная матрица ошибок измерений, причем в предположении, что измерения некоррелированы:

$$\mathbf{R}_n = \sigma^2\mathbf{I}_n,$$

где $\mathbf{I}_n$—единичная матрица размерностью $n \times n$.

С учетом этого

$$(\mathbf{H}_n^T\mathbf{H}_n)^{-1} = \Psi_n\mathbf{R}_n^{-1} = \Psi_n/\sigma^2. \qquad (9)$$

Рассмотрим теперь простейший частный случай, когда контролируемая переменная $\tilde{y}(t)$ является линейной функцией параметров, т. е. $\tilde{y}(t) = a_0 + a_1 t$, и измерения выполняются через равные промежутки времени с периодом T_0. Тогда

$$(t_1, t_2, \ldots, t_n) = (T_0, 2T_0, \ldots, nT_0),$$

$$\mathbf{H}_n = \begin{vmatrix} 1 & T_0 \\ 1 & 2T_0 \\ 1 & 3T_0 \\ \cdot & \cdot \\ 1 & nT_0 \end{vmatrix}.$$

При этом

$$\mathbf{H}_n^T \mathbf{H}_n = \begin{vmatrix} 1 & 1 & \ldots & 1 \\ T_0 & 2T_0 & \ldots & nT_0 \end{vmatrix} \begin{vmatrix} 1 & T_0 \\ 1 & 2T_0 \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & nT_0 \end{vmatrix} =$$

$$= \begin{vmatrix} n & T_0 \sum_{i=1}^{n} i \\ T_0 \sum_{i=1}^{n} i & T_0^2 \sum_{i=1}^{n} i^2 \end{vmatrix} =$$

$$= \begin{vmatrix} n & \frac{n(n+1)}{2} T_0 \\ \frac{n(n+1)}{2} T_0 & \frac{T_0^2 n(n+1)(2n+1)}{6} \end{vmatrix}.$$

Как известно, элементы $(\mathbf{B}^{-1})_{ij}$ обратной матрицы $\mathbf{B}^{-1}$ по отношению к заданной $\mathbf{B}$ вычисляются по формуле

$$(\mathbf{B}^{-1})_{ij} = \frac{1}{\det \mathbf{B}} C_{ij},$$

где C_{ij} — алгебраическое дополнение элемента ij матрицы $\mathbf{B}$.

Поэтому после простых вычислений имеем

$$\det(\mathbf{H}_n^T\mathbf{H}_n) = T_0^2\frac{n^2(n^2-1)}{12},$$

$$(\mathbf{H}_n^T\mathbf{H}_n)^{-1} = \begin{vmatrix} \frac{2(2n+1)}{n(n-1)} & -\frac{6}{n(n-1)T_0} \\ -\frac{6}{n(n-1)T_0} & \frac{12}{n(n^2-1)T_0^2} \end{vmatrix}. \qquad (10)$$

Подставляя (10) в (8), получаем корреляционную матрицу ошибок оценок параметров линейной функции по методу наименьших квадратов для случая равноточных и равнодискретных измерений ее значений:

$$\Psi_n = \sigma^2 \begin{vmatrix} \frac{2(2n+1)}{n(n-1)} & -\frac{6}{n(n-1)T_0} \\ -\frac{6}{n(n-1)T_0} & \frac{12}{n(n^2-1)T_0^2} \end{vmatrix}.$$

Анализ элементов корреляционной матрицы Ψ_n показывает, что дисперсия ошибки оценки параметра a_0 убывает пропорционально количеству сделанных измерений, а дисперсия ошибки оценки параметра a_1 — пропорционально кубу числа измерений.

11.2. Метод МНК в вероятностной интерпретации

Пусть уравнение модели объекта имеет вид

$$z_n = \varphi(\mathbf{x}_n, \mathbf{A}, \xi_n),$$

где $\mathbf{x}_n$ — вектор наблюдаемых входных переменных; n — номер наблюдения; $\mathbf{A}$ - вектор неизвестных параметров; ξ_n - вектор ненаблюдаемых помех.

Имеется N наблюдений x_n и z_n. Минимизация по $\mathbf{A}$ выражения

$$\sum_{n=1}^{N}[z_n - \varphi(\mathbf{x}_n, \mathbf{A}, \xi_n)]^2$$

приводит к оценкам метода наименьших квадратов вектора $\mathbf{A}$. Очевидно, на данной реализации для записанного уравнения полученные оценки $\mathbf{A}$ дают наименьшее среднеквадратическое отклонение (СКО) реализации $\hat{z}_n$, полученной по модели от наблюдаемой реализации z_n, $n = 1,...,N$.

Оценки МНК в представленном виде обычно не могут быть найдены, (поскольку в выражение суммы входят ненаблюдаемые помехи. Термин МНК обычно относят к модели с аддитивным шумом вида

$$z_n = f(\mathbf{x}_n, \mathbf{A}) + \xi_n. \qquad (1)$$

Оценками МНК называются значения вектора $\hat{\mathbf{A}} = [\hat{a}^{(i)}]$, $i = 1, ..., m$, (полученные при минимизации выражения

$$Q = \sum_{n=1}^{N} [z_n - f(\mathbf{x}_n, \mathbf{A})]^2. \qquad (2)$$

В случае, если шумы имеют нормальное распределение и

$$\operatorname{cov}[f(\mathbf{x}_n, \mathbf{A}), \xi_n] = 0, \qquad (3)$$

полученные оценки вектора **A** при $n \to \infty$ не смещены, т.е. $M\hat{\mathbf{A}} = \mathbf{A}$ Если размерность вектора параметров **A** равна *m,* то чтобы найти оценки $\hat{\mathbf{A}}$, необходимо решить систему из *m* в общем случае нелинейных уравнений вида

$$S = dQ/d\mathbf{A} = 0. \qquad (4)$$

В ретроспективном режиме оценивания параметров функция *Q* на конкретной выборке является детерминированной функцией парамет-|ров. Корни функции *S* являются значениями оценок $\hat{\mathbf{A}}$, в которых функция *Q* минимальна на собранной выборке $z_n, \mathbf{x}_n, n = 1, ..., N$.

Для нахождения $\hat{\mathbf{A}}$ используются различные численные методы решения экстремальных задач или итерационные процедуры решения системы нелинейных уравнений. На практике часто применяются различные методы линеаризации. Приведем здесь итерационную процедуру с линеаризацией функции разложением ее в ряд Тейлора по оцениваемым параметрам.

Для минимизации критерия (2) представим $f(\mathbf{x}_n, \mathbf{A})$ в виде

$$f(\mathbf{x}_n, \mathbf{A}) \approx f(\mathbf{x}_n, \hat{\mathbf{A}}_{(0)}) + \sum_{i=1}^{m} (a^{(i)}_{(1)} - a^{(i)}_{(0)}) \left.\frac{\partial f}{\partial a^{(i)}}\right|_{\hat{\mathbf{A}}_{(0)}}, \qquad (5)$$

где $\hat{\mathbf{A}}_{(0)}$ — заданное исходное значение оценки вектора $\mathbf{A} = [a^{(1)}, a^{(2)}, ... a^{(m)}]^{\mathrm{T}}$

После подстановки (5) в (2) получим

$$Q(\mathbf{A}) = \sum_{n=1}^{N} [z_n - f(\mathbf{x}_n, \hat{\mathbf{A}}_{(0)}) - \sum_{i=1}^{m} (a^{(i)} - a^{(i)}_{(0)}) \left.\frac{\partial f}{\partial a^{(i)}}\right|_{\hat{\mathbf{A}}_{(0)}}]^2 =$$

$$= (\mathbf{R} - \mathbf{P}\boldsymbol{\delta}_{(1)})^{\mathrm{T}} (\mathbf{R} - \mathbf{P}\boldsymbol{\delta}_{(1)}),$$

где **R**—*N*-мерный вектор с элементами $r_k = z_k - f(\mathbf{x}_k, \hat{\mathbf{A}}_0)$, $k = 1, ..., N$; **P** — матрица размером $N \times m$ с элементами $[p^{(ij)}] = \partial f(x_i, \mathbf{A})/\partial a^{(j)} |_{\hat{\mathbf{A}}_{(0)}}$;

$\delta_{(1)}$ - m-мерный неизвестный вектор приращений параметров, $\delta_{(1)} = \hat{\mathbf{A}}_{(1)} - \hat{\mathbf{A}}_{(0)}$.

Минимизация $Q(\mathbf{A})$ по $\mathbf{A}$ дает оценку приращения вектора параметров в виде

$$\boldsymbol{\delta}_{(1)} = (\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{R}.$$

Следующая итерация проводится таким же образом в точке

$$\hat{\mathbf{A}}_{(1)} = \hat{\mathbf{A}}_{(0)} + \boldsymbol{\delta}_{(1)};$$

получается следующее приращение $\boldsymbol{\delta}_{(2)}$ и т.д.

По существу, предположенная процедура является одной из модификаций алгоритма Ньютона поиска корней нелинейных уравнений.

Если функция f в (1) линейно зависит от параметров, то система уравнений (4) имеет аналитическое решение. Пусть модель объекта записана в форме

$$z_n = \sum_{i=1}^{r} a^{(i)} z_{n-i} + \sum_{i=1}^{l} b^{(i)} x_{n-i} + \xi_n, \qquad (6)$$

где $a^{(i)}\ (i = 1, ..., r), b^{(i)}\ (i = 1, ..., l)$ — неизвестные параметры; ξ_n — ненаблюдаемый шум; x_{n-i}, z_{n-i} - наблюдаемые переменные (или их известные функции).

Для определенности полагаем $r \geqslant l$, $r + l = m$.

Модель объекта (6) может быть записана в виде соотношения между массивами собранных данных в виде векторно-матричного уравнения

$$\mathbf{Z} = \mathbf{X}\mathbf{A} + \boldsymbol{\Xi}, \qquad (7)$$

где

$$\mathbf{X} = \begin{bmatrix} z_{N+r} & \dots & z_N, & x_{N+r} & \dots & x_N \\ \vdots & & \vdots & \vdots & & \vdots \\ z_r & \dots & z_0, & x_r & \dots & x_0 \end{bmatrix} \text{ – матрица размером } N \times m;$$

$\mathbf{Z} = [z_{N+r+1}, z_{N+r}, ..., z_{r+1}]^{\mathrm{T}}$ – вектор размерности N;

$\boldsymbol{\Xi} = [\xi_{N+r+1}, \xi_{N+r}, ..., \xi_{r+1}]^{\mathrm{T}}$ – вектор размерности N;

$\mathbf{A} = [a^{(1)}, ..., a^{(r)}, b^{(1)}, ..., b^{(l)}]^{\mathrm{T}}$ – вектор размерности m.

Минимизация суммы N квадратов невязок по параметрам, т.е. минимизация квадратичной формы

$$Q = (\mathbf{Z} - \mathbf{XA})^{\mathrm{T}} (\mathbf{Z} - \mathbf{XA}), \tag{8}$$

приводит к оценкам неизвестных параметров, вычисляемым из выражения

$$\hat{\mathbf{A}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} (\mathbf{X}^{\mathrm{T}}\mathbf{Z}). \tag{9}$$

Если в уравнении (6) шум ξ_n, имеет гауссовское распределение, не коррелирован с членами в правой части (6) и $\mathrm{cov}(\xi_n, \xi_k) = 0$ при $n \neq k$, то выражение (9) для оценок может быть получено из условий

$$M(\hat{\mathbf{A}}) = \mathbf{A}; \quad D_{\hat{\mathbf{A}}} = \mathrm{diag}\,[(\mathbf{A} - \hat{\mathbf{A}})(\mathbf{A} - \hat{\mathbf{A}})^{\mathrm{T}}] \to \min. \tag{10}$$

Следовательно, оценки параметров МНК для гауссовских шумов при выполнении условия (3) не смещены ($M(\hat{\mathbf{A}}) = \mathbf{A}$) и эффективны ($D_{\hat{\mathbf{A}}}$ — минимальна), эти оценки состоятельны, поскольку $D_{\hat{\mathbf{A}}} \xrightarrow[N \to \infty]{} 0$.

Дисперсии оценок могут быть найдены с помощью матрицы $\mathbf{R} = M[\mathbf{X}^{\mathrm{T}}\mathbf{X}]$. Матрица $\mathbf{R}$ размером $m \times m$ называется *матрицей вторых моментов,* или *корреляционной матрицей.* Она состоит из корреляционных моментов пар случайных величин z_n, x_n. Матрица ковариаций оценок параметров вычисляется по формуле

$$\mathrm{cov}(\hat{\mathbf{A}}) = M[(\mathbf{A} - \hat{\mathbf{A}})(\mathbf{A} - \hat{\mathbf{A}})^{\mathrm{T}}] = \sigma^2 \mathbf{R}^{-1}. \tag{11}$$

На диагонали полученной ковариационной матрицы расположены дисперсии оценок параметров.

Оптимальная с точки зрения минимума среднеквадратической ошибки оценка выходной переменной z_n записывается в виде

$$z_n = \sum_{i=1}^{r} \hat{a}^{(i)} z_{n-i} + \sum_{i=1}^{l} \hat{b}^{(i)} x_{n-i}.$$

Такая оценка, как и оценки параметров, не смещена, с ростом выборки дисперсия этой оценки стремится к дисперсии шума σ^2.

Рекуррентная форма оценок МНК для модели (6) записывается в виде

$$\hat{\mathbf{A}}_{n+1} = \mathbf{A}_n + \mathbf{P}_{n+1}\mathbf{x}_{n+1}(z_{n+1} - \mathbf{x}_{n+1}^{\mathrm{T}}\hat{\mathbf{A}}_n), \tag{12}$$

где $\mathbf{P}_{n+1} = \mathbf{P}_n - \gamma_n \mathbf{P}_n \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\mathrm{T}} \mathbf{P}_n^{\mathrm{T}}$ — матрица размером $m \times m$; $\gamma_n = (\mathbf{x}_{n+1}^{\mathrm{T}} \mathbf{P}_n \mathbf{x}_{n+1} + 1)^{-1}$ — скаляр; $\mathbf{x}_{n+1}$ — вектор всех входных переменных на $(n+1)$-м шаге, транспонированная $(n+1)$-я строка матрицы $\mathbf{X}$; z_{n+1} - замер выходной переменной.

Поскольку пересчет матрицы $\mathbf{P}_n = \mathbf{R}_n^{-1} = [\mathbf{X}_n^{\mathrm{T}} \mathbf{X}_n]^{-1}$ на каждом шаге производится по формуле (12), обращения матрицы не требуется.

11.3. Обобщенный метод наименьших квадратов (ОМНК)

Обычно для процессов распознавания характерна коррелированность во времени шумов, действующих на объект. В этой условиях несправедливо утверждение о некоррелированности шума и регрессоров (наблюдаемых координат) в правой части уравнения (6). Коррелированность шума при использовании МНК, т.е. при минимизации формы (8), вызывает смещение оценок параметров, увеличение дисперсии этих оценок. Ухудшение оценок параметров приводит к ухудшению свойств оценок переменных состояния $\hat{z}_n$.

Для получения несмещенных оценок используется *обобщенный метод наименьших квадратов* (ОМНК).

Для использования ОМНК необходима модель шума. Линейные модели шума принято записывать в форме *авторегрессии* (АР)

$$\xi_n = \sum_{i=1}^{G} g^{(i)} \xi_{n-i} + \sigma \epsilon_n \qquad (1)$$

или *скользящ его среднего* (СС)

$$\xi_n = \sum_{i=0}^{G} \sigma^{(i)} \epsilon_{n-i}; \qquad (2)$$

в обоих случаях белый шум $\epsilon_n \sim N(0, 1)$, где N - обозначение нормального распределения, причем первая цифра в скобках — значение математического ожидания, а вторая - среднего квадратического отклонения в этом распределении.

Если коэффициенты $\sigma^{(i)}$ в (2) или $g^{(i)}$ в (1) известны, оценки ОМНК для модели (7 п.11.2) находятся из выражения

$$\hat{\mathbf{A}} = (\mathbf{X}^{\mathrm{T}} \Omega^{-1} \mathbf{X})^{-1} \mathbf{X}^{\mathrm{T}} \Omega^{-1} \mathbf{Z}, \qquad (3)$$

где $\Omega = (\Xi \Xi^{\mathrm{T}})$ - ковариационная матрица шума Ξ, которая может быть найдена, если известны коэффициенты $g^{(i)}$ или $\sigma^{(i)}$, $i = 1, ..., G$.

Оценки (3) получаются из условия минимизации критерия

$$Q = \sum_{n=1}^{N} (z_n - \hat{z}_n)^2, \tag{4}$$

где $\hat{z}_n = \sum_{i=1}^{r} \hat{a}^{(i)} z_{n-i} + \sum_{i=1}^{l} \hat{b}^{(i)} x_n + \sum_{i=1}^{G} g^{(i)} \hat{\xi}_{n-i}$ для модели шума (1);

$\hat{z}_n = \sum_{i=1}^{r} \hat{a}^{(i)} z_{n-i} + \sum_{i=1}^{l} \hat{b}^{(i)} x_n + \sum_{i=1}^{G} \sigma^{(i)} \hat{\epsilon}_{n-i}$ для модели шума (2).

Полученные из 3) оценки ОМНК не смещены и имеют минимальную дисперсию.

Если коэффициенты $g^{(i)}$ и $\sigma^{(i)}$ неизвестны, что чаще всего имеет мест на практике, применение критерия (4) приводит к системе нелиейных уравнений, полученных из условий

$$\left.\begin{aligned} \partial Q/\partial \mathbf{A} &= 0 \\ \partial Q/\partial \mathbf{G} &= 0. \end{aligned}\right\} \tag{5}$$

Второе уравнение системы относится к записи в форме АР. Для записив форме СС второе уравнение системы (5) записывается как $\partial Q/\partial \sigma = 0$.

Для определения оценок параметров можно применить любые методы нахождения корней системы нелинейных уравнений, используемые в нычислительной математике. Наиболее удобно использование методов, учитывающих билинейный характер уравнений (5). Среди этих методов в практических расчетах хорошо зарекомендовали себя метод трансформации переменных и метод преобразования модели. Оба метода хорошо приспособлены для авторегрессионной формы (1) записи модели шума.

Метод трансформации переменных. Метод трансформации переменных (МТП) представляет собой итерационную процедуру решения двух систем линейных уравнений и используется в ретроспективном режиме. Иначале для модели (6 п.11.2), (1) предполагаются известными параметры шума $g^{(i)}_{(0)}$, все значения $\xi_{n\,(0)}$ $(n = 1, ..., N)$ предполагаются равными нулю.

При этих условиях минимизация формы (4) на шаге n приводит к системе m линейных уравнений

$$\partial Q_1/\partial \mathbf{A} = 0. \tag{6}$$

Решение этой системы позволяет получить первую итерацию для оценок параметров $\hat{a}^{(i)}_{(1)}$, $\hat{b}^{(i)}_{(1)}$. По этим оценкам вычисляются оценки

ненаблюдаемых координат шума

$$\hat{\xi}_n = z_n - \sum_{i=1}^{r} \hat{a}_{(1)}^{(i)} z_{n-i} - \sum_{i=1}^{l} \hat{b}_{(1)}^{(i)} x_n.$$

Зная эти координаты и минимизируя по параметрам $g^{(\nu)}$ форму

$$Q_1 = \sum_{i=1}^{n} (\hat{\xi}_i - \sum_{\nu=1}^{G} g^{(\nu)} \hat{\xi}_{i-\nu})^2,$$

находим первую итерацию оценок $\hat{g}_{(1)}^{(\nu)}$ как решение линейной системы уравнений

$$\partial Q_1 / \partial g^{(\nu)} = 0, \quad \nu = \overline{1,G}. \tag{7}$$

Подставив эти оценки в (6), определим второе приближение $\hat{a}_{(2)}^{(i)}$, $\hat{b}_{(2)}^{(i)}$ и снова $\hat{g}_{(2)}^{(\nu)}$ и т.д.

Сходимость такой процедуры к истинным значениям не доказана. Однако МТП хорошо зарекомендовал себя на практике, а результаты статистического моделирования показывают, что оценки МТП близки к истинным значениям параметров.

Метод преобразования модели (МПМ). Если единственной целью распознавания значений параметров модели является определение выходных переменных, целесообразно таким образом преобразовать исходную модель, чтобы ненаблюдаемые координаты шума и регрессоры (члены правой части уравнения) были некоррелированы. В этом случае оценки МНК параметров состоятельны и эффективны, а полученные на их основе оценки выходных переменных (и значения этих переменных) имеют минимальную дисперсию.

Пусть уравнение объекта задано в форме

$$\left.\begin{aligned} z_n &= \sum_{i=1}^{r} a^{(i)} z_{n-i} + \sum_{i=1}^{l} b^{(i)} x_{n-i} + \xi_n; \\ \xi_n &= \sum_{i=1}^{G} g^{(i)} \xi_{n-i} + \sigma_0 \epsilon_n. \end{aligned}\right\} \tag{8}$$

Такая модель может быть преобразована к виду

$$z_n = \sum_{i=1}^{r+G} u^{(i)} z_{n-i} + \sum_{i=1}^{l+G} v^{(i)} x_{n-i} + \sigma_0 \epsilon_n. \tag{9}$$

Состоятельные эффективные оценки параметров $u^{(i)}$ и $v^{(i)}$ определяются МНК. Если обозначить $\mathbf{A} = [u^{(1)}, \ldots, u^{(r+G)}, v^{(1)}, \ldots, v^{(l+G)}]^{\mathrm{T}}$, то оценка МНК вектора $\mathbf{A}$

определяется из выражения (9 п.11.2), где для модели (9) матрица **X** записывается в виде

$$\mathbf{X} = \begin{bmatrix} z_{n-1} \cdots z_{n-r-G} & x_{n-1} \cdots x_{n-l-G} \\ \vdots \quad\quad \vdots & \vdots \quad\quad \vdots \\ z_0 \cdots z_{-r-G} & x_0 \cdots x_{-l-G} \end{bmatrix}.$$

Оценивание z_{n+1} по полученным $\hat{u}^{(i)}$ и $\hat{v}^{(i)}$ оптимально в среднеквадратическом смысле.

Если все же необходимо найти параметры $a^{(i)}$, $b^{(i)}$, $g^{(i)}$, можно воспользоваться следующими приемами. Обозначим в соответствии с правилами дискретного преобразования Лапласа:

$$\hat{G}(q) = \sum_{i=0}^{G} \hat{g}^{(i)} q^{-i}, \; g_0^{(i)} = 1;$$

$$\hat{A}(q) = \sum_{i=0}^{r} \hat{a}^{(i)} q^{-i}, \; \hat{a}_{(0)}^{(i)} = 1;$$

$$\hat{B}(q) = \sum_{i=1}^{l} \hat{b}^{(i)} q^{-i},$$

$$\hat{U}(q) = \sum_{i=0}^{r+G} \hat{u}^{(i)} q^{-i}, \; \hat{u}_{(0)}^{(i)} = 1;$$

$$\hat{V}(q) = \sum_{i=1}^{l+G} \hat{v}^{(i)} q^{-i}, \; x_n q^{-i} \triangleq x_{n-i}.$$

Пользуясь алгоритмом последовательного деления многочленов *U(q)* и *V(q)*, выделим наибольший общий делитель этих многочленов, который и является полиномом $\hat{G}(q)$; полиномы $\hat{A}(q)$ и $\hat{B}(q)$ определяютсяиз выражений

$$\hat{A}(q) = \hat{U}(q)/\hat{G}(q); \quad \hat{B}(q) = \hat{V}(q)/\hat{G}(q).$$

11.4. Метод максимального правдоподобия (ММП)

Метод максимального правдоподобия (ММП) имеет только вероятностную интерпретацию. Он справедлив лишь для стохастических моделей, т.е. для моделей, в которых выходная переменная зависит от ненаблюдаемого случайного сигнала.

За оценку параметра принимается то его значение, при котором функция плотности вероятности наблюдаемых случайных переменных имеет максимальное значение.

Для того чтобы записать выражение плотности распределения вероятностей, нужно знать аналитическое выражение закона распределения. Часто предполагается, что аддитивные случайные помехи в уравнении модели распределены нормально. В этом случае оценки ММП для линейных моделей с независимым шумом совпадают с оценками МНК, а для линейных моделей с зависимым шумом — с оценками ОМНК.

Удается доказать ряд весьма привлекательных свойств оценок ММП: для широкого класса распределений помех оценки ММП состоятельны, асимптотически эффективны и нормально распределены. Поэтому ММП более широко используется для исследования свойств различных видов оценок, сопоставления их эффективности с эффективностью оценок ММП, чем для решения практических задач нахождения оценок параметров.

11.5. Байесовские оценки (БО)

Вероятностная интерпретация МНК и ММП для стохастических моделей предполагает, что определяемые составляющие вектора параметров являются числами - истинными значениями параметров. Получаемые нами на конечных выборках при различных критериях оценки являются случайными величинами, имеющими вероятностные характеристики.

В байесовских методах оценивания предполагается, что сами определяемые параметры *a, b* являются случайными величинами. Как и любые случайные величины, эти параметры характеризуются условными и безусловными функциями плотности вероятностей. Обозначим условную функцию распределения при наблюдаемых *z, x* через $p(\boldsymbol{\theta}|z, x)$, где $\boldsymbol{\theta} = [a^{(1)} \ldots a^{(r)}, b^{(1)} \ldots b^{(l)}]^{\mathrm{T}}$ — вектор неизвестных параметров.

Для определения критерия вводится понятие функции потерь $c(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, которая зависит от принятой оценки $\hat{\boldsymbol{\theta}}(z, x)$. В частности, функция потерь может представлять собой квадратическую форму вектора ошибки параметров, т.е может быть записана в виде

$$c[\hat{\boldsymbol{\theta}}(z, x)] = [\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(z, x)]^{\mathrm{T}} \mathbf{S} [\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(z, x)], \qquad (1)$$

где **S** - неотрицательно определенная симметричная матрица, с помощью которой могут быть заданы веса различных составляющих случайного вектора $\theta - \hat{\theta}$. В случае гауссовских аддитивных независимых помех такая функция потерь приводит для линейных моделей к оценкам МНК.

Функция риска *R* представляет собой осредненное значение функции потерь, т.е.

$$R = \int_{-\infty}^{\infty} c[\hat{\theta}(z, x)]\, p(\theta | z, x) d\theta. \qquad (2)$$

Минимизация функции риска (2) по параметрам θ и определяет байесовскую оценку (БО) параметров.

В частности, минимизация при квадратичной функции потерь (1) приводит к выводу, что лучшей байесовской оценкой $\hat{\theta}(z, x)$ является условное математическое ожидание оцениваемой переменной, т.е.

$$\hat{\theta}(z, x) = \int_{-\infty}^{\infty} \theta p(\theta | z, x) d\theta. \qquad (3)$$

Для расчетов по этой формуле необходимо задание условной плотности распределения $p(\theta | z, x)$, которая почти никогда не известна заранее, поэтому применение байесовских оценок осуществляется, как правило, когда наблюдаемые переменные состояния *x, z* вводятся в систему вычислений последовательно и плотность $p(\theta | z, x)$ пересчитывается на каждом такте ввода по формулам Байеса для пересчета априорной плотности вероятности в апостериорную. При этом необходимо задать априорную плотность распределения до первого такта измерения. В дальнейшем осуществляется уточнение плотности распределения оценок искомых параметров.

Рассмотрим в качестве примера рекуррентного пересчета плотностей распределения и оценок простейшую линейную модель

$$y_n = ax_n + \sigma\epsilon_n;\ \ \epsilon_n \sim N(0, 1);\ \ \mathrm{cov}(\epsilon_n, x_n) = 0. \qquad (4)$$

Как упоминалось выше, нужно теперь предположить, что параметр *a* является случайной величиной, и следует задать его априорное распределение *p(a).* Допустим, что на *n*-м шаге априорное распределение зададим гауссовским, и в этом распределении $M(\hat{a}) = a$ и $D(\hat{a}) = \sigma_a^2$.

Перейдем на (*n* + 1)-й шаг, где получим измерения x_{n+1}, y_{n+1}.
Из модели (4) легко получить

$$M(y_{n+1}) = ax_{n+1};\ \ D(y_{n+1}) = \sigma_a^2 x_{n+1}^2 + \sigma^2.$$

Нужно найти условное распределение

$$p(a | x_{n+1}, y_{n+1}) \triangleq p(a | z_{n+1}).$$

Байесовская оценка является функцией этого распределения. В частности, если функция потерь c квадратичная, т.е. $c(a) = (a - \alpha)^2$, то в соошетствии с (3) байесовская оценка является математическим ожиданием условного распределения $p(a|z_{n+1})$.

Значение $p(a|z_{n+1})$ на $(n + 1)$-м шаге может быть найдено по формуле Байеса

$$p(a|z_{n+1}) = p(a)\, p(y|a)/p(y).$$

Все распределения в правой части мы знаем. Поэтому можно записать

$$p(a|z_{n+1}) = \text{const} \cdot \exp\left[-\frac{1}{2}\left(\frac{(a-\alpha)^2}{\sigma_a^2} + \frac{(y_{n+1} - ax_{n+1})^2}{\sigma^2} - \frac{(y_{n+1} - \alpha x_{n+1})^2}{x_{n+1}^2\sigma_a^2 + \sigma^2}\right)\right].$$

Отсюда, воспользовавшись (3), найдем

$$\hat{a}_{n+1} = (\sigma^2\alpha + \sigma_a^2 x_{n+1} y_{n+1})/(x_{n+1}^2\sigma_a^2 + \sigma^2).$$

Слагаемое $\sigma^2\alpha$ учитывает прошлую информацию. Воспользовавшись оценкой a на $(n + 1)$-м шаге, мы должны сделать следующий шаг. Заметим, что апостериорное распределение $\hat{a}$ осталось гауссовским. Байесовский метод нашел широкое практическое применение в решении задач оптимального оценивания частично наблюдаемых последовательностей после разработки оптимального фильтра Калмана (ФК). Калманом решена задача оценивания ненаблюдаемых переменных состояния θ_n по наблюдениям ξ_n. Модели Калмана для дискретного времени принято записывать в следующей форме:

$$\left.\begin{aligned}\boldsymbol{\xi}_{n+1} &= \mathbf{A}_0 + \mathbf{A}_1\boldsymbol{\theta}_n + \mathbf{B}_1\boldsymbol{\epsilon}_{1\,(n+1)} + \mathbf{B}_2\,\boldsymbol{\epsilon}_{2\,(n+1)};\\ \boldsymbol{\theta}_{n+1} &= \mathbf{a}_0 + \mathbf{a}_1\boldsymbol{\theta}_n + \mathbf{b}_1\boldsymbol{\epsilon}_{1\,(n+1)} + \mathbf{b}_2\boldsymbol{\epsilon}_{2\,(n+1)},\end{aligned}\right\} \qquad (5)$$

где $\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \mathbf{A}_0, \mathbf{a}_0$ — векторы; $\mathbf{A}_1, \mathbf{a}_1, \mathbf{B}_1, \mathbf{B}_2, \mathbf{b}_1, \mathbf{b}_2$ — матрицы.

Элементы векторов $\mathbf{a}_0$, $\mathbf{A}_0$ и всех матриц могут зависеть от $\boldsymbol{\xi}_0^n = (\xi_0, \xi_1, \ldots, \xi_n)$ - последовательности наблюдений от 0 до n.

Фильтр Калмана для оценивания неизвестных параметров применяется при очевидном предположении о неизменности параметров объекта во времени, т.е. $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n$, где θ — вектор неизвестных параметров.

В качестве оценки параметров на шаге $n + 1$ принимается условное математическое ожидание вектора θ при данных наблюдениях, $\xi_0, \xi_1, \ldots \ldots, \xi_{n+1}$, т.е. оценкой параметра является величина

$$m_{n+1} \triangleq M(\theta | \xi_0^{n+1}).$$

Применительно к модели (6 п.11.2) при $\xi_n = \sigma_0 \epsilon_n$ модель Калмана может быть записана в виде

$$\left.\begin{aligned} \xi_{n+1} &= \varphi_n \theta_n + \sigma_0 \epsilon_{n+1}; \\ \theta_{n+1} &= \theta_n, \end{aligned}\right\} \qquad (6)$$

где в (6 п.11.2) обозначено

$$\varphi_n = [z_{n-1}, \ldots, z_{n-r}, x_{n-1}, \ldots, x_{n-l}]; \quad \theta_n = [a_1, \ldots, a_r, b_1, \ldots, b_l]^T;$$

$$\xi_{n+1} = z_{n+1}, \ \sigma_0 \epsilon_{n+1} = \xi_{n+1}.$$

(В модели Калмана (5) и (6) буквой ξ обозначаются все наблюдаемые переменные, т.е. x и z в (6 п.11.2). Вместо ненаблюдаемых переменных шума ξ_n в (6 п.11.2) в модели Калмана используется обозначение $\sigma\epsilon_n$, но для модели Калмана ϵ_n - всегда только независимая последовательность.)

Используя формулы для условного математического ожидания и условной дисперсии фильтра Калмана, можно записать уравнение для оценок параметров в виде

$$\mathbf{m}_{n+1} \triangleq \hat{\theta}_{n+1} = \hat{\theta}_n + k_{n+1} \mathbf{e}_{n+1}, \qquad (7)$$

где

$$\mathbf{k}_{n+1} = \gamma_n \varphi_n / [\sigma_0^2 + \varphi_n^T \gamma_n \varphi_n]; \quad \gamma_{n+1} = \gamma_n - \gamma_n \varphi_n \varphi_n^T \gamma_n^T / [\sigma_0^2 + \varphi_n^T \gamma_n \varphi_n];$$

$$\mathbf{e}_{n+1} = \xi_{n+1} - \varphi_n \mathbf{m}_n.$$

Для запуска этих алгоритмов необходимо задание значений m_0 и γ_0, т.е. вектора начальных значений параметров и матрицы начальных значений их ковариаций. Сходимость к истинным значениям будет иметь место при любых начальных значениях. Эксперименты показывают, что можно получить некоторое повышение скорости, если значения дисперсий и ковариаций в матрице $\gamma(0)$ приняты больше их истинных значений. Существенным ограничением для применения алгоритмов Калмана является необходимость знания дисперсии независимого ненаблюдаемого шума σ_0^2. Неверное задание σ_0^2 приведет к смещенным сценкам.

Предложено несколько эмпирических алгоритмов, построенных с помощью фильтра Калмана для определения оценок параметров объ-

екта вместе с оценками параметров шума. Один из наиболее конструктивных алгоритмов для линейной модели с шумом рассмотрен ниже. Пусть модель распознаваемого объекта записана в виде

$$z_n = \sum_{i=1}^{r} a_i z_{n-i} + \sum_{i=1}^{l} b_i x_{n-i} + \sum_{i=0}^{G} \sigma_i \epsilon_{n-i} \qquad (8)$$

и $\sigma_0 = 1$. Тогда параметры a_i, b_i, σ_i можно определить с помощью следующего алгоритма. Обозначим

$$\theta^T = [a_1, ..., a_r, b_1, ..., b_l, \sigma_1, ..., \sigma_G];$$

$$\varphi_n = [z_{n-1}, ..., z_{n-r}, x_{n-1}, ..., x_{n-l}, \epsilon_{n-1}, ..., \epsilon_{n-G}].$$

Тогда уравнение (8) может быть записано в виде

$$\left.\begin{aligned} \xi_{n+1} &= \varphi_n \theta_n + \epsilon_{n+1}; \\ \theta_{n+1} &= \theta_n, \end{aligned}\right\} \qquad (9)$$

где, как и в (6), $\xi_{n+1} = z_{n+1}$.

Поскольку это уравнение идентично (6), для определения $\mathbf{m}_{n+1}$, γ_{n+1} можно применить (7). Вектор-строка φ_n включает ненаблюдаемые переменные последовательности случайного шума ϵ_n. Оценки ненаблюдаемых переменных вычисляются на каждом шаге из соотношения $\hat{\epsilon}_n = z_n - \hat{\varphi}_n \hat{\theta}_{n-1}$, где

$$\hat{\varphi}_n = [z_{n-1}, ..., z_{n-r}, x_{n-1}, ..., x_{n-l}, \hat{\epsilon}_{n-1}, ..., \hat{\epsilon}_{n-G}]. \qquad (10)$$

Вместо истинных значений ϵ_{n-i} в вектор $\hat{\varphi}_n$ подставляются оценки $\hat{\epsilon}_{n-i}$. Таким образом можно проводить одновременно оценивание параметров и объекта, и шума с помощью зависимостей (7). Для этого должна быть известна дисперсия независимого шума σ_0^2. Подстановка неверного значения σ_0^2 приводит к смещению оценок параметров θ.

В случае, если в модели (6 п.11.2) шум гауссовский и функция потерь квадратична, БО отличаются от оценок МНК и ММП только из-за влияния априорно заданных начальных условий. С увеличением выборки влияние начальных условий уменьшается и байесовские оценки приближаются к оценкам МНК и ММП.

11.6. Метод инструментальной переменной (МИП)

Зададим модель в виде (7 п.11.2), т.е. $\mathbf{Z} = \mathbf{XA} + \Xi$. Пусть $\lim\limits_{T\to\infty} \frac{1}{T} \mathbf{X}^{\mathrm{T}} \Xi \neq 0.$

Следовательно, измеряемые переменные x в правой части и ненаблюдаемый шум Ξ коррелированы, оценки, полученные, например, МНК, смещены, а оценки, полученные ОМНК, ММП и БО, требуют знания параметров, описывающих модель шума, которые обычно неизвестны. Допустим, удается подобрать некоторые измерения, образующие матрицу Y размером $N \times m$, такие, что

$$\lim_{T\to\infty} \frac{1}{T} \mathbf{Y}^{\mathrm{T}} \Xi = 0. \tag{1}$$

Тогда из (7 п.11.2) следует

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Z} = \mathbf{Y}^{\mathrm{T}}\mathbf{XA} + \mathbf{Y}^{\mathrm{T}}\Xi, \tag{2}$$

и поскольку $\mathbf{Y}^{\mathrm{T}} \Xi \to 0$, можно определять оценки $\hat{\mathbf{A}}$ из выражения

$$\hat{\mathbf{A}} = (\mathbf{Y}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Z}. \tag{3}$$

Матрица **Y** называется *матрицей инструментальной (вспомогательной) переменной.* Она может быть сформирована из результатов измерений сигналов, не воздействующих на систему непосредственно, но коррелированных с сигналами в системе. Эта матрица может быть сформирована из тех же сигналов *x, z*, но с измерениями, сдвинутыми таким образом, чтобы корреляция с шумом исчезла *(метод сдвига).* Например, пусть модель записана в виде (8 п.11.5), т.е.

$$z_n = \sum_{i=1}^{r} a_i z_{n-i} + \sum_{i=1}^{l} b_i x_{n-1} + \sum_{i=0}^{G} \sigma_i \epsilon_{n-i}.$$

Матрица **X** и вектор **Z**, как и раньше, формируются в виде

$$\mathbf{X} = \begin{bmatrix} z_{n-1}, \ldots, z_{n-r}, x_{n-1}, \ldots, x_{n-l} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ z_{n-N+1}, \ldots, z_{n-r-N}, x_{n-N+1}, \ldots, x_{n-l-N} \end{bmatrix};$$

$$\mathbf{Z}_n = [z_n, \ldots, z_{n-N}]^{\mathrm{T}}.$$

Ненаблюдаемый шум $\xi_n = \sum\limits_{i=0}^{G} \sigma_i \epsilon_{n-i}$ описывается моделью "скользящее среднее", т.е. компоненты шума $\epsilon_{n-i}, \epsilon_{n-j}$ некоррелированы межу собой, если $i \neq j$.

Допустим, что переменные x_{n-i} некоррелированы с ξ_n. Из уравнения модели следует, что переменные z_{n-i} коррелированы с ϵ_{n-i}, если $r > i > j$.

Если $r \leqslant G$, матрица инструментальной переменной **Y** при использовании метода сдвига может быть сформирована в виде

$$\mathbf{Y} = \begin{bmatrix} z_{n-D}, \ldots, z_{n-D-k}, x_{n-1}, \ldots, x_{n-l} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ z_{n-D-N}, \ldots, z_{n-D-k-N}, x_{n-N+1}, \ldots, x_{n-l-N} \end{bmatrix}.$$

Выбор D в **Y** определяется корреляцией компонент матриц **Y** и **X**. Точность оценок на конечной выборке тем больше, чем больше значение определителя матрицы $\mathbf{Y}^{\mathrm{T}}\mathbf{X}$. Модификации метода инструментальной переменной получили наименования методов копирования, или сдвига, Юла-Уокера.

Рассмотрим метод Юла—Уокера. Пусть модель записана в виде

$$z_n = \sum_{i=1}^{r} a_i x_{n-i} + \xi_n; \qquad (4)$$

сигнал y_{n-l} коррелирован z и x, но не коррелирован с ξ_n. Умножим уравнение (4) на y_{n-l}. Осреднив обе части уравнения по всей выборке, в пределе при $n \to \infty$ получим уравнение связи между вторыми моментами в виде

$$K_{zy}^{l} = \sum_{i=1}^{r} a_i K_{xy}^{i-l}, \qquad (5)$$

где $K_{zy}^{l} = M(z_n y_{n-l})$ — корреляционный момент сигналов z_n и y_{n-l}.

Используя различные сигналы y_i (в их числе могут быть и сигналы x, z), можно получить систему уравнений вида (5). Эти уравнения связи между вторыми моментами называются уравнениями Юла—Уокера (Ю-У).

Пусть образована система $r + l$ таких уравнений и определитель этой системы не равен нулю. Тогда система имеет решение относительно неизвестных параметров.

Нетрудно убедиться, что уравнения Ю—У при выборе одинаковых инструментальных переменных идентичны уравнениям (2). Обычно, разумеется, и в (2), и в (5) вместо математических ожиданий пользуются оценками вторых моментов, подсчитанных путем осреднения по формулам

$$\hat{K}_{z,y}^{l} = \frac{1}{N} \sum_{i=1}^{N} z_i y_{i-l}.$$

Оценки, полученные перечисленными модификациями МИП, состоятельны, т.е. с ростом выборки они стремятся к истинным значениям, и дисперсии оценок стремятся к нулю. Однако эффективность (дисперсия на конечной выборке) в сдвиговых методах МИП хуже, чем в ОМНК и ММП. Тем не менее МИП часто применяются в практических расчетах из-за относительной простоты и отсутствия необходимости знать параметры ненаблюдаемого шума.

Критерием методов инструментальных переменных по существу является требование некоррелированности компонент матрицы инструментальных переменных и ненаблюдаемого шума, т.е. минимизация суммы вторых моментов этих компонент.

11.7. Метод стохастической аппроксимации (СА)

Метод стохастической аппроксимации разработан для определения корней уравнения, когда значение функции при заданном значении аргумента наблюдается с помехой.

Пусть, например, в уравнении

$$Y(a) = 0 \tag{1}$$

нужно определить корни a^*, но Y при каждом a не наблюдается, а наблюдается некоторое значение $Z(a)$, о котором известно, что

$$M[Z(a)] = Y(a). \tag{2}$$

Метод СА организует некоторую последовательность a_n, такую, что $a_n \to a^*$ при $n \to \infty$. Члены этой последовательности образуются рекуррентной формулой

$$a_{n+1} = a_n + \gamma_n [Z_{n+1}(a_n)]. \tag{3}$$

Если уравнение (1) записано в виде $Y(a) = a$, то последовательность (3) имеет форму

$$a_{n+1} = a_n + \gamma_{n+1} [Z_{n+1}(a_n) - a] \tag{4}$$

или, обозначив $[Z_{n+1}(a_n) - a] = S_{n+1}$, получим:

$$a_{n+1} = a_n + \gamma_{n+1} S_{n+1}. \tag{5}$$

Доказано, что если

$$\sum_{n=1}^{\infty} \gamma_n = \infty, \quad \sum_{n=1}^{\infty} \gamma_n^2 < \infty, \tag{6}$$

дисперсия помех, наложенных на функцию $Y(a)$, ограничена и $Y(a)$ — монотонная функция, то a_n сходится к a^*.

Выражение в квадратных скобках в формулах (3) и (4), обозначенное S_{n+1} в (6), называется *невязкой,* γ_{n+1} - *коэффициентом усиления.* Вектору параметров A_n соответствуют вектор невязок S_n и матрица коэффициентов усиления Γ_n.

Условиям (6) отвечает большое число последовательностей, например $\gamma_n = c/n$, где c — произвольное число.

Метод СА легко переносится на задачи определения параметров стохастических систем в условиях последовательного получения оценок (рекуррентное распознавание).

Пусть уравнение модели объекта задано в виде

$$z_n = f(x_n, a, \xi_n),$$

где z_n, x_n - соответственно наблюдаемые выходные и входные переменные; a — неизвестный параметр; ξ_n — ненаблюдаемый шум.

Если $M(z_n) = f(x_n a)$, что имеет место, например, для аддитивных центрированных шумов ξ_n, т.е. для модели вида

$$z_n = f(x_n, a) + \xi_n, \qquad (7)$$

то в соответствии с методом СА может быть организована последовательность

$$a_{n+1} = a_n + \gamma_{n+1}[z_{n+1} - f(x_{n+1}, a_n)]. \qquad (8)$$

Единственное отличие от (3) заключается в том, что теперь измеренное значение функции z_n зависит не только от шума ξ_n и определяемого параметра a, но и от входной переменной x_n, т.е. в уравнении (3) появляется дополнительный параметр x_n. Алгоритм определения параметра a и условия на γ_n не изменяются.

Для случая, когда **a** является вектором, а z_n — скаляром, использовать (8) нельзя, поскольку исходное уравнение (1) не имеет однозначного решения (число уравнений меньше числа неизвестных).

Чтобы для модели (8) в случае векторного a воспользоваться алгоритмом СА (5), необходимо иметь размерность вектора невязки равную размерности вектора неизвестных параметров **a**.

Введем для этого некоторый критерий идентификации, в точке экстремального значения которого находятся искомые параметры.

Пусть, например, модель объекта задана в виде (7) и $M(\xi_n) = 0$. Скалярный показатель качества распознавания (функция потерь) может быть определен в виде

$$\eta_{n+1}(a_n) = [z_{n+1} - f(\mathbf{x}_{n+1}, \mathbf{a}_n)]^2.$$

Тогда вектор невязок $\mathbf{S}_{n+1}$ может быть определен с помощью выражения

$$\mathbf{S}_{n+1} = (d\eta / d\mathbf{a})_{\mathbf{a}_n}.$$

Заметим, что математическое ожидание вектора невязок в точке $\mathbf{a}^*$ будет на каждом шаге равно 0, как и в (1).

Сходимость алгоритма стохастической аппроксимации доказана для зависимых и независимых последовательностей $\{z_n\}$.

Основной недостаток метода СА — медленная сходимость оценок параметров к истинному значению, даже если дисперсия шума существенно меньше дисперсии выходного сигнала. Несмотря на медленную сходимость оценок, алгоритмы СА благодаря своей простоте находят применение в практических задачах распознавания линейных и нелинейных моделей объектов с аддитивным независимым шумом. Сушествует ряд методов ускорения сходимости путем соответствующего подбора матрицы $\mathbf{\Gamma}_n$.

Существенное ускорение сходимости методов стохастической аппроксимации для линейных моделей можно получить, используя известные алгоритмы. Расчет элементов диагональной матрицы усиления $\mathbf{\Gamma}_n = [\gamma_n^{(i)}]$ в этих алгоритмах выполняется по формулам

$$\gamma_n^{(i)} = R_n^{(ii)} / n \sum_{j=1}^{m} [R_n^{(ij)}]^2, \qquad (9)$$

где $R_n^{(ij)}$ — элементы матрицы $\mathbf{R}_n$ оценок вторых моментов, которая определяется для модели (7 п.11.2) выражением

$$\mathbf{R}_n = \mathbf{X}_n^{\mathrm{T}} \mathbf{X}_n.$$

Однако, если элементы матрицы $\mathbf{\Gamma}_n$ зависят не только от номера шага n (времени), но и от наблюдаемых переменных, алгоритмы СА становятся близки к другим алгоритмам рекуррентного оценивания, в частности к рекуррентному МНК или алгоритмам фильтра Калмана.

Известны методы рекуррентного оценивания параметров, построенные на основе алгоритмов стохастической аппроксимации, но с оптимальным выбором матрицы $\mathbf{\Gamma}_n$,. Здесь построены оптимальные алгоритмы рекуррентного распознавания, но, в отличие от методов БО и ФК, построение этих алгоритмов проведено исходя из предположения, что параметры являются постоянными величинами, а не случайными числами. В качестве критерия оптимальности принята асимптотическая матрица ковариаций ошибок (АМКО) определяемых параметров. Значение АМКО, по определению, записывается в виде

$$\mathbf{V} = \lim_{n\to\infty} \mathbf{V}_n = \lim_{n\to\infty} M\{(\mathbf{A}_n - \mathbf{A}^*)(\mathbf{A}_n - \mathbf{A}^*)^{\mathrm{T}}\}.$$

Показано, что АМКО приближенно характеризует асимптотическую скорость сходимости оценок параметров, и получена зависимость АМКО от плотности распределения помех p_0 и функции потерь $F(\epsilon)$, $\epsilon_n = z_n - \hat{z}_n$, т.е. $V = V[F(\epsilon), p_0]$. Минимизация АМКО по F позволяет найти значение F при различных функциях плотности распределения помех p_0. Найденные функции потерь обеспечивают минимум АМКО или максимум асимптотической скорости сходимости. В частности, оказалось, что для гауссовской помехи оптимальным алгоритмом оценивания параметров является МНК.

Следует помнить, однако, что полученные результаты, важные для теории распознавания, справедливы в асимптотике. К сожалению, поведение алгоритмов распознавания в асимптотике не всегда характеризует их поведение на конечных выборках.

11.8. Метод осредненных невязок (МОИ)

Общая постановка. Основное отличие метода от других рекуррентных алгоритмов заключается в том, что в МОН рекуррентная процедура используется для определения статистик процесса, а оценки параметров модели строятся на этих статистиках. В других рекуррентных процедурах используются мгновенные значения переменных процесса для нахождения оценок параметров.

Поясним сказанное примером. Рассмотрим статическую линейную модель

$$y_n = \theta^* x_n + \sigma\epsilon_n, \quad M\{\epsilon_n\} = 0. \tag{1}$$

Пусть оценка параметра θ^* на шаге n определяется с помощью обычного рекуррентного алгоритма

$$\theta_n = \theta_{n-1} + \gamma_n S_n(\theta_{n-1}), \tag{2}$$

где γ_n — коэффициент усиления; S_n — невязка.

Чтобы проиллюстрировать особенности МОН, рассмотрим вначале способы формирования γ_n и S_n в (2) для алгоритмов стохастической аппроксимации (СА). Невязка S_n формируется в алгоритме СА путем минимизации функции потерь на каждом шаге. Пусть на любом n-м шаге функция потерь для модели (1) записывается в виде

$$F(\epsilon_n, \theta) = (y_n - \theta x_n)^2. \tag{3}$$

Дифференцируя эту функцию по θ, найдем выражение невязки в виде

$$S_n(\theta) = \partial F / \partial\theta = -2(y_n - \theta x_n).$$

Истинное значение параметра является корнем функции

$$R(\theta) = M\{S_n(\theta) \mid x_n, \theta\}. \qquad (4)$$

Докажем это утверждение. Подставив в (4) значение y_n из (1). получим выражение

$$S_n(\theta) = 2x_n^2(\theta - \theta^*) - 2\sigma x_n \epsilon_n.$$

Отсюда

$$R_n(\theta) = 2x_n^2(\theta - \theta^*) - 2\sigma x_n M(\epsilon_n \mid x_n, \theta).$$

Поскольку ϵ_n и x_n, а также ϵ_n и θ_n независимы и $M(\epsilon_n) = 0$, то

$$R_n(\theta) = 2x_n^2(\theta - \theta^*).$$

Следовательно $R_n(\theta) = 0$, если $\theta = \theta^*$. Из сопоставления выражений Для S_n и R_n следует, что невязка S_n представляет собой сумму функции $R_n(\theta)$ и шума со среднеквадратическим отклонением, равным $2\sigma x_n$. Функцию $R_n(\theta)$ будем называть в дальнейшем *функцией регрессии.*

Известно, что если на некоторую функцию наложен независимый аддитивный шум, то корень математического ожидания этой функции (т.е. функции регрессии) может быть нейден с помощью процедуры (2). При этом необходимо выполнение условий Дворецкого на γ_n:

$$\Sigma\gamma_n = \infty, \ \Sigma\gamma_n^2 < \infty. \qquad (5)$$

Поскольку θ* является корнем R_n, то последовательность (2) сходится к истинному значению искомого параметра.

Некоторым отличием рассмотренной постановки от известной процедуры Робинса-Монро (РМ) является наличие дополнительного параметра x_n. Поэтому при распознавании методами СА объекта со нходным сигналом x_n вместо одной линии регрессии будет семейство таких линий. Каждая из них имеет одно наблюдаемое значение распознающего воздействия x_n. На все эти линии наложен аддитивный шум с одинаковой дисперсией σ^2. Поведение линий регрессии $R_n(\theta)$ и линий $S_n(\theta)$ иллюстрируется графиками на рис. 1, *б.*

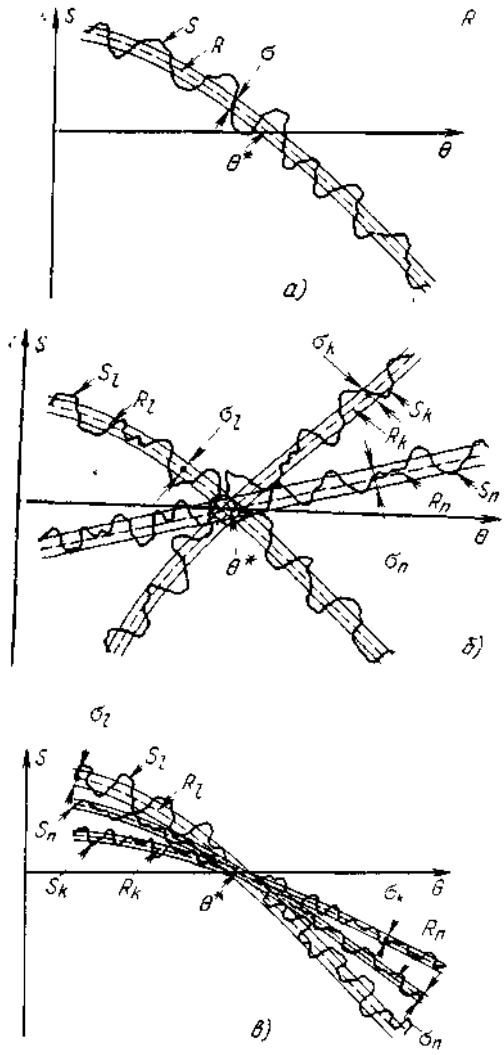


Рис. 1. Зависимость функций невязки и линий регрессии от параметра для алгоритмов РМ *(а)*, СА *(б)* и МОН *(в)* *(r <k <n)*

Искомое значение θ^* является точкой пересечения линий регрессии $M\{S_n|\theta, x_n\} = R_n(\theta, x_n)$ с осью абсцисс.

В методах СА на каждом шаге поиска осуществляется переход на новую линию регрессии, определяемую параметром x_n.

При постоянном распознавании ($x_n = \text{const}$) имеет место постановка РМ; поиск θ^* осуществляется на единственной линии $R(\theta)$ (рис. 1,а).

Пусть в отличие от (3) критерий качества распознавания записывается в форме

$$F_n^* = \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta x_i)^2, \qquad (6)$$

т.е. в форме критерия МНК. Тогда, произведя для этого критерия те же выкладки, что и для СА, найдем

$$S_n^* = \frac{\partial F_n^*}{\partial \theta} = -2 \left(\frac{1}{n} \sum_{i=1}^{n} y_i x_i - \frac{1}{n} \sum_{i=1}^{n} \theta x_i^2 \right). \qquad (7)$$

После подстановки в (7) значения y_n из (1) получим

$$S_n^* = 2(\theta - \theta^*) \frac{1}{n} \sum_{i=1}^{n} x_i^2 - \frac{2}{n} \sigma \sum_{i=1}^{n} x_i \epsilon_i. \qquad (8)$$

Линии регрессии R^* для этих функций невязки записываются в виде

$$R_n^*(\theta) = M\{S_n | \theta, x_n\} = 2(\theta - \theta^*) \frac{1}{n} \sum_{i=1}^{n} x_i^2. \qquad (9)$$

Очевидно, и это семейство функций регрессии имеет единственный корень θ^*, но семейство $R_n^*(\theta)$ отличается от семейства (4) $R_n(\theta)$ следующими особенностями:

1) поскольку $\lim\limits_{n \to \infty} \frac{1}{n} \sum\limits_{i=1}^{n} x_i^2 = R_{xx} = \text{const}$, то семейство линий регрессии стягивается в этом случае с ростом выборки в одну линию регрессии; в постановке СА этого стягивания с ростом выборки нет;

2) случайные величины $x_i \epsilon_i$ и $(1/n) \sum\limits_{i=1}^{n} x_i \epsilon_i$ для независимых x_i и ϵ_i имеют нулевые математические ожидания.

Предположим также, что независимы $x_i \epsilon_i$ и $x_j \epsilon_j$, если $i \neq j$. Пусть $M\{x_i \epsilon_i\}^2 = \sigma_\varphi^2$.

Тогда

$$M \left\{ \frac{1}{n} \sum_{i=1}^{n} x_i \epsilon_i \right\}^2 = \sigma_\varphi^2 / n.$$

Следовательно, в (7) дисперсия помех, наложенных на линии регрессии, убывает, стремясь к 0 с ростом выборки. Это последнее

обстоятельство отличает разбираемую задачу от условий постановки РМ, где дисперсия помех постоянна. Заметим, что если на линию регрессии не наложены помехи, вместо алгоритмов стохастической аппроксимации можно применить алгоритмы Ньютона, служащие для нахождения корней детерминированных систем уравнений и сходящиеся со скоростью геометрической прогрессии. Коэффициенты γ_n в алгоритмах Ньютона не стремятся к 0, а определяются значением производной линии регрессии.

Для итерационной процедуры с осредненной невязкой вида (7) также нет необходимости требовать убывания последовательности $\{\gamma\}$.

На рис. 1,*в* иллюстрируется поведение функций невязки (7) линий регрессии (9).

Рассмотрим подробнее метод распознавания, который в дальнейшем будет именоваться методом осредненных невязок (МОН).

МОН предназначен для рекуррентного оценивания параметров моделей вида

$$\mathbf{y}_n = \varphi(\mathbf{x}_n, \boldsymbol{\theta}^*) + \sigma\boldsymbol{\epsilon}_n, \qquad (10)$$

где $\mathbf{y}_n$ и $\mathbf{x}_n$ — векторы наблюдаемых переменных процесса; $\boldsymbol{\theta}^*$ — вектор оцениваемых параметров; σ — матрица ковариаций; $\boldsymbol{\epsilon}_n$ — вектор ненаблюдаемого шума.

Для оценивания вектора параметров используются рекуррентные алгоритмы типа

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \boldsymbol{\Gamma}_n \mathbf{S}_n, \qquad (11)$$

где $\boldsymbol{\theta}_n$ — оценка вектора $\boldsymbol{\theta}^*$ на n-м шаге; $\boldsymbol{\Gamma}_n$ — матрица коэффициентов усиления; $\mathbf{S}_n$ — вектор невязки.

В алгоритмах МОН вектор невязки строится на статистиках, оценки которых вычисляются рекуррентно. Если обозначить вектор наблюдений, поступающих на n-м шаге через $\mathbf{z}_n = (\mathbf{y}_n, \mathbf{x}_n)^T$ то $\mathbf{S}_n = \mathbf{S}(\mathbf{S}_{n-1}, \mathbf{z}_n)$.

Таким образом, рекуррентные алгоритмы МОН записываются в форме

$$\boldsymbol{\theta}_n = \theta(\boldsymbol{\theta}_{n-1}, \mathbf{S}_n, \boldsymbol{\Gamma}_n) = \theta_{\text{МОН}}(\boldsymbol{\theta}_{n-1}, \mathbf{z}_n, \mathbf{S}_{n-1}, \boldsymbol{\Gamma}_n).$$

В этой записи алгоритмы СА имеют вид

$$\boldsymbol{\theta}_n = \theta_{\text{СА}}(\boldsymbol{\theta}_{n-1}, \mathbf{z}_n, \boldsymbol{\Gamma}_n).$$

Идея осреднения невязок вдоль выборки встречалась в некоторых работах. Эти алгоритмы в получили название модифицированных *алгоритмов одновременного и поочередного действия. С*ущественное отличие алгоритмов МОН от модифицированных алгоритмов заключается в том, что коэффициенты усиления в модифицированных

алгоритмах принимались сходящимися к 0 с ростом выборки. В МОН предусматривается выбор $\mathbf{\Gamma}_n$, вообще не стремящихся к 0, в частности, в виде постоянной матрицы. МОН был разработан и использован не только для линейных, но и для нелинейных систем и показал хорошую сходимость параметров.

Существуют синтезированые алгоритмы распознавания, обеспечивающие максимальную асимптотическую скорость сходимости параметров распознаваемой линейной модели при заданной функции распределения помех В практических задачах распознавания весьма важным оказывается поведение алгоритмов распознавания не только в асимптотике, т.е. на бесконечных выборках, но и на конечной выборке, характерной для реальных условий. Поведение различных критериев распознавания в асимптотике и на конечных выборках может не совпадать К сожалению, поведение критериев качества распознавания на конечных выборках для большинства стохастических моделей не удается исследовать аналитически Поэтому эти работы выполняются в основном путем статистического моделирования. Результаты обобщаются путем осреднения показателей качества, полученных на отдельных выборках. Отдельный, выборочный показатель качества не является представительным, поскольку получен по одной конкретной реализации случайного сигнала.

Результаты статистического моделирования показали, что с точки зрения важных для практики показателей качества распознавания (точность оценивания параметров и точность прогнозирования выходной переменной на конечной выборке, чувствительность самого алгоритма к выбору параметров) МОН не уступает традиционным алгоритмам, а иногда существенно превосходит их, причем это достигается ценой незначительного проигрыша в асимптотической скорости сходимости. Некоторые результаты таких исследований приведены ниже.

Линейные модели. В этом разделе рассматриваются вопросы сходимости и выбор матриц коэффициентов усиления алгоритмов МОН для линейных моделей.

Рассмотрим линейную модель с одномерным выходом y_n и m-мерным входом $z_n^{(i)}$ $(i = 1, ..., m)$, заданную уравнением

$$y_n = \sum_{i=1}^{k} a^{(i)} * y_{n-i} + \sum_{i=1}^{l} b^{(i)} * x_n^{(i)} + \xi_n \quad (k + l = m). \qquad (12)$$

Перепишем это уравнение в форме

$$y_n = \mathbf{Z}_n \boldsymbol{\theta}^* + \xi_n, \qquad (13)$$

где y_n — одномерный выход:

$$\mathbf{Z}_n = (y_{n-1}, y_{n-2}, \ldots, y_{n-k}, x_n^{(1)}, \ldots, x_n^{(l)})^{\mathrm{T}} = [z_n^{(1)}, \ldots, z_n^{(m)}]^{\mathrm{T}}.$$

В этих обозначениях

$$z_n^{(i)} = \begin{cases} y_{n-i}, & \text{если } 1 \leqslant i \leqslant k; \\ x_n^{(i-k)}, & \text{если } k < i \leqslant k+l; \end{cases}$$

$$\theta^* = [a^{*(1)}, \ldots, a^{*(k)}, b^{*(1)}, \ldots, b^{*(l)}] = [\theta^{*(1)}, \ldots, \theta^{*(k+l)}],$$

где $\theta^{*(i)} = \begin{cases} a^{*(i)}, & \text{если } 1 \leqslant i \leqslant k; \\ b^{*(i)}, & \text{если } k < i \leqslant k+l. \end{cases}$

Положим

$\mathbf{Z}_{(n)} = (\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n)^{\mathrm{T}}$; $\mathbf{Y}_n = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$; $\Xi_n = (\xi_1, \ldots, \xi_n)^{\mathrm{T}}$. Тогда (13 можно переписать в виде

$$\mathbf{Y}_n = \mathbf{Z}_n \theta^* + \Xi_n. \tag{14}$$

Пусть для этой модели известно:

1) $\operatorname{cov}[\mathbf{Z}_n \xi] = 0$;

2) система устойчива, т.е. корни характеристического полинома $P(z) = z_n - \sum_{i=1}^{k-1} \theta^{*(i)} z^{k-i}$ лежат вне единичного круга.

Требуется по наблюдениям $\mathbf{Z}_n$, $\mathbf{Y}_n$ оценить вектор параметров θ^* с помощью алгоритма МОН.

Сформируем вектор невязки $\mathbf{S}_n$. Для этого построим оценку осредненной функции потерь в виде

$$J_n = \frac{1}{n} (\mathbf{Y}_n - \mathbf{Z}_n \theta)^{\mathrm{T}} (\mathbf{Y}_n - \mathbf{Z}_n \theta).$$

Формируя вектор невязок из условия $dJ_n/d\theta = \mathbf{S}_n = 0$, придем к выражению

$$\mathbf{S}_n = \mathbf{L}_n - \mathbf{R}_n \theta = 0, \tag{15}$$

где $\mathbf{L}_n = (1/n) \mathbf{Z}_n^{\mathrm{T}} \mathbf{Y}_n$ — вектор $m \times 1$; $\mathbf{R}_n = (1/n) \mathbf{Z}_n^{\mathrm{T}} \mathbf{Z}_n$ — матрица $m \times m$.

Хорошо известна рекуррентная процедура для вычисления вектора $\mathbf{L}_n$ и матрицы $\mathbf{R}_n$ корреляционных моментов, например,

$$\mathbf{L}_n = \left(\frac{n-1}{n}\right) \mathbf{L}_{n-1} + \left(\frac{1}{n}\right) \mathbf{l}_n, \tag{16}$$

где $\mathbf{l}_n = \mathbf{z}_n \mathbf{y}_n$.

Будем определять оценку вектора $\boldsymbol{\theta}^*$ с помощью алгоритмов МОН вида

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \boldsymbol{\Gamma}_n (\mathbf{L}_n - \mathbf{R}_n \boldsymbol{\theta}_{n-1}). \qquad (17)$$

Очевидно, алгоритм (17) не требует хранения в памяти машины всей выборки. Для его реализации достаточно хранить в памяти на шаге n только матрицу $\mathbf{R}_{n-1}$, векторы $\mathbf{L}_{n-1}$ и $\boldsymbol{\theta}_{n-1}$.

Перейдем к способам формирования матрицы $\boldsymbol{\Gamma}_n$. Показано, что для линейных систем алгоритмы МОН, определенные последовательностью (17) при любом начальном значении $\boldsymbol{\theta}_0$, будут сходиться к истинному значению $\boldsymbol{\theta}^*$, если выполнено условие

$$\sup_n \| \mathbf{E} - \boldsymbol{\Gamma}_n \mathbf{R} \| = \lambda < 1,$$

где $\mathbf{R} = \lim_{n \to \infty} \mathbf{R}_n$; E — единичная матрица $m \times m$.

Условие $\lambda < 1$ допускает выбор постоянной матрицы $\boldsymbol{\Gamma}_n$, не зависящей от n. В частности, если $\mathbf{R}$ невырожденная матрица, можно взять $\boldsymbol{\Gamma}_n = \mathbf{R}^{-1}$. Однако на конечной реализации можно знать $\mathbf{R}$ только приближенно.

Выбор матрицы $\mathbf{R}_n^{-1}$ неудобен из-за сложности вычисления оценок, поскольку этот алгоритм приводит к необходимости расчета всех элементов $\mathbf{R}_n^{-1}$ и пересчета этих элементов вдоль выборки. Заметим, что выбор $\mathbf{R}_n^{-1}$ приведет к алгоритму, аналогичному рекуррентному МНК. Для простоты численной реализации метода потребуем, чтобы матрица $\boldsymbol{\Gamma}_n$ была диагональной. Найдем диагональные элементы матрицы $\boldsymbol{\Gamma}_n$, обеспечивающие минимум квадратичной нормы матрицы $\mathbf{V}_n = \mathbf{E} - \boldsymbol{\Gamma}_n \mathbf{R}_n$.

Минимум такой нормы обеспечит на каждом шаге максимальное уменьшение вектора невязки.

Запишем выражение для квадратичной нормы матрицы $\mathbf{V}_n$:

$$\|\mathbf{V}_n\|^2 = \sum_{i=1}^{m} (e^{(i)} - g_n^{(i)} r_n^{(i)}) (e^{(i)} - g_n^{(i)} r_n^{(i)})^{\mathrm{T}},$$

где $e^{(i)}$, $r_n^{(i)}$ — i-е строки матриц $\mathbf{E}$; $\mathbf{R}_n$ — соответственно на шаге n, $g_n^{(i)}$ — диагональный элемент матрицы $\boldsymbol{\Gamma}_n$.

Минимизация этой нормы позволяет определить диагональные элементы $\gamma_n^{(i)}$ матрицы $\boldsymbol{\Gamma}_n$.

Из условия $\partial \|\mathbf{V}_n\|^2 / \partial g^{(i)} = 0$ находим систему уравнений (в данном

случае независимых). Решение этой системы позволяет найти диагональные элементы $\mathbf{\Gamma}_n$ в виде

$$\gamma_n^{(i)} = r_n^{(ii)} / \sum_{j=1}^{m} [r_n^{(ij)}]^2. \qquad (18)$$

Аналогичные матрицы коэффициентов усиления, умноженные на $1/n$, были получены для алгоритмов СА, т.е. без осреднения невязок, как квазиоптимальные алгоритмы для диагональных матриц $\mathbf{\Gamma}$.

При найденных значениях $\gamma_n^{(i)}$ квадратичная норма матрицы $\mathbf{V}$ вычисляется по формуле

$$\lambda_n \triangleq \|\mathbf{V}_n\|^2 = \sum_{i=1}^{m} \left(1 - \frac{[r_n^{(ii)}]^2}{\sum_{j=1}^{m} [r_n^{(ij)}]^2} \right). \qquad (19)$$

Эта величина может оказаться больше 1. Тогда необходимо изменить элементы матрицы $\mathbf{\Gamma}_n$ таким образом, чтобы обеспечить $\lambda_n < 1$. В некоторых случаях целесообразно перейти к другой норме, потребовав, например, $\lambda_1 \triangleq \sum_{j=1}^{m} |r^{(ij)}| < 1$. При этом $0 < \gamma_i < 2 / \sum_{j=1}^{m} |r^{(ij)}|$.

Если имеется априорная информация о диапазоне изменения неизвестного параметра θ^*, то вместо алгоритма (17) естественно пользоваться алгоритмом с ограничениями на параметры. Утверждение о сходимости сохраняется для алгоритмов этого вида.

В алгоритмах МОН нет необходимости пересчета матрицы $\mathbf{\Gamma}_n$ на каждом шаге. Такой пересчет имеет смысл проводить только в случае существенных (более чем на 20 %) изменений вторых моментов. Поэтому целесообразно в начале процесса распознавания, когда оценки вторых моментов существенно изменяются, проводить пересчет матриц $\mathbf{\Gamma}_n$ по формулам (18) через 15-20 шагов до 50-60-го шага. В дальнейшем достаточно проводить такой пересчет через 100 шагов или по сигналу о существенном изменении корреляционного момента.

В случае, когда трудно получить значение $\lambda_n < 1$, имеет смысл перейти к циклическим алгоритмам типа Зайделя. При этом на каждом цикле, состоящем из нескольких (от 5 до 15) шагов, выполняется распознавание только одного параметра, на следующем цикле следующего и т.д. Целесообразно несколько изменить этот алгоритм, разделив все неизвестные параметры на 2—3 группы так, чтобы для

каждой группы параметров заведомо выполнялось условие $\lambda_n^{(k)}$ (*k* – номер группы).

Сопоставление качества алгоритмов МОН с другими алгоритмами будет приведено дальше.

Нелинейные модели. Свойство МОН уменьшать с ростом выборки дисперсию помех, наложенных на функцию невязки, позволяет надеяться на возможность использования его для распознавания параметров нелинейных моделей. Если бы помехи были столь незначительны, что их влиянием можно было бы пренебречь, задача свелась бы к достаточно полно разработанным итеративным процедурам определения корней нелинейных систем уравнений.

В алгоритмах МОН для стохастических моделей помеха в системах уравнений, хотя и стремится к 0 с ростом выборки, но существует. Поэтому нельзя непосредственно воспользоваться обычными итерационными процедурами, используемыми для нахождения корней в системах без помех.

Перейдем к построению алгоритмов МОН. Для пояснения некоторых особенностей оценивания параметров нелинейных моделей сделаем несколько предварительных уточнений.

Рассмотрим сначала статическую модель с вектором неизвестных параметров $\boldsymbol{\theta}^*$ размерностью *m*. Запишем в момент времени *n* модель в виде

$$y_n = \varphi(\mathbf{x}_n, \boldsymbol{\theta}^*) + \xi_n, \qquad (20)$$

где $\{\mathbf{x}_n\}$ - последовательность входных величин, независимых между собой и одинаково распределенных; $\{\xi_n\}$ - последовательность независимых и одинаково распределенных ненаблюдаемых шумов.

Пусть, как и для линейных моделей, функция потерь записывается в виде выражения

$$\eta = \frac{1}{n} \sum_{i=1}^{n} [y_i - \varphi(\mathbf{x}_i \boldsymbol{\theta})]^2. \qquad (21)$$

Тогда в соответствии с правилами формирования алгоритмов МОН невязка формируется из условия: $\mathbf{S}_n = d\eta/d\theta = 0$. Отсюда

$$s_n^{(j)} = \frac{1}{n} \sum_{i=1}^{n} y_i \left.\frac{d\varphi}{d\theta^{(j)}}\right|_{\boldsymbol{\theta}_{i-1}} - \frac{1}{n} \sum_{i=1}^{n} \varphi(\mathbf{x}_i \boldsymbol{\theta}_{i-1}) \left.\frac{d\varphi}{d\theta^{(j)}}\right|_{\boldsymbol{\theta}_{i-1}}$$

или

$$\mathbf{S}_n = \mathbf{L}_n - \mathbf{Q}_n. \qquad (22)$$

Для построения алгоритма определения $\mathbf{L}_n$ и $\mathbf{Q}_n$ запишем рекуррентную процедуру построения одной из этих статистик в виде

$$L_n^{(j)}(\boldsymbol{\theta}_{n-1}) = \frac{n-1}{n} L_{n-1}^{(j)}(\boldsymbol{\theta}_{n-1}) + \frac{1}{n} l_n^{(j)}(\boldsymbol{\theta}_{n-1}), \quad (23)$$

где $l_n^{(j)} = y_n \left.\frac{d\varphi}{d\theta^{(j)}}\right|_{\boldsymbol{\theta}_{n-1}}$

Поскольку для нелинейных моделей статистика $\mathbf{L}_n$ зависит от параметров, изменяющихся вдоль выборки, будем на каждом шаге пересчитывать статистику с помощью линейного приближения степенного ряда функции $\mathbf{L}(\boldsymbol{\theta})$. Это возможно только в случае, если статистики $\mathbf{L}_n$ и $\mathbf{Q}_n$ не имеют скачков в диапазоне изменения параметров, т.е. $\partial \mathbf{L}/\partial\theta^{(i)} < \infty$ и $\partial \mathbf{Q}/\partial\theta^{(i)} < \infty$, $i = 1, ..., m$. Алгоритм пересчета статистики записывается в виде

$$L_n^{(j)}(\boldsymbol{\theta}_n) = L_n^{(j)}(\boldsymbol{\theta}_{n-1}) + \sum_{i=1}^{m} \left.\frac{\partial L_n^{(j)}}{\partial \theta^{(i)}}\right|_{\boldsymbol{\theta}_{n-1}} \Delta\theta_n^{(i)}, \quad (24)$$

где $\Delta\theta_n^{(i)} = \theta_n^{(i)} - \theta_{n-1}^{(i)}$, $i = 1, ..., m$.

Таким образом, статистики пересчитываются на каждом шаге не только по новым данным $\mathbf{x}_n$, y_n, в соответствии с (23), но и на новое значение параметра $\boldsymbol{\theta}_n$. При формировании вектора невязок (22) по приведенным в (23) и (24) соотношениям алгоритм МОН для нелинейных систем записывается в виде (22).

Возможны некоторые модификации алгритмов МОН для нелинейных моделей. В каждой из модификаций алгоритмы нахождения матрицы $\boldsymbol{\Gamma}_n$ различны.

Рассмотрим две наиболее работоспособных модификации, которые обозначим $МОН_Н2$ и $МОН_Н3$, где индекс "н" указывает на нелинейность модели. Введем цикл, состоящий из *c* шагов. В отличие от порядкового номера реализации *n* номер шага с начала цикла будем обозначать нижним индексом ν ($\nu = 1, 2, ..., c$). Нижним индексом в скобках *(k)* будем обозначать номер цикла $(k) = n/c + 1$ — целая часть. В алгоритмах $МОН_Н2$ уравнение (20) записывается в виде

$$y_{\nu(k)} = \varphi(\mathbf{x}_{\nu(k)}, \boldsymbol{\theta}_{(k-1)}) + \sum_{i=1}^{m} (\partial\varphi/\partial\theta^{(i)})|_{\boldsymbol{\theta}_{(k-1)}} A_{\nu(k)}^{(i)} + \Delta\varphi_\nu + \xi_\nu, \quad (25)$$

где $\Delta\varphi_\nu$ - сумма старших членов разложения функции *y* в ряд;

$\mathbf{A}_{\nu(k)}$ - поправка на шаге ν цикла k вектора $\boldsymbol{\theta}_{(k-1)}$, т.е. $\boldsymbol{\theta}_{\nu(k)} = \boldsymbol{\theta}_{k-1} + \mathbf{A}_{\nu(k)}$.

Примем $\mathbf{A}_{0(k)} = 0$ и обозначим

$$\left.\begin{aligned} &\Delta\varphi_{\nu} + \xi_{\nu} = \eta_{\nu}; \\ &\partial\varphi(x_{\nu(k)}\boldsymbol{\theta})/\partial\theta^{(i)}\Big|_{\boldsymbol{\theta}_{(k-1)}} = u^{(i)}_{\nu(k)}; \\ &y_{\nu(k)} - \varphi(x_{\nu(k)}, \boldsymbol{\theta}_{(k-1)}) = w_{\nu(k)}. \end{aligned}\right\} \qquad (26)$$

В этих обозначениях (25) перепишется в виде

$$w_{\nu(k)} = \sum_{i=1}^{m} A^{(i)}_{\nu(k-1)}\, u^{(i)}_{\nu(k)} + \eta_{\nu}. \qquad (27)$$

В (27) помехи η_ν не являются независимыми от $\mathbf{u}_{\nu(k)}$. Тем не менее определим оценки $\mathbf{A}_\nu$ с помощью алгоритма МОН для линейных систем с независимым шумом. После цикла, состоящего из c шагов, определим $\boldsymbol{\theta}_{(k)} = \boldsymbol{\theta}_{(k-1)} + \mathbf{A}_{c(k-1)}$. Пересчет вторых моментов по формулам (24) будем выполнять также после каждого цикла, при этом $\Delta\theta^{(i)}_{n} = A^{(i)}_{c(k)}$. Таким образом, расчет вторых моментов внутри цикла осуществляется при одном и том же значении θ. Расчет матрицы $\boldsymbol{\Gamma}_{(k)}$, постоянной на каждом цикле в алгоритмах $\text{МОН}_\text{Н}2$, осуществляется по формулам (18), поскольку система (27) является линейной, хотя и с зависимыми регрессорами и шумом.

Вместо матрицы **R** в (17) в $\text{М0Н}_\text{н}2$ используется матрица вида $\mathbf{P}_{(k)}$

$$\mathbf{P}_{(k)} = [\rho^{(ij)}_{(k)}] = [d\varphi/d\boldsymbol{\theta}|_{\boldsymbol{\theta}_{(k-1)}}]^{\mathrm{T}}\,[d\varphi/d\boldsymbol{\theta}|_{\boldsymbol{\theta}_{(k-1)}}], \qquad (28)$$

где $d\varphi/d\boldsymbol{\theta}$ — вектор-строка.

Таким образом, модификация $\text{М0Н}_\text{н}2$ представляет собой поиск корней функции S путем линейного приближения этой функции. Точка, в которой производится разложение, на каждом цикле изменяется, приближаясь к значению корня $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. Если $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, то $M\{\mathbf{A}_\nu\} = 0$, поскольку $M\{y_n - \varphi(x_n, \theta^*)u_n(\theta^*)\} = 0$, и, следовательно, в этой точке процесс поиска корней останавливается. Ниже будет показано, что на исследованных примерах $\text{М0Н}_\text{н}2$ обеспечивал в широком диапазоне начальных условий хорошую сходимость параметров к истинным значениям. Однако этот алгоритм нуждается в целом ряде ограничений, поскольку случайные отклонения переменных и существенно различные градиенты по

разным переменным приводит к трудностям в процессе поиска. К наиболее действенным ограничениям, значительно сократившем длину реализации, относятся следующие.

1. Ограничение составляющих матрицы $\mathbf{\Gamma}_{(k)}$:

$$\gamma^{(i)} \leqslant 2,\ i = 1, ..., m.$$

2. После пересчета в конце цикла вектора **L** на новое значение параметра должно выполняться неравенство

$$\|\mathbf{L}_{0\,(k)}\| \leqslant \|\mathbf{L}_{c\,(k-1)}\|.$$

Если это неравенство не выполняется, следует принять

$$L_{0\,(k)}^{(i)} = \begin{cases} L_{c\,(k-1)}^{(i)}, & \text{если } |L_{0\,(k)}^{(i)}| > |L_{c\,(k-1)}^{(i)}|; \\ 0, & \text{если } \operatorname{sign} L_{0\,(k)}^{(i)} \neq \operatorname{sign} L_{c\,(k-1)}^{(i)}; \\ L_{0\,(k)}^{(i)}, & \text{если } |L_{0\,(k)}^{(i)}| \leqslant |L_{c\,(k-1)}^{(i)}|. \end{cases}$$

$i = 1, ..., m.$

Значение приращений к вектору $\boldsymbol{\theta}_{(k-1)}$ определяется из решения относительно $A^{(i)}$ системы линейных уравнений

$$L_{0\,(k)}^{(i)} = L_{c\,(k-1)}^{(i)} + \sum_{j=1}^{m} \left(\frac{\partial L^{(i)}}{\partial \theta^{(j)}} \right) \Bigg|_{\boldsymbol{\theta}_{k-1}} A^{(j)},\ i = 1, ..., m.$$

Это ограничение вводится в связи с тем, что норма вектора **L** должна уменьшаться, если значение θ приближается к значению корня θ*.

3. Вектор **u** представляет собой вектор производных функций φ. Для конкретных моделей функция φ может не быть заданной в явном виде, и значение ее производных аналитически определить трудно. Определение производных в этом случае осуществляется путем введения специальных дополнительных приращений к вектору $\theta_{(k)}$ по каждому направлению и расчету вторых моментов в соответствующих дополнительных точках. Но случайные отклонения переменных $\mathbf{x}_n$, y_n могут привести к существенным выбросам в рассчитанных значениях производных. Поэтому целесообразно ограничить мгновенное значение l_ν в (23) условием $l_{\nu\,(k)}^{(i)} \leqslant 5L_{0\,(k)}^{(i)}$. Тоже ограничение следует ввести на $r_{\nu\,(k)}^{(ij)}$.

4. Наконец, обязательно введение ограничений на параметры $\boldsymbol{\theta} \in D$, где D — заданная ограниченная область в m-мерном пространстве.

Модификация алгоритмов МОН$_н$3 близка по своему содержанию к алгоритмам метода Ньютона для приближенного решения систем нелинейных уравнений.

Эти алгоритмы используются обычно для нахождения корней нелинейных систем уравнений без помех. Суть алгоритмов Ньютона заключается в следующем. Пусть задана нелинейная система уравнений вида $f(\boldsymbol{\theta}^*) = 0$, где f — вектор-функция размерности m; θ^* — вектор неизвестных параметров той же размерности. Корни этого уравнения, т.е. искомые параметры θ^*, могут быть найдены с помощью итеративной процедуры вида

$$\boldsymbol{\theta}_{l+1} = \boldsymbol{\theta}_l + \Gamma(\boldsymbol{\theta}_l) f(\boldsymbol{\theta}_l). \qquad (29)$$

В алгоритмах Ньютона $\Gamma(\boldsymbol{\theta}_l) = D^{-1}$, где D - матрица частных производных размером $[m \times m]$.

Элементы этой матрицы

$$[d^{(ij)}(\boldsymbol{\theta}_l)] = \partial f^{(i)} / \partial \theta^{(j)}. \qquad (30)$$

В задаче распознавания искомые параметры являются корнями уравнения регрессии $M\{S(\boldsymbol{\theta})\} = 0$. Рекуррентная процедура расчета статистик процесса и итеративная процедура поиска корней осуществляются одновременно. При этом по мере роста выборки уменьшается дисперсия помех, налагаемых на функцию регрессии, т.е. семейство функций $S(\theta)$ стягивается в единственную функцию $M\{S(\boldsymbol{\theta})\}$. Воспользовавшись этим обстоятельством, применим алгоритмы Ньютона для определения корней уравнения $S(\boldsymbol{\theta}) = 0$.

Для защиты этой процедуры от случайных помех введем, как и в алгоритмах МОН$_н$2, дополнительные ограничения, которые обычно обеспечивают сходимость процедуры, несмотря на случайный характер функций. Как и в алгоритмах МОН$_н$2, введем циклы постоянной длины вдоль выборки (на длине цикла параметры расчета статистик постоянны), ограничения на статистики и параметры, аналогичные ограничениям 1—4 в алгоритмах МОН$_н$2.

В обозначениях МОН$_н$3 матрица R отсутствует, вектор L_n из МОН$_н$2 заменяется вектором S_n в МОН$_н$3.

Целесообразно и для МОН$_н$3 перейти к диагональной матрице $\boldsymbol{\Gamma}$. Для этого необходимо найти диагональную матрицу F, отвечающую условиям $\mathbf{D}^{-1}\mathbf{S} = \mathbf{FS}$. Отсюда если $\mathbf{D}$ неособенная матрица, то $\mathbf{DFS} = \mathbf{S}$. Обозначим вектор $\mathbf{FS} = \mathbf{B}$. Этот вектор определяется решением линейной системы $\mathbf{DB} = \mathbf{S}$. Диагональные элементы матрицы $\mathbf{F}$ находятся из соотношений $f^{(i)} = b^{(i)} / s^{(i)}$. Значения составляющих вектора $\mathbf{B}$ при расчетах следует ограничивать.

В алгоритмах МОН$_Н$2 и МОН$_Н$3 для пересчета статистик после окончания цикла необходимо вычисление частных производных от этих статистик по параметрам. С этой целью на каждом цикле производится расчет статистик не только в точке $\boldsymbol{\theta}_{(k)}$,* но и в сдвинутых на выбранные значения $\Delta^{(i)}$ точках $i = 1, ..., m$. Расчет производных по полученным статистикам очевиден.

Заметим, что с точки зрения расчета производных алгоритм МОН$_Н$3 существенно, в m-1 раз экономичней алгоритма МОН$_Н$2.

Дальнейшее сокращение времени счета достигается путем перехода к алгоритмам Зейделя. Они обозначаются в дальнейшем МОН$_Н$2 S, МОН$_Н$3 S. Если в приведенных выше алгоритмах изменение вектора параметров после окончания цикла осуществлялось сразу по всем составляющим, то в алгоритмах Зейделя на каждом цикле осуществляется изменение только одной координаты, на следующем цикле другой и т.д., в любой заданной последовательности.

Проигрыш в скорости сходимости, возникающий благодаря замораживанию координат вектора параметров, в значительной степени компенсируется увеличением коэффициента усиления по каждой координате. Алгоритмы с покоординатным поиском имеют также преимущества по экономии машинного времени.

Если на цикле изменяется только одна составляющая вектора 0, то соответствующий коэффициент усиления $\gamma_{(k)}^{(i)}$ в диагональной матрице $\boldsymbol{\Gamma}_{(k)}$ вычисляется для МОН$_Н$2 S по формулам

$$\gamma_k^{(i)} = [\rho^{(ii)}|_{\boldsymbol{\theta}(k-1)}]^{-1}, \qquad (31)$$

$\rho^{(ii)}$ см. (28). Для МОН$_Н$3 S

$$\gamma_{(k)}^{(i)} = (\partial s^{(i)}/\partial \theta^{(i)}|_{\boldsymbol{\theta}(k-1)})^{-1}. \qquad (32)$$

Для алгоритмов Зейделя нет необходимости проверять условие сходимости по формулам (19). Введенные ограничения на параметры и статистики должны выдерживаться и для этих алгоритмов.

О сходимости алгоритмов МОН. Выше говорилось о сходимости алгоритмов МОН вида (17) для линейных систем. Сходимость этих алгоритмов обеспечивается при любых начальных условиях, если система устойчива, аддитивный ненаблюдаемый шум и регрессоры с неизвестными коэффициентами независимы и выполняется условие

$$\sup_n \|\mathbf{E} - \boldsymbol{\Gamma}_n \mathbf{R}\| = \lambda < 1.$$

Перечисленные условия позволяют выбирать постоянную матрицу Γ_n и, как показывают проведенные исследования (см. далее), получать хорошие оценки параметров.

Для нелинейных моделей нет оснований предполагать, что алгоритм МОН будет сходиться к истинному значению θ* при любом начальном значении $\theta_{(0)}$. Для этого случая удалось доказать только, что есть некоторая область около точки θ*, попадание в которую гарантирует алгоритмам МОН сходимость к θ*.

Сходимость доказана при следующих предположениях.

1. $\rho^{(ij)}(\theta) < \infty$ $(i, j = 1, 2, ..., m)$; $M(|\mathbf{L}_n|) < \infty$; $M(|\mathbf{Q}_n|) < \infty$, где $\rho^{(ij)}(\theta)$ — элементы матрицы $P(\theta) = M\{(d\varphi/d\theta)^{\mathrm{T}}(d\varphi/d\theta)\}$; $d\varphi/d\theta$ — вектор-строка; обозначения $\mathbf{L}_n$ и $\mathbf{Q}_n$ см. в (22).

2. Ограничены первые четыре производные функции φ, т.е.

$$\psi(q, i) = \partial^q \varphi/(\partial\theta^{(i)})^q < \infty, \quad i = 1, 2, ..., m; \quad q = 1, 2, 3, 4.$$

3. Известно, что параметры могут изменяться в ограниченных пределах $\theta \in \mathbf{D}$, где **D** — известная конечная область пространства размерности *m*.

Алгоритм МОН записывается в виде

$$\theta_{n+1} = \mathrm{pr}_{\mathbf{D}}\{\theta_n + \Gamma_{n+1}[\mathbf{L}_{n+1}(\theta_n) - \mathbf{Q}_{n+1}(\theta_n)]\}, \quad (33)$$

где $\mathrm{pr}_{\mathbf{D}}$ (проекция на **D**) означает, что параметры θ могут изменяться только в границах **D**.

При этих ограничениях справедлива теорема:

Пусть матрица Γ_{n+1} в алгоритме (33) такова, что

$$\sup_n \|\mathbf{E} - \Gamma_{n+1} P(\theta)\| = \lambda < 1.$$

Тогда имеется такая область $\mathbf{D}' \subset \mathbf{D}$, содержащая θ *, что оценки (33) сходятся в θ* с вероятностью 1 (почти наверное).

К сожалению, из этой теоремы не удается извлечь информацию для конструирования области **D′**. Поэтому определение области, в которой оценки сходятся, проводилось экспериментально, путем статистического моделирования. Ниже приведены некоторые результаты статистического моделирования.

Эксперименты проводились с целью нахождения тех начальных значений $\theta_{(0)}$ оценок параметров, при которых оценки θ_n с ростом выборки сходятся к истинным значениям. Под областью сходимости условно понималось множество тех $\theta_{(0)}$, при которых, начиная с некоторого n<900 и до n=1000, выполнялось неравенство $\|\theta_n - \theta^*\| < 0{,}1\|\theta^*\|$.

Оценивание параметров проводилось с помощью программы, реализующей различные модификации алгоритмов МОН ($МОН_Н2$ G, $МОН_Н2$ S, $МОН_Н3$S).

Исследовались следующие модели:

1. $y_n = e^{\theta^* x_n} + \sigma\epsilon_n$, $\sigma = 0{,}1, 0{,}5, 1$; $x_n, \epsilon_n \sim N(0, 1)$, $\theta^* = 0{,}5$, $0 \leqslant \theta \leqslant 1$.

2. $y_n = e^{\theta_1^* x_n} + 4/(x_n + \theta_2^*) + \sigma\epsilon_n$.

3. $y_n = \theta_3^* e^{\theta_1^* x_n} + \theta_4^*/(x_n + \theta_2^*) + \sigma\epsilon_n$.

4. $y_n = e^{\theta_1^* y_{n-1}} + 4/(x_n + \theta_2^*) + \sigma\epsilon_n$

5. $\begin{cases} y_n = e^{\theta_1^* x_n} + 4/(x_{n-1} + \theta_2^*) + \sigma\epsilon_n; \\ x_n = b^* x_{n-1} + \epsilon_{1n}. \end{cases}$

Для моделей 2–5 $\theta_1^* = 0{,}5$, $\theta_2^* = 5$, $\theta_3^* = 1$, $\theta_4^* = 4$, $b^* = 0{,}8$;

$$\mathbf{D} = \left\{ \begin{array}{l} \theta_1,\ 0{,}2 \leqslant \theta_1 \leqslant 1 \\ \theta_2,\ 2 \leqslant \theta_2 \leqslant 18 \\ \theta_3,\ 0 \leqslant \theta_3 \leqslant 5 \\ \theta_4,\ 0 \leqslant \theta_4 \leqslant 10 \end{array} \right\}, \quad x_n, \epsilon_n, \epsilon_{1n} \sim N(0, 1), \quad \sigma = 0{,}5,$$

$\operatorname{cov}(\epsilon_n, \epsilon_{1n}) = 0$.

Осреднялись результаты расчета по 10 выборкам длиной 1000. Кроме вышеуказанных ограничений на параметры вводились также ограничения на мгновенные значения корреляционных функций $\rho^{(ij)} \leqslant 100$. Во всех алгоритмах, кроме модели 1, использовался циклический процесс распознавания с длиной цикла $c = 10$. Области **D** и **D'** для различных модификаций $МОН_Н2$ приведены на рис. 2.

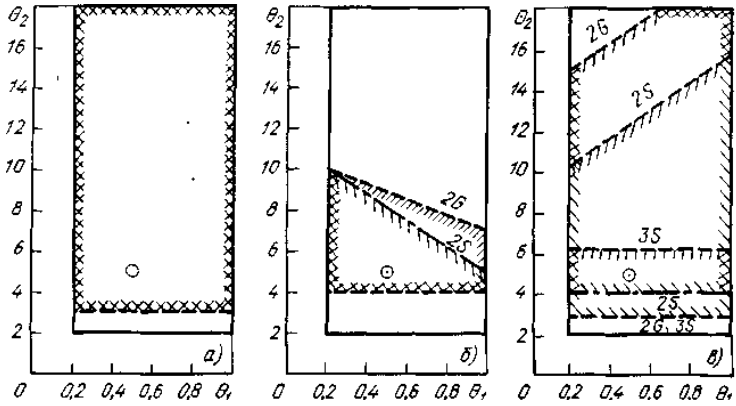


Рис. 2. Области сходимости алгоритмов МОН для нелинейных моделей 2 *(а)*, 4 *(б)* и 5 (*в*)

Для модели 3 область сходимости **D′** оказалась сравнительно небольшой. Отклонение одновременно по всем переменным за диапазон $0{,}4 \leqslant \theta_1 \leqslant 0{,}6,\ \ 4 \leqslant \theta_2 \leqslant 6,\ \ 0{,}8 \leqslant \theta_3 \leqslant 1{,}2,\ \ 3{,}9 \leqslant \theta_4 \leqslant 4{,}1$ вызывало в большинстве случаев расхождение процесса распознавания. Количество одновременно распознаваемых параметров существенно зависит от характера нелинейной модели. Тем не менее при современном состоянии разработки МОН не следует вести распознавание более трех параметров нелинейной модели.

Сопоставление различных модификаций алгоритмов МОН приводит к следующим выводам: наилучшие результаты были получены для моделей 2, 4, 5 с помощью алгоритма $МОН_Н3$ S. Однако этот алгоритм проигрывает алгоритму $М0Н_Н2$ S по надежности попадания в область сходимости (соответственно 78 и 65 %) на выборке из 450 реализаций, включая эксперименты на границах области сходимости.

Скорость сходимости алгоритмов МОН характеризуется для модели 2 табл 1, показывающей длину выборки, на которой оба параметра входят в 10-процентную зону и больше не выходят из нее.

Таблица 1

θ_1	θ_2					
	3	6	8	10	12	16
0,2	780	160	160	200	500	740
0,6	780	20	20	460	780	900
0,8	780	360	320	360	620	860

11.9. Оценка различных методов

Ниже рассмотрены различные показатели качества распознавания и на конкретных моделях приведены результаты сопоставления по этим показателям различных методов, рассмотренных выше.

11.9.1. Показатели качества

Задача распознавания для детерминированных систем часто ставится как задача отыскания такого аналитического выражения модели, которое в l заданных точках совпадает с экспериментально полученными характеристиками распознаваемого объекта при заданных входных воздействиях. Если l - число распознаваемых параметров, то задача распознавания сводится к задаче отыскания решений некоторой системы уравнений. В большинстве случаев и для детерминированных моделей количество экспериментальных значений гораздо больше числа неизвестных параметров. Тогда необходимо выполнить аппроксимацию экспериментальных точек аналитическим выражением по некоторому критерию близости. Задача распознавания сводится к некоторой экстремальной задаче минимизации этого критерия.

В стохастической постановке задача распознавания также сводится к экстремальной задаче минимизации критерия качества оценивания. Но, если известны или предполагаются известными статистические характеристики возмущений, действующих на распознаваемую систему, часто удается решить несколько дополнительных задач. К этим задачам относятся, в частности, исследование поведения критерия качества в асимптотике, т.е. при осреднении на реализациях бесконечной длины. Асимптотическое поведение систем в ряде случаев удается исследовать аналитически.

Выше кратко упоминались основные результаты, полученные для стохастических моделей в асимптотике. В промышленных условиях чаще всего приходится иметь дело с небольшими выборками. Характеристики распознаваемого объекта, т.е. параметры модели, изменяются. Кроме того, задача распознавания для промышленных условий обычно является многокритериальной: кроме очевидного требования близости характеристик модели и объекта имеется целый ряд других требований, имеющих часто определяющее значение. К таким требованиям можно отнести следующие: алгоритмы распознавания должны быть достаточно "грубыми", т.е. не быть очень "чувствительными" к точности настроек самого алгоритма

распознавания, поскольку эти настройки определяются неизвестными характеристиками объекта и возмущений.

Требуется также, чтобы в ходе распознавания не было резких изменении характеристик модели. Изменение характеристик модели вызывает изменение воздействий распознавания. Следовательно, если такие изменения вызваны свойствами алгоритма распознавания, а не действительным изменением параметров объекта, это неминуемо приводит к "перерегулированию", дополнительным возмущениям распознаваемого объекта. Численное значение, характеризующее эти изменения оценок параметров вдоль выборки, будем называть в дальнейшем *изменчивостью* оценок параметров.

Наконец, самому понятию близости характеристик объекта и модели также можно придавать различный смысл. Наиболее распространенными трактовками этих понятий являются близость в пространстве выходных переменных состояния и близость в пространстве оцениваемых, распознаваемых параметров. Если под близостью понимается среднеквадратическое отклонение, то для линейных моделей с аддитивным нормальным шумом в соответствии с п. 11.2 МНК в асимптотике обеспечивает минимум в обоих названных выше пространствах. К сожалению, асимптотические свойства методов далеко не всегда характеризуют их поведение на конечных выборках. Поскольку аналитические расчеты, позволяющие оценить свойства методов на конечных выборках, чаще всего невыполнимы, обычно для исследования свойств методов прибегают к статистическому моделированию, проводят серию экспериментов на различных выборках с одинаковыми статистическими свойствами и рассматривают осредненные результаты таких исследований.

Ниже приведены результаты одного из исследований, проведенного для линейных моделей с нормальным шумом с целью сопоставления различных методов на конечных выборках. Сравнение проводилось по следующим показателям качества (критериям) для моделей, записанных в виде $y_n = Z_n^T \theta^* + \xi_n$, где Z_n — вектор наблюдений; θ^* — вектор распознаваемых параметров; ξ_n — ненаблюдаемый шум:

1) среднеквадратическое отклонение выходной переменной y от прогнозируемого значения (близость в пространстве выходных переменных)

$$\eta_{1n} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (y_k - Z_k^1 \theta_k)^2} \; ; \qquad (1)$$

2) среднеквадратическое отклонение оценок параметров от истинных значений (близость в пространстве параметров)

$$\eta_{2n} = \sqrt{\frac{1}{nm} \sum_{k=1}^{n} \sum_{i=1}^{m} \left(\frac{\theta_k^{(i)} - \theta^{(i)*}}{\theta^{(i)*}} \right)^2}; \qquad (2)$$

3) изменчивость оценок параметров вдоль выборки в зависимости от n

$$\eta_{3n} = \sqrt{\frac{1}{nm} \sum_{k=1}^{n} \sum_{i=1}^{m} \left(\frac{\theta_k^{(i)} - \theta_{k-1}^{(i)}}{\theta^{(i)*}} \right)^2}. \qquad (3)$$

В (2) и (3) предполагается,что $\theta^{(i)*} \neq 0$.

После проведения серии из q экспериментов, в каждом из которых использовались новые реализации случайных входных и возмущающих воздействий с теми же распределениями, результаты осреднялись. Средние значения показателей обозначены соответственно $\overline{\eta}_{1n}, \overline{\eta}_{2n}, \overline{\eta}_{3n}$.

Кроме средних значений по q экспериментам рассчитываются также выборочные дисперсии случайных величин η_{1n}, η_{2n} и η_{3n}, обозначаемых соответственно $D_{\eta_1}, D_{\eta_2}, D_{\eta_3}$.Оценивается также время расчета оценок на вычислительной машине $t_{\text{м}}$, характеризующее в некоторой степени сложность алгоритма.

11.9.2. Модели и основные результаты сопоставления

Приведем вначале иллюстративные материалы, характеризующие поведение различных алгоритмов. Графики, полученные по одной реализации, поясняют результаты статистического моделирования, приведенные ниже.

П р и м е р 1. Рассмотрена модель вида

$$y_n = -1{,}4\, y_{n-1} - 0{,}48\, y_{n-2} - 2x_{n-1} - 0{,}8x_{n-2} + \epsilon_n;$$

$$x_n \sim N(0;\ 0{,}5);$$

$$\epsilon_n \sim N(0,\ 1);$$

$$\operatorname{cov}(x_{n-1}, \epsilon_n) = \operatorname{cov}(x_{n-2}, \epsilon_n) = \operatorname{cov}(x_{n-1}, x_{n-2}) = 0.$$

На рис. 1 приведены графики поведения оценок МОН и рекуррентного МНК вдоль выборки. Сплошные линии - оценки МОН, штриховые - МНК. В алгоритмах МОН (11 п. 11.8) для проверки "грубости" кроме матрицы $\mathbf{\Gamma}_n$, вычисленной по (18 п. 11.8), были применены матрицы, увеличенные и уменьшенные в 2 раза.

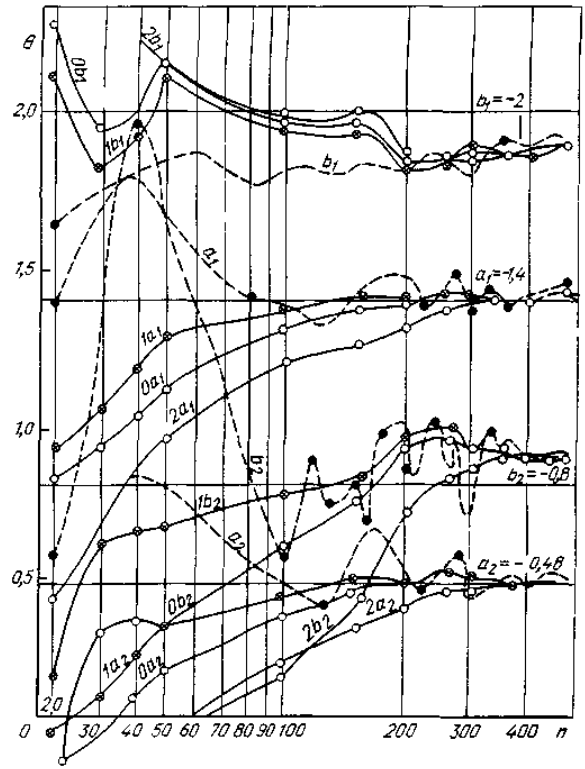


Рис. 1 Графики зависимости оценок МОН и МНК от длины выборки

На рис. 1 линии, обозначенные цифрой 0, соответствуют экспериментам, в которых расчет $\Gamma_{,,}$ проводился по формулам (18 п. 11.8). Численные значения элементов матрицы $\mathbf{\Gamma}_n$ на линиях, обозначенных цифрами 0, 1 и 2, приведены в табл. 1.

Таблица 1

Обозначение диагональных элементов матрицы Γ_n	Значения диагональных элементов для различных графиков рис 5 1			Допустимые значения элементов матрицы
	0	1	2	
$\gamma^{(1)}$	0,04	0,055	0.02	0–0,06
$\gamma^{(2)}$	0,04	0,55	0 02	0–0,06
$\gamma^{(3)}$	2,28	5	1	0–8
$\gamma^{(4)}$	0,66	1	0,3	0–1,32

Из графиков рис. 1 видно, что МОН позволяет получить лучшие приближения параметров к истинным значениям по сравнению с МНК. Эта разница особенно ощутима на сравнительно коротких выборках. В то же время с точки зрения приближения выходной переменной к ее оценке, т.е. по критерию η_{1n}, приведенному в формуле (1), МНК позволяет получить лучшее приближение, что иллюстрируется графиком на рис. 2.

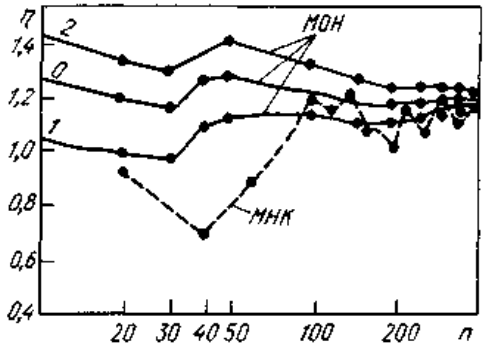


Рис 2. Графики зависимости показателя качества МОН и МНК от длины выборки

Из рис. 1 видно также, что оценки МОН изменяются вдоль выборки почти монотонно, в то время как оценки МНК имеют существенно колебательный характер. Важно отметить также, что существенные изменения коэффициентов усиления в алгоритме МОН сравнительно несущественно влияют на поведение оценок, т.е. оценки МОН достаточно "грубые".

Пример 2. Сопоставление "грубости" оценок МОН и метода стохастической аппроксимации (СА) иллюстрируется графиками на рис. 3, где приведены зависимости показателя качества η_{2n} (2) от коэффициента усиления γ для модели $y_n = a^* x_n + \epsilon_n$, $a^* = 0{,}8$.

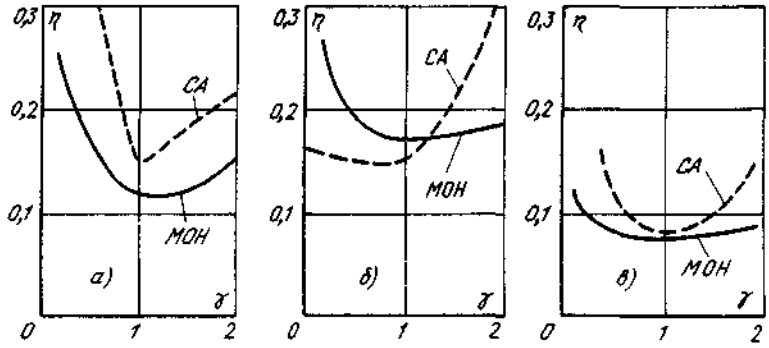


Рис. 3. Графики зависимости показателя качества η МОН и СА от коэффициента усиления γ: $а - n = 30$; $б - n = 100$; $в - n = 300$

Для этой простейшей модели алгоритм МОН записывается в виде

$$a_n = a_{n-1} + \gamma\,(L_n - a_{n-1} R_n),$$

где

$$L_n = \frac{n-1}{n} L_{n-1} + (1/n) x_n y_n;\ R_n = \frac{n-1}{n} R_{n-1} + (1/n) x_n^2;$$

$L_0 = R_0 = 0$, а алгоритм СА в виде

$$a_n = a_{n-1} + (\gamma/n)\,(y_n x_n - a_{n-1} x_n^2).$$

Оптимальные значения $\gamma = 1$ для обоих алгоритмов вычисляют в соответствии с (18 п.11.8).

На рис. 3 приведены графики, полученные при различных длинах реализаций *(n* = 30, 100, 300). Из графиков видно, что МОН "грубее" к изменению параметров, чем СА.

Статистическое моделирование. Ниже приведены результаты статистического моделирования, проведенного с целью сопоставления различных рекуррентных методов распознавания.

Статистическим испытаниям подвергались модели в разомкнутом контуре и в замкнутой системе.

Исследование проводилось на моделях, уравнения которых приведены в табл. 2.

Таблица 2

Обозначение модели	Уравнение модели	Примечание
1	$y_n = 0{,}8\, y_{n-1} + 0{,}5\, x_{n-1} + \sigma\epsilon_n$	$\epsilon_n,\ \epsilon_{1n} \sim N(0, 1)$
1 OC_1	Модель объекта 1; модель обратной связи $x_n = -1{,}6\, y_n + \epsilon_{1n}$	$\mathrm{cov}(\epsilon_n\ \epsilon_{1n}) = 0$
1 $OC_{0,2}$	Модель объекта 1; модель обратной связи $x_n = -1{,}6\, y_n + 0{,}2\, \epsilon_{1n}$	$\sigma = 1$
2	$y_n = 0{,}5\, y_{n-1} - 0{,}2\, y_{n-2} - 0{,}3\, x_{n-1} + 0{,}8\, x_{n-2} + \sigma\epsilon_n$	
2 OC_1	Модель объекта 2; модель обратной связи $x_n = -0{,}9\, y_n + 0{,}4\, y_{n-1} + \epsilon_{1n}$	
2 $OC_{0,2}$	Модель объекта 2; модель обратной связи $x_n = -0{,}9\, y_n + 0{,}4\, y_{n-1} + 0{,}2\, \epsilon_{1n}$	

Заметим, что при отсутствии шума в обратной связи, модели с обратной связью (1 ОС и 2 ОС) при $\epsilon_{1n} = 0$ оказались бы нераспознаваемы, а оценки параметров - многозначными. Поэтому уменьшение дисперсии шума в обратной связи ухудшает качество распознавания параметров системы. Следовательно, следует ожидать, что качество распознавания для моделей 1 $OC_{0,2}$ и 2 $OC_{0,2}$ хуже, чем для моделей 1 OC_1 и 2 OC_1.

В табл. 3 представлены результаты сопоставления следующих трех методов.

1. Рекуррентный МНК.
2. Стохастическая аппроксимация (СА). Матрицы $\mathbf{\Gamma}_n$ коэффициентов усиления принимались диагональными, значения диагональных элементов рассчитывались по субоптимальным алгоритмам, соответствующим выражению (9 п.11.7).
3. Метод осредненных невязок (МОИ). Матрица $\mathbf{\Gamma}_n$ коэффициентов усиления принималась диагональной, значения диагональных элементов рассчитывались по (18 п.11.8).

Таблица 3

Модель	Метод	$\overline{\eta}_{1n}$	D_{η_1}	$\overline{\eta}_{2n}$	D_{η_2}	$\overline{\eta}_{3n}$	D_{η_3}	Время
1	МНК	0,999	0,999	1,616	2,213	3,256	10,738	0,850
	СА	1,002	1,032	1,151	1,330	3,256	10,738	0,778
	МОН	1,001	0,998	1,164	1,262	1,296	1,900	0,796
$1\,OC_1$	МНК	0,999	1,002	1,616	2,371	3,344	1,107	0,850
	СА	1,002	1,019	1,151	1,332	1,273	1,911	0,778
	МОН	1,002	1,001	1,164	1,261	1,291	1,900	0,796
$1\,OC_{0,2}$	МНК	0,992	1,002	2,658	2,829	9,964	61,714	0,850
	СА	1,002	1,033	1,626	2,041	1,891	5,857	0,778
	МОН	1,003	1,017	1,766	1,890	2,17	8,286	0,796
2	МНК	0,999	1,001	1,553	3,453	2,957	11,579	0,748
	СА	1,021	1,108	1,128	1,919	0,968	1,605	0,564
	МОН	1,016	1,000	1,065	1,209	0,968	1,553	0,600
$2\,OC_1$	МНК	0,997	1,000	1,390	2,324	2,341	6,23	0,748
	СА	1,128	1,996	1,889	2,963	0,832	1,503	0,564
	МОН	1,035	1,062	1,128	1,125	0,827	1,542	0,600
$2\,OC_{0,2}$	МНК	1,025	4,485	4,518	38,55	20,21	56,10	0,748
	СА	1,032	1,167	1,124	1,707	0,848	1,838	0,564
	МОН	1,017	0,996	0,879	0,908	0,847	1,931	0,600

Все эти методы сопоставлялись по всем показателям с фильтром Калмана (ФК), в котором параметры шума, в отличие от других методов, считались известными. Поэтому фильтр Калмана мог обеспечить лучшее качество оценивания по сравнению с другими методами и принимался в качестве оптимального алгоритма.

Сравнение проводилось по следующим показателям: $\overline{\eta}_{1n}$, $\overline{\eta}_{2n}$, $\overline{\eta}_{3n}$, D_{η_1}, D_{η_2}, D_{η_3} и времени расчета. В табл. 3 приведены отношения указанных показателей к соответствующим показателям фильтра Калмана. Осреднение выполнено по $q = 50$ реализациям, каждая длиной $n = 200$ точек.

Рассматривая результаты этих испытаний, легко заметить, что даже при нормальном шуме МНК, обеспечивая незначительный выигрыш по показателю $\overline{\eta}_{1n}$, существенно (в некоторых случаях почти на два порядка) проигрывает другим методам по остальным показателям. МОН и СА практически по всем показателям, кроме $\overline{\eta}_{1n}$, превосходят МНК. Показатель $\overline{\eta}_{1n}$ для МОН не более чем на 4 %, а для СА не более чем на 13 % хуже показателя МНК.

Наиболее существенно отличие МНК от МОН и СА по показателю $\overline{\eta}_{3n}$, т.е. по изменчивости оценок и по выборочной дисперсии этого

показателя (D_{η_3}). Эти величины в МНК в 2—4, иногда в 20 раз хуже, чем в МОН и СА.

Отметим, что на преимущество осреднения невязок в смысле изменчивости оценок указывалось еще ранее в различных литературных источниках.

МОН и СА близки в смысле большинства показателей для простых моделей. При усложнении моделей МОН превосходит СА. Если осреднить по всем моделям и по первым шести показателям табл. 1, то средние показатели будут:

$$\eta_{\text{МНК}} = 19{,}04;\ \eta_{\text{СА}} = 1{,}81;\ \eta_{\text{МОН}} = 1{,}43.$$

Время расчета для МОН и СА на 20-40 % меньше, чем для ФК. Время расчета для СА на 5-7 % меньше, чем для МОН.

Для сопоставления МОН и ФК проводились также эксперименты, определяющие ухудшение показателей качества ФК при неточной информации о ковариации шумов (для приведенных моделей дисперсии шумов). С этой целью дисперсии шумов σ изменялись в ту и другую сторону в 2 раза без изменения алгоритмов оценивания ФК. Средний показатель качества оценивания ФК даже для простых моделей (1 и 2) по сравнению с МОН при этом оказался больше в 2,06 раза. Поэтому в промышленных условиях применение для распознавания фильтра Калмана возможно при удовлетворительной информации о ковариациях шумов.

Проводились также исследования влияния изменения дисперсии ненаблюдаемого шума на показатели качества распознавания. С этой целью в исследуемых моделях значение σ увеличивалось в 1,8 и 3 раза, МОН, СА и ФК ухудшали при этом все показатели качества примерно одинаково, в 1,5—2,5 раза. Для МНК такое увеличение ненаблюдаемого шума для некоторых моделей приводило к срыву процесса распознавания.

Заметим, что проведенные исследования справедливы для стационарных процессов, в которых параметры процесса и статистические характеристики возмущений остаются неизменными.

Для нестационарных процессов нужно модернизировать алгоритмы так, чтобы обеспечить отслеживание изменяемых параметров, поскольку по мере накопления выборки все приведенные алгоритмы начинают очень медленно реагировать на изменение параметров моделей.

11.10. Оценка параметров распознаваемых объектов

11.10.1. Аппроксимация функций совокупностью полиномов, ортогональных на системе равноотстоящих точек

Система нормальных уравнений Гаусса (5 п.11.1) дает хорошие результаты по аппроксимации функций, если число измерений достаточно велико (много больше, чем степень аппроксимирующего полинома) или ошибки измерений малы. В противном случае определитель системы оказывается близким к нулю и система становится, как говорят, плохо обусловленной. При этом возможны большие ошибки в оценке параметров аппроксимирующего полинома. В таких случаях хорошие результаты можно получить, если аппроксимировать функцию системой полиномов, ортогональных в равноотстоящих точках.

Пусть, как и ранее, имеется совокупность измерений $\mathbf{Y}_n = (y_1, y_2, \ldots, y_n)$, проведенных в равноотстоящие друг от друга моменты времени $t_1, t_2, t_3, \ldots, t_n$, так что

$$t_j = t_0 + jh, \quad j = 1, 2, \ldots, n.$$

Выберем совокупность полиномов, ортогональных на этой системе равноотстоящих точек. В качестве таких полиномов могут быть использованы, например, полиномы Чебышева.

Введем полиномы вида

$$P_i(j) = P_i\left(\frac{t_j - t_0}{h}\right) = \sum_{s=0}^{i} (-1)^s C^s_i C^s_{i+s} \frac{j^{(s)}}{n^{(s)}}, \tag{1}$$

где $i = 0, 1, 2, \ldots, k$, $j = 1, 2, \ldots, n$,

$$j^{(s)} = j(j-1)(j-2)\ldots(j-s+1),$$

$$n^{(s)} = n(n-1)(n-2)\ldots(n-s+1).$$

Легко проверить, что эти многочлены обладают дискретным свойством ортогональности, т. е.

$$\sum_{j=1}^{n} P_{i_1}(j) P_{i_2}(j) = 0, \text{ если } i_1 \neq i_2. \tag{2}$$

Теперь будем искать наилучшую в смысле метода наименьших квадратов аппроксимацию функции в виде линейной комбинации полиномов (1).

Введем многочлен степени k

$$Q_k(j)=\sum_{i=0}^{k} a_i P_i(j). \qquad (3)$$

Задача состоит в отыскании набора $(a_0, a_1, \ldots, a_k)$, минимизирующего сумму квадратов отклонений измеренных значений $(y_1, y_2, \ldots, y_n)$ от предсказываемых многочленом (3). Поэтому аналогично предыдущему

$$\mathcal{J}=\frac{1}{2}\sum_{j=1}^{n}[y_j-Q_k(j)]^2,$$

$$\frac{\partial \mathcal{J}}{\partial a_i}=\sum_{j=1}^{n}[y_j-Q_k(j)]P_i(j)=0,\ i=0,\ 1,\ 2,\ldots,\ k.$$

Отсюда

$$\sum_{j=1}^{n} Q_k(j)P_i(j)=\sum_{j=1}^{n} y_j P_i(j),\ i=0,\ 1,\ 2,\ldots,\ k. \qquad (4)$$

Подставляя в (4) $Q_k(j)$ из (3) и меняя порядок суммирования, имеем

$$\sum_{l=0}^{k} a_l \sum_{j=1}^{n} P_l(j)P_i(j)=\sum_{j=1}^{n} y_j P_i(j),\quad i=0,\ 1,\ 2,\ldots,\ k. \qquad (5)$$

В силу (2) соотношение (5) упрощается к виду

$$a_i\sum_{j=1}^{n} P_i^2(j)=\sum_{j=1}^{n} y_j P_i(j),\quad i=0,\ 1,\ 2,\ldots,\ k,$$

откуда

$$\hat{a}_i=\frac{\sum_{j=1}^{n} y_j P_i(j)}{\sum_{j=1}^{n} P_i^2(j)},\quad i=0,\ 1,\ 2,\ldots,\ k. \qquad (6)$$

Поэтому искомый аппроксимирующий многочлен, оптимальный в смысле метода наименьших квадратов, имеет вид

$$Q_k(j)=\sum_{i=0}^{k}\frac{\sum_{j=1}^{n} y_j P_i(j)}{\sum_{j=1}^{n} P_i^2(j)}P_i(j).$$

Таким образом, описанный выше метод аппроксимации функции совокупностью полиномов, ортогональных на системе равноотстоящих точек, позволяет получить численные значения коэффициентов аппроксимации, не решая системы нормальных уравнений Гаусса.

Теперь получим выражение для расчета элементов корреляционной матрицы ошибок оценок параметров. В соответствии с общим правилом

$$K_{\widehat{a}_{i_1}\widehat{a}_{i_2}} = M[(\widehat{a}_{i_1} - a_{i_1})(\widehat{a}_{i_2} - a_{i_2})] =$$

$$= M\left[\left(\frac{\sum_{j=1}^{n} y_j P_{i_1}(j)}{\sum_{j=1}^{n} P^2_{i_1}(j)} - a_{i_1}\right)\left(\frac{\sum_{j=1}^{n} y_j P_{i_2}(j)}{\sum_{j=1}^{n} P^2_{i_2}(j)} - a_{i_2}\right)\right].$$

Так как

$$y_j = \tilde{y}_j + v_j = \sum_{i=0}^{k} a_i P_i(j) + v_j,$$

то, с учетом (2), имеем

$$K_{\widehat{a}_{i_1}\widehat{a}_{i_2}} =$$

$$= M\left[\frac{\sum_{j=1}^{n}\sum_{i=0}^{k} a_i P_i(j) P_{i_1}(j) - a_{i_1}\sum_{j=1}^{n} P^2_{i_1}(j) + \sum_{j=1}^{n} v_j P_{i_1}(j)}{\sum_{j=1}^{n} P^2_{i_1}(j)} \times \right.$$

$$\left. \times \frac{\sum_{j=1}^{n}\sum_{i=0}^{k} a_i P_i(j) P_{i_2}(j) - a_{i_2}\sum_{j=1}^{n} P^2_{i_2}(j) + \sum_{j=1}^{n} v_j P_{i_2}(j)}{\sum_{j=1}^{n} P^2_{i_2}(j)}\right] =$$

$$= M\left[\frac{\sum_{j_1=1}^{n}\sum_{j_2=1}^{n} v_{j_1} v_{j_2} P_{i_1}(j_1)\, P_{i_2}(j_2)}{\sum_{j_1=1}^{n}\sum_{j_2=1}^{n} P^2_{i_1}(j_1)\, P^2_{i_2}(j_2)}\right] =$$

$$= \frac{\sum_{j_1=1}^{n}\sum_{j_2=1}^{n} K_{j_1 j_1} P_{i_1}(j_1)\, P_{i_2}(j_2)}{\sum_{j_1=1}^{n}\sum_{j_2=1}^{n} P^2_{i_1}(j_1)\, P^2_{i_2}(j_2)}, \quad i_1,\ i_2 = 0,\ 1,\ 2, \ldots,\ k. \tag{7}$$

Здесь $K_{j_1 j_2}$ — коэффициент корреляции между измерениями j_1 и j_2.

Если измерения некоррелированы, то соотношение (7) упрощается к виду

$$K_{\hat{a}_{i_1}\hat{a}_{i_2}} = \frac{\sum_{j=1}^{n} \sigma_j^2 P_{i_1}(j)\, P_{i_2}(j)}{\sum_{j_1=1}^{n}\sum_{j_2=1}^{n} P^2_{i_1}(j_1)\, P^2_{i_2}(j_2)}.$$

Наконец, для равноточных измерений имеем

$$K_{\hat{a}_l \hat{a}_l} = Д(\hat{a}_i) = \frac{\sigma^2}{\sum_{j=1}^{} P_i^2(j)}; \quad K_{\hat{a}_{i_1}\hat{a}_{i_2}} = 0,\ i_1 \neq i_2,$$

т. е. корреляционная матрица становится диагональной. Как уже отмечалось, достоинства метода особенно ощутимы, когда количество измерений невелико, а их ошибки значительны. Помимо этого отметим, что соотношения (6 п.11.1) дают явное аналитическое представление для коэффициентов аппроксимации, позволяющее провести непосредственный анализ качества аппроксимации, который был бы практически неосуществим при наличии только соотношений (5 п.11.1)или (6 п.11.1), обеспечивающих лишь численное решение поставленной задачи.

11.10.2. Рекуррентные соотношения для метода наименьших квадратов

Полученное ранее соотношение (6 п.11.1) позволяет осуществить оценку параметров функции отклика системы, используя некоторую совокупность измерений значений этой функции.

Однако в ряде практических случаев такая обработка измерений оказывается неудобной или трудно реализуемой, главным образом, по следующим причинам.

1. Выше было показано, что точность оценок существенно зависит от количества измерений. Поскольку в обработке по формуле (6 п.11.1) участвуют все измерения одновременно, для получения хорошего качества оценок параметров необходимо располагать достаточно большой памятью с целью хранения результатов измерений. Поэтому ограничение памяти системы обработки результатов измерений по существу фиксирует достижимую точность оценки параметров.

2. Оценки параметров по описанному выше методу могут быть получены только после накопления достаточного количества измерений. В то же время часто определенный интерес представляют текущие оценки параметров, сделанные на базе имеющихся к текущему моменту измерений.

По этим причинам возникает необходимость трансформировать соотношение (6 п.11.1) таким образом, чтобы оно имело рекуррентный характер, т. е. позволяло рассчитать оценки параметров на очередном шаге (после очередного измерения) через оценки на предыдущем шаге и сделанное на очередном шаге измерение.

Введем следующие обозначения:

y_{n+1} — измеренное значение контролируемой переменной в момент времени t_{n+1};

$\tilde{y}_{n+1} = \sum_{i=0}^{k} a_i t^i_{n+1}$ — предсказываемое значение контролируемой переменной в тот же момент времени;

$\mathbf{h}_{n+1} = |1 t_{n+1},\ t^2_{n+1},\ \ldots,\ t^k_{n+1}|$ — вектор пересчета значений параметров в значения контролируемой переменной (этот вектор имеет смысл оператора экстраполяции значения контролируемой переменной на момент времени t_{n+1}). Теперь

$$\tilde{y}_{n+1} = \mathbf{h}_{n+1}\mathbf{A},$$

$$\mathbf{Y}_{n+1}=\begin{vmatrix}\mathbf{Y}_n\\ \cdots\\ y_{n+1}\end{vmatrix};\ \widetilde{\mathbf{Y}}_{n+1}=\begin{vmatrix}\widetilde{\mathbf{Y}}_n\\ \cdots\\ \widetilde{y}_{n+1}\end{vmatrix};\ \mathbf{H}_{n+1}=\begin{vmatrix}\mathbf{H}_n\\ \cdots\\ \mathbf{h}_{n+1}\end{vmatrix}. \quad (8)$$

В соответствии с (6 п.11.1)

$$\widehat{\mathbf{A}}_{n+1}=(\mathbf{H}_{n+1}^T\mathbf{H}_{n+1})^{-1}\mathbf{H}_{n+1}^T\mathbf{Y}_{n+1}. \quad (9)$$

Используя (8), имеем

$$(\mathbf{H}_{n+1}^T\mathbf{H}_{n+1})^{-1}=\left\{(\mathbf{H}_n^T\vdots\mathbf{h}_{n+1}^T)\begin{pmatrix}\mathbf{H}_n\\ \cdots\\ \mathbf{h}_{n+1}\end{pmatrix}\right\}^{-1}=$$
$$=(\mathbf{H}_n^T\mathbf{H}_n+\mathbf{h}_{n+1}^T\mathbf{h}_{n+1})^{-1}.$$

Введем

$$\mathbf{P}_n^{-1}=\mathbf{H}_n^T\mathbf{H}_n.$$

При этом

$$\mathbf{P}_{n+1}^{-1}=\mathbf{H}_{n+1}^T\mathbf{H}_{n+1}=\mathbf{H}_n^T\mathbf{H}_n+\mathbf{h}_{n+1}^T\mathbf{h}_{n+1}=\mathbf{P}_n^{-1}+\mathbf{h}_{n+1}^T\mathbf{h}_{n+1}. \quad (10)$$

Используем лемму об обращении матриц, в соответствии с которой

$$(\mathbf{B}+\mathbf{CDC}^T)^{-1}=\mathbf{B}^{-1}-\mathbf{B}^{-1}\mathbf{C}(\mathbf{C}^T\mathbf{B}^{-1}\mathbf{C}+\mathbf{D}^{-1})^{-1}\mathbf{C}^T\mathbf{B}^{-1},$$
$$(\mathbf{B}+\mathbf{CC}^T)^{-1}=\mathbf{B}^{-1}-\mathbf{B}^{-1}\mathbf{C}(\mathbf{C}^T\mathbf{B}^{-1}\mathbf{C}+1)^{-1}\mathbf{C}^T\mathbf{B}^{-1}.$$

Поэтому

$$\mathbf{P}_{n+1}=(\mathbf{H}_{n+1}^T\mathbf{H}_{n+1})^{-1}=(\mathbf{P}_n^{-1}+\mathbf{h}_{n+1}^T\mathbf{h}_{n+1})^{-1}=$$
$$=\mathbf{P}_n-\mathbf{P}_n\mathbf{h}_{n+1}^T(\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T+1)\mathbf{h}_{n+1}\mathbf{P}_n. \quad (11)$$

Подставляя теперь (11) в (9), получаем

$$\hat{\mathbf{A}}_{n+1} = \mathbf{P}_{n+1}\mathbf{H}_{n+1}^T \mathbf{Y}_{n+1} = \mathbf{P}_{n+1}(\mathbf{H}_n^T \vdots \mathbf{h}_{n+1}^T)\begin{pmatrix}\mathbf{Y}_n \\ \cdots \\ y_{n+1}\end{pmatrix} =$$

$$= \mathbf{P}_{n+1}(\mathbf{H}_n^T \mathbf{Y}_n + \mathbf{h}_{n+1}^T y_{n+1}) = \mathbf{P}_n\mathbf{H}_n^T\mathbf{Y}_n + \mathbf{P}_n\mathbf{h}_{n+1}^T y_{n+1} -$$

$$- \mathbf{P}_n\mathbf{h}_{n+1}^T(\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T + 1)^{-1}\mathbf{h}_{n+1}\mathbf{P}_n(\mathbf{H}_n^T\mathbf{Y}_n + \mathbf{h}_{n+1}^T y_{n+1}) =$$

$$= \mathbf{P}_n\mathbf{H}_n^T\mathbf{Y}_n + \mathbf{P}_n\mathbf{h}_{n+1}^T y_{n+1}(\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T + 1)^{-1}(\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T +$$

$$+ 1) - \mathbf{P}_n\mathbf{h}_{n+1}^T(\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T + 1)^{-1}\mathbf{h}_{n+1}\mathbf{P}_n(\mathbf{H}_n^T\mathbf{Y}_n +$$

$$+ \mathbf{h}_{n+1}^T y_{n+1}) = \hat{\mathbf{A}}_n + \mathbf{P}_n\mathbf{h}_{n+1}^T(\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T + 1)^{-1} \times$$

$$\times (y_{n+1}\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T + y_{n+1} - \mathbf{h}_{n+1}\hat{\mathbf{A}}_n -$$

$$- y_{n+1}\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T) = \hat{\mathbf{A}}_n + \mathbf{P}_n\mathbf{h}_{n+1}^T(\mathbf{h}_{n+1}\mathbf{P}_n\mathbf{h}_{n+1}^T +$$

$$+ 1)^{-1}(y_{n+1} - \mathbf{h}_{n+1}\hat{\mathbf{A}}_n). \qquad (12)$$

Используя (9 п.11.1), перепишем (12) следующим образом:

$$\hat{\mathbf{A}}_{n+1} = \hat{\mathbf{A}}_n + \frac{\Psi_n}{\sigma^2}\mathbf{h}_{n+1}^T\left(\mathbf{h}_{n+1}\frac{\Psi_n}{\sigma^2}\mathbf{h}_{n+1}^T + 1\right)^{-1}(y_{n+1} -$$

$$- \mathbf{h}_{n+1}\hat{\mathbf{A}}_n) = \hat{\mathbf{A}}_n + \Psi_n\mathbf{h}_{n+1}^T(\mathbf{h}_{n+1}\Psi_n\mathbf{h}_{n+1}^T +$$

$$+ \sigma^2)^{-1}(y_{n+1} - \mathbf{h}_{n+1}\hat{\mathbf{A}}_n). \qquad (13)$$

Одновременно, используя (10), имеем

$$\Psi_{n+1}^{-1} = \frac{1}{\sigma^2}\mathbf{P}_{n+1}^{-1} = \frac{1}{\sigma^2}(\mathbf{P}_n^{-1} + \mathbf{h}_{n+1}^T\mathbf{h}_{n+1}) =$$

$$= \left(\Psi_n^{-1} + \mathbf{h}_{n+1}^T\frac{1}{\sigma^2}\mathbf{h}_{n+1}\right).$$

Отсюда в силу леммы об обращении матриц получаем

$$\Psi_{n+1} = \left(\Psi_n^{-1} + \mathbf{h}_{n+1}^T\frac{1}{\sigma^2}\mathbf{h}_{n+1}\right)^{-1} = \Psi_n -$$

$$- \Psi_n\mathbf{h}_{n+1}^T(\mathbf{h}_{n+1}\Psi_n\mathbf{h}_{n+1}^T + \sigma^2)^{-1}\mathbf{h}_{n+1}\Psi_n. \qquad (14)$$

Соотношения (13) и (14) представляют собой рекуррентные расчетные формулы для определения вектора оценок параметров системы $\mathbf{A}_{n+1}$ и корреляционной матрицы ошибок оценок параметров Ψ_{n+1} после очередного (n+1)-го измерения. Анализ этих соотношений показывает, что для их использования необходима лишь информация о предыдущих оценках ($\mathbf{A}_n$ и Ψ_n) и очередное измерение — y_{n+1}.

Кроме того, заметим, что эволюция элементов корреляционной матрицы ошибок оценок параметров (14) не зависит от измерений и, таким образом, элементы этой матрицы могут быть рассчитаны заранее для любого числа измерений.

11.10.3. Оценка параметров по критерию максимума правдоподобия

Описанную выше методику оценки параметров по методу наименьших квадратов целесообразно использовать при решении задач наилучшей аппроксимации неизвестной функции отклика в классе соотношений определенной структуры (например, в классе ортогональных полиномов заданной степени). Этот же метод может применяться и в случае, когда структура функции отклика известна априорно, но на измерения накладывается стационарный случайный процесс, так что все измерения в информационном отношении оказываются равноценными.

Если же ошибки измерения представляют собой нестационарный случайный процесс, причем статистические характеристики его известны (например, закон изменения дисперсии), то естественно обрабатывать измерения с учетом их веса, зависящего от дисперсии ошибки измерения. Такой учет в наиболее ясной форме проводится при оценке параметров по методу максимума правдоподобия.

Рассмотрим задачу оценки параметров системы для случая, когда на измерения накладывается нестационарная помеха с нормальным законом распределения.

В отличие от предыдущего будем читать, что наблюдаемый процесс является многомерным. Поэтому каждое измерение представляет собой вектор, компонентами которого являются координаты процесса. Пусть, как и ранее, в результате обработки n измерений $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n$, выполненных в моменты времени $t_1 < t_2 < \ldots < t_n$, получен вектор оценок параметров $\hat{\mathbf{a}}_n$. В момент времени t_{n+1} выполняется еще одно измерение $\mathbf{Y}_{n+1}$, причем

$$\mathbf{Y}_{n+1} = \mathbf{H}\mathbf{a} + \mathbf{V}_{n+1}, \qquad (15)$$

где $\mathbf{V}_{n+1}$ — вектор накладывающейся на измерение помехи; $\mathbf{H}$ — нелинейный оператор пересчета параметров в координаты; $\mathbf{a}$ — вектор истинных значений параметров.

Введем следующие обозначения:

$\overline{[\mathbf{V}_{n+1}\mathbf{V}_{n+1}^T]} = \mathbf{R}_{n+1}$ — корреляционная матрица ошибок измерения, проводимого в момент t_{n+1};

$\overline{[(\mathbf{a}-\hat{\mathbf{a}}_n)(\mathbf{a}-\hat{\mathbf{a}}_n)^T]} = \Psi_n$ — корреляционная матрица ошибок оценок параметров по n измерениям.

Требуется получить рекуррентные соотношения для расчета оценок параметров с учетом нового измерения и соответствующей корреляционной матрицы ошибок оценок. Выпишем совместную плотность вероятностей неизвестных случайных векторов оценки параметров $\mathbf{a}$ и помехи $\mathbf{V}_{n+1}$:

$$P(\mathbf{a}, \mathbf{V}_{n+1}) = \frac{1}{(2\pi)^{\frac{k+1}{2}} |\Psi|^{1/2}} \times$$

$$\times \exp\left[-\frac{1}{2}(\mathbf{a}-\hat{\mathbf{a}}_n)^T\Psi_n^{-1}(\mathbf{a}-\hat{\mathbf{a}}_n)\right] \frac{1}{(2\pi)^{r/2}|\mathbf{R}_{n+1}|^{1/2}} \times$$

$$\times \exp\left[-\frac{1}{2}(\mathbf{Y}_{n+1}-\mathbf{H}\mathbf{a})^T\mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1}-\mathbf{H}\mathbf{a})\right]. \quad (\ )$$

Здесь $k+1$ — размерность вектора параметров; r — размерность вектора измерений.

Вектор оценок параметров после $(n+1)$-го измерения выберем таким образом, чтобы (отсюда название метода) максимизировать (16). Вполне ясно, что

$$\max_{\hat{\mathbf{a}}}\{P(\mathbf{a}, \mathbf{V}_{n+1})\}; \quad \max_{\hat{\mathbf{a}}}\{\ln P(\mathbf{a}, \mathbf{V}_{n+1})\}$$

и

$$\min_{\hat{\mathbf{a}}}\left\{\frac{1}{2}[(\mathbf{a}-\hat{\mathbf{a}}_n)^T\Psi_n^{-1}(\mathbf{a}-\hat{\mathbf{a}}_n) + (\mathbf{Y}_{n+1}-\mathbf{H}\mathbf{a})^T \times \right.$$

$$\left. \times \mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1}-\mathbf{H}\mathbf{a})]\right\}$$

достигаются одновременно.

Итак, задача сведена к стысканию вектора $\hat{\mathbf{a}}_{n+1}$, минимизирующего функционал

$$\mathcal{J} = \frac{1}{2}[(\mathbf{a}-\hat{\mathbf{a}}_n)^T\Psi_n^{-1}(\mathbf{a}-\hat{\mathbf{a}}_n) + (\mathbf{Y}_{n+1}-\mathbf{H}\mathbf{a})^T \times$$

$$\times \mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1}-\mathbf{H}\mathbf{a})],$$

откуда

$$\left.\frac{\partial \mathcal{J}}{\partial \hat{\mathbf{a}}^T}\right|_{\mathbf{a}=\hat{\mathbf{a}}_{n+1}} = \Psi_n^{-1}(\hat{\mathbf{a}}_{n+1} - \hat{\mathbf{a}}_n) - \mathbf{H}^T\mathbf{R}_{n+1}^{n}(\mathbf{Y}_{n+1} - \mathbf{H}\hat{\mathbf{a}}_{n+1}) = 0. \tag{17}$$

Разрешая (17) относительно $\hat{\mathbf{a}}_{n+1}$, имеем

$$(\Psi_n^{-1} + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{H})\hat{\mathbf{a}}_{n+1} = \Psi_n^{-1}\hat{\mathbf{a}}_n + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{Y}_{n+1},$$

$$\hat{\mathbf{a}}_{n+1} = (\Psi_n^{-1} + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{H})^{-1}[\Psi_n^{-1}\hat{\mathbf{a}}_n + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{Y}_{n+1}] =$$

$$= (\Psi_n^{-1} + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{H})^{-1}[\Psi_n^{-1}\hat{\mathbf{a}}_n + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1} + \mathbf{H}\hat{\mathbf{a}}_n -$$

$$- \mathbf{H}\hat{\mathbf{a}}_n)] = (\Psi_n^{-1} + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{H})^{-1}(\Psi_n^{-1} + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{H})\hat{\mathbf{a}}_n +$$

$$+ (\Psi_n^{-1} + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1} - \mathbf{H}\hat{\mathbf{a}}_n) =$$

$$= \mathbf{a}_n + \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1} - \mathbf{H}\hat{\mathbf{a}}_n), \tag{18}$$

где

$$\Psi_{n+1} = (\Psi_n^{-1} + \mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{H})^{-1}, \tag{19}$$

или, используя лемму об обращении матриц,

$$\Psi_{n+1} = \Psi_n - \Psi_n\mathbf{H}^T(\mathbf{H}\Psi_n\mathbf{H}^T + \mathbf{R}_{n+1})^{-1}\mathbf{H}\Psi_n.$$

Покажем, что Ψ_{n+1} является корреляционной матрицей ошибок оценок параметров после $(n+1)$-го измерения. Так как

$$\hat{\mathbf{a}}_{n+1} - \mathbf{a} = \hat{\mathbf{a}}_n - \mathbf{a} + \hat{\mathbf{a}}_{n+1} - \hat{\mathbf{a}}_n,$$

то, используя (15) и (18), имеем

$$\hat{\mathbf{a}}_{n+1} - \mathbf{a} = \hat{\mathbf{a}}_n - \mathbf{a} + \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}[\mathbf{V}_{n+1} - \mathbf{H}(\hat{\mathbf{a}}_n - \mathbf{a})] =$$

$$= \hat{\mathbf{a}}_n - \mathbf{a} - \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{H}(\hat{\mathbf{a}}_n - \mathbf{a}) + \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{V}_{n+1} =$$

$$= (\mathbf{I} - \mathbf{K}\mathbf{H})(\hat{\mathbf{a}}_n - \mathbf{a}) + \mathbf{K}\mathbf{V}_{n+1},$$

где $\mathbf{K} = \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}$. Поскольку $(\hat{\mathbf{a}}_n - \mathbf{a})$ и $\mathbf{V}_{n+1}$ — независимые случайные векторы, то

$$M[(\hat{\mathbf{a}}_{n+1} - \mathbf{a})(\hat{\mathbf{a}}_{n+1} - \mathbf{a})^T] = (\mathbf{I} - \mathbf{K}\mathbf{H})M[(\hat{\mathbf{a}}_n - \mathbf{a}) \times$$

$$\times (\hat{\mathbf{a}}_n - \mathbf{a})^T](\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}M[\mathbf{V}_{n+1}\mathbf{V}_{n+1}^T]\mathbf{K}^T =$$

$$= (\mathbf{I} - \mathbf{K}\mathbf{H})\Psi_n(\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{R}_{n+1}\mathbf{K}^T. \tag{20}$$

Из (19) следует, что

$$\Psi_{n+1}^{-1} = \Psi_n^{-1} + \mathbf{H}^T \mathbf{R}_{n+1}^{-1} \mathbf{H}. \tag{21}$$

Умножая (21) слева на Ψ_{n+1} и справа на Ψ_n, имеем

$$\Psi_n = \Psi_{n+1} (\Psi_n^{-1} + \mathbf{H}^T \mathbf{R}_{n+1}^{-1} \mathbf{H}) \Psi_n = \Psi_{n+1} + \mathbf{K}\mathbf{H}\Psi_n.$$

Отсюда

$$\Psi_{n+1} = (\mathbf{I} - \mathbf{K}\mathbf{H}) \Psi_n. \tag{22}$$

Подставляя (22) в (20), получаем

$$M[(\hat{\mathbf{a}}_{n+1} - \mathbf{a})(\hat{\mathbf{a}}_{n+1} - \mathbf{a})^T] = \Psi_{n+1} (\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{R}_{n+1}\mathbf{K}^T =$$
$$= \Psi_{n+1} - \Psi_{n+1}\mathbf{H}^T\mathbf{K}^T + \mathbf{K}\mathbf{R}_{n+1}\mathbf{K}^T =$$
$$= \Psi_{n+1} - \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}\mathbf{R}_{n+1}\mathbf{K}^T + \mathbf{K}\mathbf{R}_{n+1}\mathbf{K}^T =$$
$$= \Psi_{n+1} - \mathbf{K}\mathbf{R}_{n+1}\mathbf{K}^T + \mathbf{K}\mathbf{R}_{n+1}\mathbf{K}^T = \Psi_{n+1},$$

что и требовалось.

Таким образом, рекуррентные соотношения для расчета вектора оценок параметров после очередного $(n+1)$-го измерения и соответствующей корреляционной матрицы ошибок оценок параметров имеют вид

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n + \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1} - \mathbf{H}\hat{\mathbf{a}}_n), \tag{23}$$

$$\Psi_{n+1} = \Psi_n - \Psi_n\mathbf{H}^T(\mathbf{H}\Psi_n\mathbf{H}^T + \mathbf{R}_{n+1})^{-1}\mathbf{H}\Psi_n. \tag{24}$$

11.10.4. Оценка параметров динамических распознаваемых объектов

Рассмотренная выше методика пригодна для оценки параметров так называемых статических систем, параметры которых не изменяются во времени. Вместе с тем, совершенно аналогичным образом можно получить необходимые расчетные соотношения для оценки параметров динамических распознаваемых объектов, закон изменения параметров которых известен.

Пусть например, динамика распознаваемого объекта описывается разностными уравнениями вида

$$\vartheta_i = \Phi\vartheta_{i-1}, \quad i = 0, 1, 2, \ldots \tag{25}$$

Здесь ϑ_i — вектор параметров распознаваемого объекта в момент времени t_i. Так же, как и ранее, вектор измерений в момент времени t_i $(i = 0, 1, 2, \ldots)$ определяется соотношением

$$\mathbf{Y}_i = \mathbf{H}\vartheta_i + \mathbf{V}_i,$$

где $\mathbf{V}_i$—вектор случайного шума, статистические характеристики которого предполагаются известными, причем

$$M[V_i] = 0; \quad M[\mathbf{V}_i \mathbf{V}_j^T] = r_{ij}\delta_{ij},$$

где δ_{ij}—символ Кронекера

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Таким образом, предполагается, что ошибки различных измерений не коррелированы между собой.

Для оценки параметров распознаваемого объекта в рассматриваемом случае могут быть использованы ранее полученные соотношения (24) и (23).

Введем следующие обозначения:

$\hat{\vartheta}_n$ — вектор оценок параметров после n измерений на момент времени t_n;

$\hat{\vartheta}_{n+1,\ni}$ — вектор оценок параметров после n измерений, экстраполированный в соответствии с (25) на момент проведения $(n+1)$-го измерения;

$\hat{\vartheta}_{n+1}$ — вектор оценок параметров после $(n+1)$-го измерения;

Ψ_n — корреляционная матрица ошибок оценок параметров после n измерений;

$\Psi_{n+1,\ni}$ — корреляционная матрица ошибок оценок параметров после n измерений, экстраполированная на момент проведения $(n+1)$-го измерения; Ψ_{n+1} — корреляционная матрица ошибок оценок параметров после $(n+1)$-го измерения. В соответствии с (25)

$$\hat{\vartheta}_{n+1,\ni} = \Phi\hat{\vartheta}_n,$$

$$\Psi_{n+1,\ni} = \overline{\hat{\vartheta}_{n+1,\ni}\hat{\vartheta}_{n+1,\ni}^T} = \Phi\overline{\hat{\vartheta}_n\hat{\vartheta}_n^T}\,\Phi^T = \Phi\Psi_n\Phi^T.$$

Подставляя теперь в (23) и (24) $\hat{\vartheta}_{n+1}$ и $\hat{\vartheta}_{n+1,\ni}$ вместо $\hat{\mathbf{a}}_{n+1}$ и $\hat{\mathbf{a}}_n$, а также $\Psi_{n+1,\ni}$ вместо Ψ_n, имеем

$$\hat{\vartheta}_{n+1} = \hat{\vartheta}_{n+1,\ni} + \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1} - \mathbf{H}\hat{\vartheta}_{n+1,\ni}),$$

$$\Psi_{n+1} = \Psi_{n+1,\ni} - \Psi_{n+1,\ni}\mathbf{H}^T(\mathbf{H}\Psi_{n+1,\ni}\mathbf{H}^T + \mathbf{R}_{n+1})^{-1}\mathbf{H}\Psi_{n+1,\ni}.$$

Введем, наконец, $\mathbf{Y}_{n+1,\ni}$—экстраполированное значение вектора измерений на момент времени t_{n+1} предсказываемое через соответствующий вектор оценок параметров по формуле

$$\mathbf{Y}_{n+1,\ni} = \mathbf{H}\hat{\vartheta}_{n+1,\ni}.$$

Теперь, окончательно, имеем систему соотношений для рекуррентной оценки параметров динамического распознаваемого объекта

$$\hat{\Theta}_{n+1,\,s} = \Phi\,\hat{\Theta}_n, \quad \Psi_{n+1,\,s} = \Phi\Psi_n\Phi^T,$$

$$\Psi_{n+1} = \Psi_{n+1,\,s} - \Psi_{n+1,\,s}\mathbf{H}^T(\mathbf{H}\Psi_{n+1,\,s}\mathbf{H}^T + \mathbf{R}_{n+1})^{-1}\mathbf{H}\Psi_{n+1,s},$$

$$\hat{\Theta}_{n+1} = \hat{\Theta}_{n+1,\,s} + \Psi_{n+1}\mathbf{H}^T\mathbf{R}_{n+1}^{-1}(\mathbf{Y}_{n+1} - \mathbf{Y}_{n+1,\,s}). \qquad (26)$$

Система соотношений (26) в литературе называется фильтром Калмана для оценки параметров линейных систем.

Отметим, что в соответствии с этими соотношениями новая оценка параметров есть результат линейной комбинации старой оценки и разности между вектором измерений на очередном шаге и предсказываемым к этому моменту его значением, взятой с весом, учитывающим соотношение между величинами неопределенности в достигнутой оценке параметров и неопределенности очередного измерения.

Литература

1. *Айзерман М. А ., Браверман Э. М., Розопоэр Л. И.* Метод потенциальных функций в теории обучения машин. М.: Наука, 1970. 240 с.
2. *Бонгард М. М.* Проблема узнавания. М.: Наука, 1967. 320 с.
3 *Вайнцвайг М. И.* Алгоритм обучения распознаванию образов «Кора» //Алгоритмы обучения распознаванию образов: Сб. ст., Под ред. В. Н. Банника. М.: Сов. радио, 1973. С. 110—116.
4. *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным. М.: Наука, 1979. 448 с.
5. *Вапник В.Н., Червоненкис А. Я.* Теория распознавания образов. М.: Наука, 1974. 415 с.
6. *Горелик А. Л., Гуревич И. Б., Скрипкин В. А.* Современное состояние проблемы распознавания. Некоторые аспекты. М.: Радио и связь, 1985. 162 с. (Кибернетика).
7. *Гренандер У.* Лекции по теории образов: В 3 т. / Пер. с англ. под ред. Ю.И.Журавлева. М.: Мир. 1979- 1983. 130с.
8. *Гуревич И. Б.* Анализ изображений методом реверсивного алгебраического замыкания // Проблемы искусственного интеллекта и распознавания образов: Тез. докл. и сообщ. Науч. конф. с участием ученых из соц. стран (Киев, 11—18 мая 1984 г.). Секция П.: Распознавание образов. Киев: Ин-т кибернетики им В. М. Глушкова АН УССР, 1984. С. 41—43.
9. *Гуревич И. Б.* Алгебраический подход к анализу и распознаванию изображений // Математические методы распознавания образов: Тез. докл. Всесоюз. конф. (Дилпжан, 16 — 21 мая 1985 г.). Ереван: Изд-во АИ Арм. ССР, 1985. С. 55—57.
10. *Гуревич И. Б.* Определение класса алгоритмов вычисления оценок по двухмерной информации для задач распознавания изображении // Методы и средства обработки графической информации: Межвуз. сб. науч. тр. / Под ред. Ю. Г. Васина. Горький: Горьк. гос. ун-т им. Н.И.Лобачевского, 19SG. С. 47—60.
11. *Гуревич И. Б.* Проблема распознавания изображений //Распознавание, классификация, прогноз. Математические методы и их применение: Ежегодник / Под ред. Ю. И. Журавлева. М.: Наука 1988. Вып. 1. С. 280—329.
12. *Гуревич И. В., Журавлев Ю. II.* Минимизация булевых функций и эффективные алгоритмы распознавания // Кибернетика. 1974. № 3. С. 16—20.

Научно-учебно-методическое издание

Кононюк Анатолий Ефимович

Общая теория распознавания

Математические средства описания распознаваемых объектов и распознающих процессов

Книга 2

Авторская редакция

Подписано в печать 25.02.2012 г.
Формат 60x84/16.
Усл. печ. л. 18,5. Тираж 300 экз.

Издатель и изготовитель:
Издательство «Освіта України»
04214, г. Киев, ул. Героев Днепра, 63, к. 40
Свидетельство о внесении в Государственный реестр
издателей ДК №1957 от 23.04.2009 г.
Тел./факс (044) 411-4397; 237-5992
E-mail: osvita2005@ukr.net, www.rambook.ru

$\dot{-}$