

Парадигма развития науки
Методологическое обеспечение

А. Е. Кононюк

ДИСКРЕТНО-НЕПРЕРЫВНАЯ
МАТЕМАТИКА

Книга 7

Графы

Часть 5

Киев
«Освіта України»
2015

Конюк А.Е. Графы



Конюк Анатолий Ефимович

Кононюк А.Е. Графы

Структурная схема развивающейся панмедийной системы наук



Кононюк А.Е. Графы

УДК 51 (075.8)

ББК В161.я7

К213

Рецензенты:

В. В. Довгай — к-т физ.-мат. наук, доц. (Национальный технический университет «КПИ»);

В. В. Гавриленко — д-р физ.-мат. наук, проф.,

О. П. Будя — к-т техн. наук, доц. (Киевский университет экономики, туризма и права);

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

Кононюк А. Е.

К213 Дискретно-непрерывная математика. (Графы. К.7,Ч.5 (в 7 частях)). — 15-и кн. Кн 7.— К.: Освіта України. 2015. 704 с.

ISBN 978-966-373-693-8 (многотомное издание)

ISBN 978-966-373-694-5 (книга 7)

Многотомная работа содержит систематическое изложение математических дисциплин, используемых при моделировании и исследованиях математических моделей систем.

В работе излагаются основы теории множеств, отношений, поверхностей, пространств, алгебраических систем, матриц, графов, математической логики, теории вероятностей и массового обслуживания, теории формальных грамматик и автоматов, теории алгоритмов, которые в совокупности образуют единую методологически взаимосвязанную математическую систему «Дискретно-непрерывная математика».

Для бакалавров, специалистов, магистров, аспирантов, докторантов и просто ученых и специалистов всех специальностей.

УДК 51 (075.8)

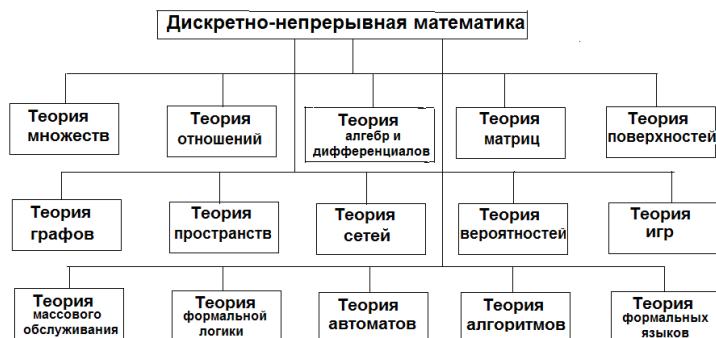
ББК В161.я7

ISBN 978-966-373-693-8 (многотомное издание) © Кононюк А. Е.,

ISBN 978-966-373-694-5 (книга 7) © Освіта України, 2015

Кононюк А.Е. Графы

Структура
открытой развивающейся панмедийной системы математических наук (дисциплин)
"Дискретно-непрерывная математика"



Оглавление

1. Графы связей.....	10
1.1. Основные понятия и определения.....	11
1.2. Метод использования графов связей при моделировании динамических систем.....	22
1.3. Методика построения многосвязных моделей в механике.....	37
1.4. Метод моделирования гидродинамических систем графами связей.....	60
1.5. Модели с распределенными параметрами сердечно-сосудистой системы человека.....	86
1.6. Метод графов связей для описания нелинейных процессов диффузии и химической реакции в однородных или неоднородных средах.....	111
1.7. Мемристор - новый элемент графов связей.....	141
1.8. Применение правила циклов к графам связей.....	152
1.9. Лагранжевы графы связей.....	181
1.10. Исследование линейных и нелинейных характеристик подвесок аппаратов на воздушной подушке с использованием моделей, основанных на графах связей.....	210
1.11. ПРИМЕНЕНИЕ ГРАФОВ СВЯЗЕЙ ПРИ РАЗРАБОТКЕ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ ПОРШНЕВЫХ ДВС.....	237
1.12. Графы связей и размещение узлов в «сетях – на - кристалле».....	245
2. Сети Петри.....	262
2.1. Основные понятия.....	262
2.1.1. Системы, процессы, сети.....	263

Кононюк А.Е. Графы

2.1.2. Формальное определение сети Петри.....	270
2.1.3. Структура сети Петри.....	277
2.1.4. Графы сетей Петри.....	280
2.1.5. Маркировка сетей Петри.....	284
2.1.6. Правила выполнения сетей Петри.....	286
2.1.7. Пространство состояний сети Петри.....	291
2.1.8. Альтернативные формы определения сетей Петри.....	293
2.2. СЕТИ ПЕТРИ ДЛЯ МОДЕЛИРОВАНИЯ.....	295
2.2.1. События и условия.....	295
2.2.2. Одновременность и конфликт.....	300
2.2.3. Аппаратное обеспечение ЭВМ.....	307
2.2.3.1. Конечные автоматы.....	307
2.2.3.2. ЭВМ с конвейерной обработкой.....	316
2.2.3.3. Кратные функциональные блоки.....	320
2.2.4. Программное обеспечение ЭВМ.....	323
2.2.4.1. Блок-схемы.....	323
2.2.4.2. Параллелизм.....	329
2.2.4.3. Синхронизация.....	330
2.2.4.4. Задача о взаимном исключении.....	331
2.2.4.5. Задача о производителе/потребителе.....	333
2.2.4.6. Задача об обедающих мудрецах.....	335
2.2.4.7. Задача о чтении/записи.....	337
2.2.4.8. P- и V-системы.....	337
2.2.5. Другие системы.....	339
2.3. Свойства сетей Петри.....	347
2.3.1. Основные свойства сетей Петри.....	347
2.3.2. Проблемы ограниченности и безопасности.....	350
2.3.3. Проблемы R-включения и R-эквивалентности.....	357
2.3.4. Проблемы достижимости и живости.....	364
2.4 Анализ сетей Петри.....	368
2.4.1. Задачи анализа сетей Петри.....	369
2.4.1.1. Безопасность.....	369
2.4.1.2. Ограниченность.....	370
2.4.1.3. Сохранение.....	371
2.4.1.4. Активность.....	374
2.4.1.5. Достижимость и покрываемость.....	377
2.4.1.6. Последовательности запусков.....	378
2.4.1.7. Задачи эквивалентности и подмножества.....	380
2.4.2. Методы анализа.....	380
2.4.2.1. Дерево достижимости.....	381
2.4.2.2. Безопасность и ограниченность.....	391

Кононюк А.Е. Графы

2.4.2.3. Сохранение.....	393
2.4.2.4. Покрываемость.....	394
2.4.2.5. Ограниченность дерева достижимости.....	396
2.4.2.6. Матричные уравнения.....	400
2.5. СЛОЖНОСТЬ И РАЗРЕШИМОСТЬ.....	405
2.5.1. Сводимость задач анализа.....	405
2.5.2. Задачи достижимости.....	410
2.5.3. Сети Петри с ограничениями.....	415
2.5.4. Активность и достижимость.....	419
2.5.5. Неразрешимые задачи.....	424
2.5.5.1. Задача включения графов полиномов.....	425
2.5.5.2. Слабое вычисление.....	427
2.5.5.3. Задача равенства.....	434
2.5.6. Сложность задачи достижимости.....	438
2.6. Языки сетей Петри.....	442
2.6.1. Предпосылки изучения языков сетей Петри.....	443
2.6.2. Некоторые понятия теории формальных языков.....	445
2.6.3. Определения языков сетей Петри.....	446
2.6.3.1. Начальное состояние.....	446
2.6.3.2. Помечение сетей Петри.....	447
2.6.3.3. Заключительные состояния сети Петри.....	447
2.6.3.4. Классы языков сетей Петри.....	449
2.6.4. Свойства языков сетей Петри.....	454
2.6.5. Свойства замкнутости.....	459
2.6.5.1. Конкатенация.....	459
2.6.5.2. Объединение.....	461
2.6.5.3. Параллельная композиция.....	464
2.6.5.4. Пересечение.....	466
2.6.5.5. Обращение.....	467
2.6.5.6. Дополнение.....	468
2.6.5.7. Повторная композиция.....	468
2.6.5.8. Подстановка.....	471
2.6.6. Языки сетей Петри и другие классы языков.....	472
2.6.6.1. Регулярные языки.....	472
2.6.6.2. Контекстно-свободные языки.....	473
2.6.6.3. Ограниченные контекстно-свободные языки.....	475
2.6.6.4. Контекстно-связанные языки.....	478
2.6.7. Помеченные сети и классы языков сетей Петри.....	482
2.6.8. Характеризация классов языков сетей Петри.....	492
2.6.9. Разрешимые и неразрешимые свойства языков сетей Петри.....	499
2.7. Подклассы сетей Петри.....	508
2.7.1. Ординарные сети Петри.....	508

Кононюк А.Е. Графы

2.7.2. Автоматные сети и синхронизационные графы.....	513
2.7.3. Свободные сети.....	519
2.8. Расширенные и ограниченные модели сетей Петри.....	526
2.8.1. Границы возможностей моделирования с помощью сетей Петри.....	527
2.8.2. Расширения.....	534
2.8.2.1. Области ограничения.....	536
2.8.2.2. Переходы исключающее ИЛИ и переключатели.....	539
2.8.2.3. Другие расширения.....	541
2.8.3. Расширенные сети Петри и регистровые машины.....	542
2.8.4. Подклассы сетей Петри.....	545
2.8.4.1. Автоматные сети Петри.....	546
2.8.4.2. Маркированные графы.....	547
2.8.4.3. Сети Петри со свободным выбором.....	549
2.8.4.4. Правильные сети Петри.....	551
2.9. Обобщения сетей Петри.....	551
2.9.1. Счетчиковые автоматы.....	551
2.9.2. Ингибиторные сети и сети с приоритетами.....	556
2.9.3. Раскрашенные, синхронные и самомодифицируемые сети.....	561
2.10. Регулярные и иерархические сети.....	567
2.10.1. Алгебра регулярных сетей.....	568
2.10.2. Некоторые свойства регулярных сетей.....	574
2.10.3. Иерархические сети.....	583
2.10.4. Сравнение иерархических сетей с другими классами сетей.....	587
2.10.5. Сети с ожиданием и строго иерархические сети.....	593
2.11. Сети-системы и сети-процессы.....	600
2.11.1. Процессы.....	601
2.11.2. Сетевое представление параллельных процессов.....	605
2.11.3. Сетевое представление последовательно-альтернативных процессов.....	612
2.11.4. Сетевое представление параллельно-альтернативных процессов.....	614
2.11.5. Развертка сетей Петри в сети-процессы.....	622
2.11.6. Сетевое представление параллельных процессов с конкуренцией.....	634
2.12. Сети Петри и программирование.....	638
2.12.1. Сети Петри и семантика структур управления.....	638
2.12.2. Поточковые сети.....	649
2.12.3. Типы управления.....	654
2.12.4. Петри-машина.....	665
2.13. Модели параллельных вычислений.....	670

Кононюк А.Е. Графы

2.13.1. Конечные автоматы.....	673
2.13.2. Маркированные графы.....	673
2.13.3. Графы вычислений.....	674
2.13.4. P/V-системы.....	677
2.13.5. Системы передачи сообщений.....	681
2.13.6. Графы UCLA.....	689
2.13.7. Системы замещения и сложения векторов.....	695
2.13.8. Расширенные модели сетей Петри.....	698
Приложение.....	700
Литература.....	704

1. Графы связей

Вступление

Моделирование и проектирование систем невозможно представить без применения вычислительной техники. При создании автоматизированных систем и их проектировании прежде всего возникает проблема выбора формальной модели представления систем. От модельного через алгоритмическое к программному обеспечению — таков путь современного моделирования и проектирования систем.

При рассмотрении физических систем (электрических, механических, гидравлических и др.) с сосредоточенными параметрами удобной моделью является линейный граф, каждая вершина которого соответствует функциональной или конструктивной компоненте, дуга — причинно-следственной связи. Граф называется линейным, если он представлен в виде линейной комбинации циклов. Графовая модель позволяет установить наглядную связь между структурой (топологией) системы и ее количественными (энергетическими, мощностными и др.) характеристиками. Линейный граф, определяющий физическую систему, в сущности представляет собой функцию, заданную на графе, областью определения которой являются его элементы, областью значений — количественные характеристики системы.

Использование линейного графа при расчете физических систем с сосредоточенными параметрами обязано принципу суперпозиции, реализованному и развитому в работах Кирхгофа (1847 г.), Максвелла (1892 г.). Крона (1957 г.). Этот принцип позволил эффективно декомпозировать (разбивать) систему и тем самым значительно уменьшать трудоемкость проводимых расчетов.

Среди основных работ, использующих линейный граф как модель физической системы с сосредоточенными параметрами, следует отметить более десятка монографий написанных такими известными учеными, как Мэсон, Сешу, Баувер, Рид и др.

Наряду с применением линейных графов при моделировании физических систем с сосредоточенными параметрами, наверное, начиная с работ Б. Н. Петрова (1948 г.) и Мэсона (1953 г.) линейные графы получают широкое применение для анализа сложных динамических систем автоматического управления. При этом особенно следует отметить направление структурных методов: В. В.

Кононюк А.Е. Графы

Солодовников (1956 г.), Г. С. Поспелов (1958 г.), А. С. Шаталов (1962 г.), Л. Т. Кузин (1962 г.) и др.

Содержание материала по теории графов связи, которое базируется на работах Кэрнопа Д., Розенберга Р., Брауна Е., Ауслендер Д., Ричардсона и др., ведущих специалистов в области моделирования и проектирования сложных динамических систем, является дальнейшим развитием применения линейных графов для расчета таких систем. Авторы заменили общепринятый термин «линейный граф» на «граф связей».

В работе развиваются основные идеи графов связей и их разнообразные применения при моделировании и расчете сложных гидравлических, механических, биологических, биофизических и других систем. Основное назначение раздела — дать специалисту инженерного профиля конкретной области новый метод решения практических задач.

1.1. Основные понятия и определения

Цель данного раздела привести в сжатом и обобщенном виде основные определения языка графов связей, предложенный Розенбергом и Кэрнопом. Язык представляет собой формальную математическую систему определений и символов. Наименования дескрипторов определены в терминах, связанных с энергией и мощностью, так как именно в энергетике впервые было применено понятие многосвязности. Графы связей являются мощным средством, используемым в качестве инструмента моделирования многосвязных систем.

Основные определения . Многосвязные элементы , полусвязи и связи. *Многосвязные элементы* представляют собой вершины графа и обозначаются буквенно-цифровыми символами (далее для удобства будем называть их элементами). На рис. 1,*a* приведены два многосвязных элемента, 1 и R.

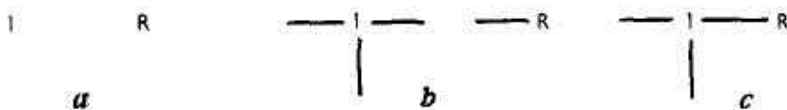


Рис. 1. Многосвязные элементы, полусвязи и связи.

a — два многосвязных элемента; *b* — элементы и их полусвязи; *c* — образование связи.

Кононюк А.Е. Графы

Полусвязи многосвязного элемента обозначаются отрезками линий инцидентными элементу с одной стороны. Полусвязи располагаются там, где элемент может взаимодействовать со средой.

Например, на рис. 1, b элемент 1 имеет три полусвязи, а элемент R — одну. Тогда говорят, что элемент 1 является 3-связным, а элемент R — 1-связным.

Связи образуются при объединении пар полусвязей. Иными словами, связи — это объединения пар многосвязных элементов. Так, на рис. 1, c объединение двух полусвязей образует связь между 1 и R .

Графы связей. *Граф связей* представляет собой объединение многосвязных элементов, связанных вместе. В общем случае это линейный граф, вершинами которого являются многосвязные элементы, а дугами — связи. Граф связей может состоять из одной или нескольких частей, может иметь несколько петель или не иметь их, в общем случае имеет характеристики произвольного линейного графа. Пример графа связей приведен на рис. 2.

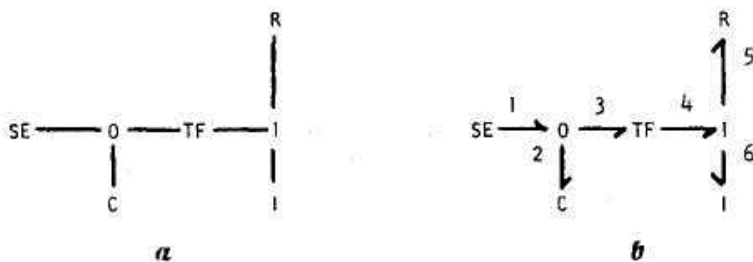


Рис. 2. Пример графа связей.

a — граф связей; b — граф связей с ориентированными мощностями и нумерованными связями.

На рис. 2, a показан граф с семью элементами и шестью связями. На рис. 2, b показан тот же граф, но с ориентированными мощностями и нумерованными связями.

Фрагмент графа связей — это граф связей (более строго здесь следует говорить о *части* графа связей), не все полусвязи которого соединены как связи. На рис. 1, c приведен пример фрагмента графа связей, имеющего одну связь и две открытые (несоединенные) полусвязи.

Переменные полусвязей. Для произвольной полусвязи определены шесть величин, три из которых являются интегральными.

Кононюк А.Е. Графы

Усилие $e(t)$ и поток $f(t)$ непосредственно относятся к данной полусвязи и называются переменными мощности полусвязи. Они определяются как скалярные функции независимой переменной (t) .

Мощность $P(t)$ определяется как скалярное произведение усилия на поток

$$P(t) = e(t) \cdot f(t).$$

Положительное направление мощности отмечается стрелкой на связи.

Момент $p(t)$ и перемещение $q(t)$ связаны с усилием и потоком полусвязи интегральными соотношениями

$$p(t) = p(t_0) + \int_{t_0}^t e(\lambda) d\lambda$$

и

$$q(t) = q(t_0) + \int_{t_0}^t f(\lambda) d\lambda,$$

соответственно.

Момент и перемещение называют иногда энергетическими переменными.

Энергия $E(t)$ выражается через мощность полусвязи с помощью следующего равенства:

$$E(t) = E(t_0) + \int_{t_0}^t P(\lambda) d\lambda.$$

Величина $E(t) - E(t_0)$ представляет собой полезную энергию, передаваемую через полусвязь в направлении, указанном полустрелкой (т. е. положительная мощность), за интервал времени (t_0, t) .

Обычно для графов связей изображают только усилие и поток следом за полусвязью (или связью). Мощность, перемещение, момент и энергия определяются с помощью вышеприведенных формул.

Основные многосвязные элементы. Имеется девять основных многосвязных элементов, которые делятся на четыре категории в соответствии с их энергетическими характеристиками. Эти элементы и их определения приведены на рис.3.

Кононюк А.Е. Графы

Символ	Определение	Наименование
SE \xrightarrow{e}	$e = e(t)$	Источник усилия
SF \xrightarrow{f}	$f = f(t)$	Источник потока
C $\xleftarrow[e]{f}$	$e = \Phi(q)$ $q(t) = q(t_0) + \int f \cdot dt$	Емкость
I $\xleftarrow[e]{f}$	$f = \Phi(p)$ $p(t) = p(t_0) + \int e \cdot dt$	Инерционность
R $\xleftarrow[e]{f}$	$\Phi(e, f) = 0$	Спротивление
$\xrightarrow{1} \text{TF} \xrightarrow{2}$ 1:m	$e_1 = m \cdot e_2$ $m \cdot f_1 = f_2$	Трансформатор
$\xrightarrow{1} \text{GY} \xrightarrow{2}$ r	$e_1 = r \cdot f_2$ $e_2 = r \cdot f_1$	Гиратор
$\xrightarrow{1} \text{O} \xrightarrow{3}$ ↑ 2	$e_1 = e_2 = e_3$ $f_1 + f_2 - f_3 = 0$	Узел общего усилия
$\xrightarrow{1} \text{I} \xrightarrow{3}$ ↑ 2	$f_1 = f_2 = f_3$ $e_1 + e_2 - e_3 = 0$	Узел общего потока

Рис. 3. Определения основных многосвязных элементов.

Источники. *Источник усилия* имеет форму записи SE^e и определяется выражением $e = e(t)$.

Источник потока имеет форму записи SF^f и определяется выражением $f = f(t)$.

Аккумуляторы. *Емкость* имеет форму записи $(e/f)C$ и определяется следующим образом:

$$e = \Phi(q) \text{ и } q(t) = q(t_0) + \int_{t_0}^t f(\lambda) d\lambda.$$

Кононюк А.Е. Графы

Таким образом, усилие является статической функцией перемещения, а перемещение — интегралом по времени от потока.

Инерционность имеет форму записи $(e/f)I$ и определяется соотношениями

$$f = \Phi(p) \quad \text{и} \quad p(t) = p(t_0) + \int_{t_0}^t e(\lambda) d\lambda.$$

Таким образом, поток является статической функцией момента, а момент — интегралом по времени от усилия.

Потери. *Сопротивление* обозначается как $(e/f)R$ и определяется соотношением

$$\Phi(e, f) = 0.$$

т. е. в полусвязи существует статическая зависимость между усилием и потоком.

Узлы: 2-связные элементы.

Трансформатор обозначается как

$$\frac{e_1}{f_1} \quad TF \quad \frac{e_2}{f_2},$$

является линейным 2-связным элементом и определяется соотношениями

$$e_1 = m \cdot e_2$$

$$m \cdot f_1 = f_2$$

где m — модуль.

Гиратор является линейным 2-связным элементом, обозначается как

$$\frac{e_1}{f_1} \quad GY \quad \frac{e_2}{f_2}$$

и определяется соотношениями

$$e_1 = r \cdot f_2$$

$$e_2 = r \cdot f_1,$$

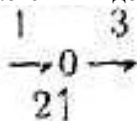
где r — модуль.

Как трансформатор, так и гиратор сохраняют мощность (т. е. $P_1 = P_2$ в каждом из обоих случаев) и обязательно имеют по две полусвязи, почему и называются существенными 2-связными узлами.

Узлы: 3-связные элементы.

Кононюк А.Е. Графы

Узел *общего усилия* обозначается в виде

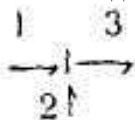


и является линейным 3-связным элементом, определяемым соотношениями

$$\begin{aligned} e_1 = e_2 = e_3 & \quad (\text{общее усилие}), \\ f_1 + f_2 - f_3 = 0 & \quad (\text{суммирование потоков}). \end{aligned}$$

Другие наименования этого элемента — *поточковый узел* и *нулевой узел*.

Узел *общего потока* обозначается в виде



и является линейным 3-связным элементом, определяемым соотношениями

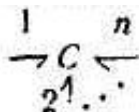
$$\begin{aligned} f_1 = f_2 = f_3 & \quad (\text{общий поток}), \\ e_1 + e_2 - e_3 = 0 & \quad (\text{суммирование усилий}). \end{aligned}$$

Другие наименования этого элемента — *узел усилий* и *единичный узел*. Как узел общих усилий, так и узел общего потока сохраняют мощность (т. е. полезная мощность в них в любой момент времени равна нулю), потому что они и называются узлами. При изменении выбранных направлений мощностей должны соответственно измениться знаки в суммировании.

Обобщающие определения

Многосвязные поля

Поля памяти. *Многосвязные емкости*, или *C-поля*, обозначаются как



и характеризуются соотношениями

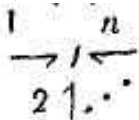
$$e_i = \Phi_i(q_1, q_2, \dots, q_n); \quad i = 1, 2, \dots, n$$

и

Кононюк А.Е. Графы

$$q_i(t) = q_i(t_0) + \int_{t_0}^t f_i(\lambda) d\lambda; \quad i=1, 2, \dots, n.$$

Многосвязные инерционности, или I -поля, обозначаются как



и характеризуются соотношениями

$$f_i = \Phi_i(p_1, p_2, \dots, p_n); \quad i=1, 2, \dots, n$$

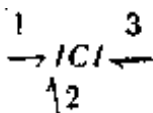
и

$$p_i(t) = p_i(t_0) + \int_{t_0}^t e_i(\lambda) d\lambda, \quad i=1, 2, \dots, n.$$

Для задания S - или I -полям определенных «энергетических» функций состояния необходимо с помощью Φ_i -функций задать соответствующие условия интегрирования. В терминах многосвязности приведенные выше соотношения достаточны для того, чтобы определить S -поле или I -поле соответственно.

Смешанные многосвязные поля памяти появляются при одновременном наличии эффектов запоминания обоих типов. Символ такого поля состоит из последовательности S - и I -элементов с соответствующими нумерованными связями.

Например,



указывает, что существует множество отношений

$$f_1 = \Phi_1(p_1, q_2, p_3),$$

$$e_2 = \Phi_2(p_1, q_2, p_3),$$

$$f_3 = \Phi_3(p_1, q_2, p_3),$$

и

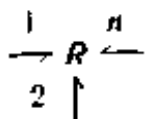
Конюк А.Е. Графы

$$p_1(t) = p_1(t_0) + \int_{t_0}^t e_1(\lambda) d\lambda;$$

$$q_2(t) = q_2(t_0) + \int_{t_0}^t f_2(\lambda) d\lambda;$$

$$p_3(t) = p_3(t_0) + \int_{t_0}^t e_3(\lambda) d\lambda.$$

Многосвязные поля потерь, или R-поля, записываются в виде

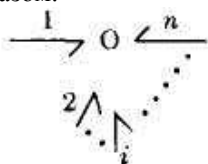


и характеризуются соотношениями

$$\Phi_i(e_1, f_1, e_2, f_2, \dots, e_n, f_n) = 0, \quad i = 1, 2, \dots, n.$$

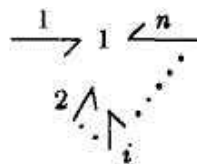
Если необходимо, чтобы R-поле представляло чистые потери, то функция мощности, соответствующая R-полю, должна быть положительной.

К *многосвязным узлам* относятся нулевые и единичные узлы с n полусвязями, $n \geq 2$. В общем случае каждый узел задается следующим образом:



$$e_1 = e_2 = \dots = e_n$$

$$\sum_{i=1}^n f_i = 0$$



$$f_1 = f_2 = \dots = f_n$$

$$\sum_{i=1}^n e_i = 0$$

Модулированные 2-связные узлы.

Модулированный трансформатор, или MTF, имеет форму записи

$$\begin{array}{ccc} & \downarrow m(x) & \\ 1 & & 2 \\ \rightarrow & MTF & \rightarrow \end{array}$$

и определяется соотношениями

Кононюк А.Е. Графы

$$e_1 = m(\mathbf{x}) \cdot e_2$$

$$m(\mathbf{x}) \cdot f_1 = f_2,$$

где $m(x)$ —функция множества переменных x . Модулированный трансформатор сохраняет мощность, т. е. $P_1(t) = P_2(t)$.

Модулированный гиратор, или *MGY*, имеет форму записи

$$\begin{array}{ccc} & r(\mathbf{x}) \downarrow & \\ 1 & & 2 \\ \rightarrow & MGY & \rightarrow \end{array}$$

и определяется следующими соотношениями:

$$e_1 = r(\mathbf{x}) \cdot f_2$$

$$e_2 = r(\mathbf{x}) \cdot f_1,$$

где $r(x)$ — функция множества переменных x . Модулированный гиратор сохраняет мощность, т. е. $P_1(t) = P_2(t)$.

Узловая структура. Узловая структура графа связей содержит нулевые, единичные, гираторные и трансформаторные узлы и их связи и полусвязи. Узловая структура является n -связным элементом, сохраняющим мощность (т. е. полезная мощность в структуре равна нулю). Узловая структура может быть модулированной (если она содержит *MGY* или *MTF*) или немодулированной.

Например, узловая структура графа на рис. 2, *b* представляет собой 4-связный элемент с полусвязями 1, 2, 5, 6 и связями 3, 4. Она содержит нулевой, трансформаторный и единичный узлы.

Физические интерпретации

Физические интерпретации, приведенные в данном разделе, определены сжато.

Механическая интерпретация. Для перевода на язык механики необходимо произвести следующие замены переменных:

1. Усилие e интерпретируется как сила.
2. Поток f интерпретируется как скорость.
3. Момент p интерпретируется как количество движения.
4. Перемещение q интерпретируется как механический сдвиг.

Тогда основные элементы графов связей получают следующую интерпретацию:

1. Источник усилия SE — как источник силы.

Кононюк А.Е. Графы

2. Источник потока SF — как источник скорости (или может быть представлен геометрической реакцией связи).
3. Сопротивление R — как трение и другие виды механических потерь.
4. Емкость C — как эффекты аккумуляции потенциальной энергии или энергии упругости.
5. Инерционность I — как аккумуляция кинетической энергии (или эффект массы).
6. Трансформатор TF — как прямой рычаг или действие сил сцепления (движение ограничено малыми углами).
7. Гиратор GY — как гироскопическая связь или взаимодействие между двумя полусвязями.
8. Нулевой узел — как сумма сил инцидентных полусвязей (или полусвязей системы, связанных в нулевой узел).
9. Единичный узел — как сумма реакций скоростей инцидентных полусвязей (или полусвязей системы, связанных в единичный узел).

Интерпретация естественным образом распространяется на случай механики вращения и основывается на следующих аналогиях:

- 1) усилию e ставится в соответствие вращающий момент, а 2) потоку f — угловая скорость.

Так как перевод терминов аналогичен общему языку механики, нет необходимости повторять его далее.

Электрические цепи. В электрических цепях ключом к интерпретации служит представление полусвязи в виде вывода проводника. Тогда аналогия для переменных выглядит следующим образом:

1. Усилие e интерпретируется как напряжение.
2. Поток f интерпретируется как ток.
3. Момент p интерпретируется как потокосцепление.
4. Перемещение q интерпретируется как заряд.

Основные элементы графа связей получают следующие интерпретации:

1. Источник усилия SE — как источник напряжения.
2. Источник потока SF — как источник тока.
3. Сопротивление R — как электрическое сопротивление.
4. Емкость C — как эффект емкости (накопленная электрическая энергия).
5. Инерционность I — как индуктивность (накопленная магнитная энергия).
6. Трансформатор TF — как идеальная трансформаторная связь.
7. Гиратор GY — как гироскопическая связь.

Кононюк А.Е. Графы

8. Нулевой узел — как параллельное соединение полусвязей (общее напряжение на концах проводников).

9. Единичный узел — как последовательное соединение полусвязей (общий ток через концы проводников).

Гидравлические цепи. Для струйных систем, в которых такая важная характеристика, как мощность струи, задается в виде произведения давления на объем потока, полезно использовать следующие интерпретации переменных:

1. *Усилие e* интерпретируется как *давление*.

2. *Поток f* интерпретируется как *объемный расход*.

3. *Момент p* интерпретируется как импульс давления.

4. Перемещение q интерпретируется как объем.

Основные элементы графа связей получают следующие интерпретации:

1. Источник усилия SE — как источник давления.

2. Источник потока SF — как источник объемного расхода.

3. Сопротивление R — как эффекты потерь (например, из-за утечки, наличия клапанов, отверстий и т. п.).

4. Емкость C — как скопление жидкости или резервуарные эффекты (аккумуляция напора).

5. Инерционность I — как инерционные эффекты вязкости.

6. Нулевой узел — как множество полусвязей с общим давлением (например, тройник трубы).

7. Единичный узел — как множество полусвязей, имеющих общий объемный расход (т. е. последовательные соединения).

Другие интерпретации. Этот краткий перечень физических интерпретаций элементов графов связей ограничивается простейшими, наиболее очевидными применениями. Они появились в процессе исторического развития, и естественным образом обуславливают различия в терминологии. Понятия графов связей могут применяться для описания исключительно богатого разнообразия сложных динамических систем. Пределы применимости не ограничиваются понятиями энергии и мощности в смысле физики: они включают любые области, где существуют величины, аналогичные энергии и мощности.

В этом описании языка графов связей опущены два важных понятия. Первое — понятие *активности связи*, при наличии которой одну из двух переменных мощности можно удалить, в результате чего вместо связи появляется простая сигнальная передача. Для активных систем это очень полезное правило моделирования.

Другим не обсуждавшимся здесь понятием является *операционная причинность*. Именно из-за существования причинности операций в

графах связей возможно систематическое представление и приведение алгебраических и дифференциальных уравнений, вытекающих из этой графовой модели к форме пространства состояний.

Далее мы рассмотрим на ряде примеров применение графов связей для моделирования различных динамических систем.

1.2. Метод использования графов связей при моделировании динамических систем

(Метод разработан Н. Мартенсом и А. Беллом)

Очень важным этапом при конструировании реальных динамических устройств является построение математических моделей таких устройств.

Без хороших моделей трудно разобраться во взаимодействии компонентов и невозможно произвести рациональный выбор параметров конструкции. Для того чтобы провести анализ работы и оптимизацию физической системы на ЭВМ, лучше всего описать ее системой дифференциальных уравнений первого порядка или уравнениями состояния.

Оказывается, что для аналитического описания системы крайне важным, если не необходимым, является овладение систематическими методами построения модели системы. Особенно пригодными при построении моделей систем, состоящих из элементов со многими входами, в которых энергия может принимать различные формы, являются методы построения линейных графов и графов связей. Метод линейных графов первоначально был разработан применительно к анализу электрических цепей. Далее в различных работах были показаны его потенциальные возможности для анализа систем с сосредоточенными параметрами, а позднее он стал широко использоваться при моделировании физических систем. Метод графов связей первоначально был разработан для описания систем с разными видами энергии; в его основе лежал анализ потока энергии между взаимодействующими частями независимо от вида энергии. В настоящее время широко используется метод графов связей при моделировании и построении моделей систем для анализа на ЭВМ.

Оба метода являются в принципе одинаково действенными средствами вывода уравнений состояния динамических систем. Однако с точки зрения конструктора метод графов связей более привлекателен своей простотой и наличием большей информации. (С другой стороны, для теоретика в области построения систем более полезным может оказаться метод линейных графов.) В настоящем разделе это

Кононюк А.Е. Графы

утверждение подкрепляется описанием применения данного метода к конкретному случаю воздушного вибронасоса, которое иллюстрирует преимущества метода графов связей при последовательном анализе работы устройства и выборе основных его параметров. Общность метода не зависит от выбора конкретного рассматриваемого устройства.

Графы связей и моделирование

Преимущество моделей в виде графов связей вытекает из следующих свойств:

- а) Граф связей однозначно описывает структуру системы.
- б) На основании данного графа без труда записываются канонические уравнения.
- в) Легко кодируются физические свойства и причинная зависимость между переменными, а также показывается вид функциональной зависимости для основных связей.

В основе моделирования физических систем лежат две противоположные тенденции: а) модель должна быть исчерпывающей, завершенной и соответствующей устройству; б) модель должна быть простой, отражать основные свойства устройства без ненужных подробностей. С одной стороны, модель все более усложняется путем включения в рассмотрение дополнительных физических свойств (массы, трения, мощности и т. д.), другая же тенденция ограничивает рассмотрение слишком большого количества деталей, особенно тех, динамические эффекты которых лежат за пределами частотного диапазона, представляющего интерес для конструктора. Поэтому в окончательном виде модель должна представлять определенный компромисс между этими двумя тенденциями. Очень часто первым шагом такого компромисса является решение считать систему состоящей из элементов с сосредоточенными, а не с распределенными параметрами.

Применение изложенного здесь метода следует начинать с построения наиболее простой модели, в которой представлены основные связи в устройстве. Затем эта исходная модель постепенно усложняется до такой степени, при которой можно учесть основные характеристики устройства и представить его функциональные свойства. Решение относительно линейности связей между элементами устройства можно отложить до завершения построения всей структуры модели.

Воздушный насос

На рис. 4 схематически показан воздушный вибронасос, в котором электромагнитный привод приводит в движение небольшой постоянный магнит, укрепленный на шарнирном рычаге, который в свою очередь приводит в действие резиновый сифонный насос.

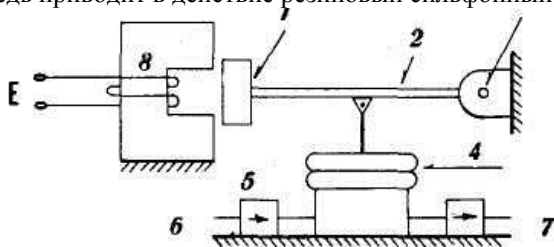


Рис. 4. Схема воздушного насоса.

1 — магнит; 2 — рычаг; 3 — шарнир; 4 — сиффон; 5 — клапан;
6 — вход; 7 — выход.

Резиновые обратные клапаны насоса подают воздух. Резонансная частота системы магнит — рычаг — сиффон должна быть ниже 50 Гц для обеспечения максимума потока воздуха при минимальном расходе энергии. Для линейной системы механический импеданс на резонансной частоте стержня будет минимальным; поскольку инерционные и упругие реактивные сопротивления в этом случае равны и противоположны по знаку, результатом приложения и нагрузкой являются активные потери энергии.

Процесс моделирования

Первый шаг любого процесса моделирования заключается в выделении рассматриваемой системы из окружающей ее среды. Для этого определяют полусвязи, которые связывают систему и окружающую среду, и полагают, что присоединение соответствующих источников к этим полусвязям системы не вносит существенного изменения окружающей среды. Затем в модель включают основные связи между рассматриваемыми входами в систему. После этого последовательно добавляют 1-связные элементы, важные с точки зрения физических процессов в устройствах, одновременно определяя причинную зависимость каждого добавленного элемента и грубо оценивая работу системы. Иногда для полного описания системы требуется включить в рассмотрение многосвязные элементы, если по каким-либо причинам 1-связные элементы оказываются неудовлетворительными. Процесс

Кононюк А.Е. Графы

состоит в построении последовательности все более сложных моделей, из которой можно выбрать модель, соответствующую заданным значениям параметров.

Общие соображения относительно процесса моделирования

Основная схема воздушного насоса состоит из трех последовательно соединенных устройств: электромагнитного привода, рычага и воздушного сильфона. Воздушный промежуток в электромагнитном приводе выбран достаточно большим, чтобы оправдать предположение о постоянстве коэффициента связи при относительно малой амплитуде колебаний. Таким образом, можно воспользоваться тремя линейными моделями, соединенными между собой.

Наиболее важные 1-связные физические свойства, которые следует учитывать при моделировании воздушного насоса, определяются массой магнита и податливостью резинового сильфона. Они образуют резонансную подсистему, которая представляет основной узел конструкции. Из других свойств (рассмотрение которых может оказаться необязательным) отметим следующие, перечисленные в произвольном порядке: сопротивление и индуктивность обмотки электромагнита, внешнее трение резинового сильфона, масса сильфона, трение в шарнире рычага, трение воздуха во входном и выходном обратных клапанах, а также сжимаемость воздуха, временно попадающего в камеру сильфона при работе насоса.

В процессе моделирования, рассмотрение которого последует ниже, будет однозначно определено, какое из этих свойств следует учесть. Из конструктивных соображений и анализа условий работы воздушного насоса можно заключить на данном этапе, что рассматриваемая модель должна удовлетворять следующим требованиям: а) учитывать основные связи в системе; б) описывать резонансные явления в подсистеме масса — пружина; в) порядок модели должен по крайней мере равняться двум.

Построение модели

Процесс моделирования заключается в построении надлежащего графа связей. Построив граф связей, адекватный рассматриваемому устройству, на его основе можем сразу же записать уравнения состояния. Уравнения состояния можно вывести или вручную (что здесь не представит труда), или воспользоваться вычислительной программой ENPORT. Эта программа выводит уравнения состояния на основании заданного графа связей.

Кононюк А.Е. Графы

Построение графа связей будем проводить постепенно. На каждом этапе построения будем проверять уравнения, определяемые данным графом связей, и принимать решение относительно достаточности и полноты модели.

А. Взаимодействие системы с окружающей средой. Очевидно, насос является преобразователем с двумя входами, получающим энергию из окружающего пространства в виде постоянного напряжения и нагнетающим воздушный поток в некоторую емкость с неизвестным пока давлением. Следовательно, система замкнута на одном конце источником энергии, но на другом конце должна быть, по крайней мере на данном этапе, открытой, пока не будет определена пневматическая нагрузка.

В. Структура связей в системе. Граф связей на рис. 5 иллюстрирует структуру связей для воздушного насоса.

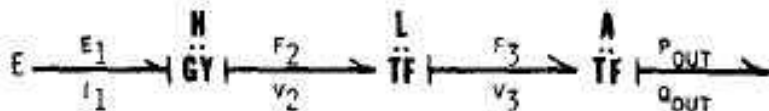


Рис.5. Граф связей для модели В.

Символы объяснены в таблице.

	Обобщенная переменная	
	Усилие	Поток
Физический смысл переменной		
Электрическая Механическая Пневматическая	Напряжение e Сила f Давление p	Ток i Скорость v Объемный поток q

Электромагнитный привод (линейный, как принято ранее) является гиратором, а рычаг и сильфон — преобразователями этого движения. Причинная связь однозначно определяется заданием переменной усилия, т. е. напряжения на приводе. Устанавливая причинные связи вдоль графа связей, определим переменную, связанную с выходным потоком q_{out} . Соответствующие уравнения будут алгебраическими, поскольку данный граф связей не учитывает никаких элементов накопления энергии.

Кононюк А.Е. Графы

Уравнения для трех элементов связи записываем в виде причинных зависимостей: электромагнитный привод

$$\begin{aligned}v_2 &= 1/H e_1 \\i_1 &= 1/H f_2.\end{aligned}\tag{1}$$

рычаг

$$\begin{aligned}v_2 &= L v_1; \\f_2 &= L f_1.\end{aligned}\tag{2}$$

воздушный сильфон

$$\begin{aligned}q_{out} &= A v_2; \\f_2 &= A p_{out};\end{aligned}\tag{3}$$

где L — отношение плечей рычага, H — постоянная электромагнитной связи, A — площадь сильфона. Объединяя уравнения (1)–(3), получим

$$\begin{aligned}q_{out} &= \frac{AL}{H} e_1; \\i_1 &= \frac{AL}{H} p_{out}.\end{aligned}\tag{B}$$

Уравнения, полученные на этом этапе построение графа связей, называемые моделью В, принципиально недостаточны в том смысле, что они не описывают резонансных явлений, при этом подразумевается идеальность выхода источника потока, т. е. возрастание тока до бесконечности при закрытом выходном клапане.

Добавление какого-либо графа приводит к изменению нумерации всех последующих графов. Заметим, что графы перенумерованы последовательно; это сделано для всех последующих графов связей.

С. Введение резонанса. Для того чтобы правильно описать резонансные характеристики воздушного насоса, следует включить в рассмотрение массу магнита и податливость воздушного сильфона. Соответствующий граф связей приведен на рис. 6.

Кононюк А.Е. Графы

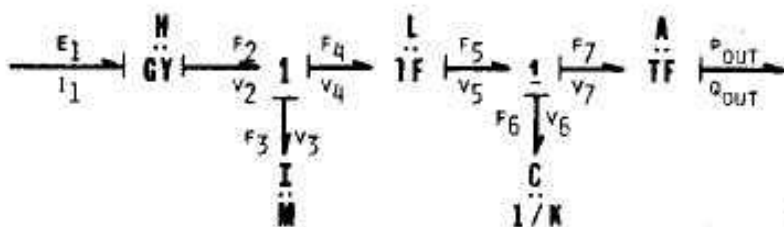


Рис. 6. Граф связей для модели С.

Добавление какого-либо графа приводит к изменению нумерации всех последующих графов. Заметим, что графы перенумерованы последовательно; это сделано для всех последующих графов связей.

При задании причинной связи обнаруживается причинное противоречие: массу M нельзя считать независимым элементом накопления энергии. Далее, скорость массы M определяется величиной e_1 независимо от требуемого тока, очевидно, невыполнимо. Кроме того, сила, развиваемая пружиной K , является независимой переменной состояния. Как альтернативу естественнее было бы выбрать в качестве переменной состояния перемещение пружины, если жесткость пружины K является функцией перемещения.

В соответствии с графом связей С, уравнения для модели С имеют вид

$$\begin{aligned} d/dt i_1 &= \frac{KL}{H} e_1; \\ i_1 &= \frac{M}{H^2} d/dt e_1; \\ q_{out} &= \frac{AL}{H} e_1. \end{aligned} \tag{С}$$

Недостатки модели С очевидны:

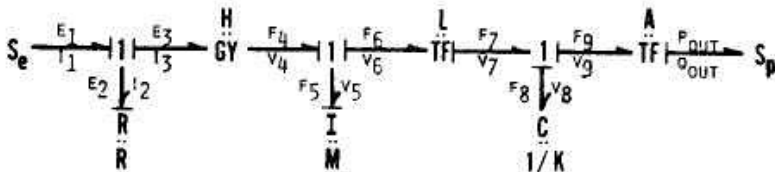
1. Она не является моделью второго порядка, следовательно, не описывает резонансные явления.
2. Любая разрывность в напряжении источника приведет к бесконечному возрастанию тока. Очевидно, ток не может возрастать до бесконечности из-за наличия импеданса обмотки электромагнита. Это соображение приводит к следующему этапу построения модели.

Кононюк А.Е. Графы

Д. Введение импеданса обмотки электромагнита.

Изучение модели С приводит к необходимости учета импеданса обмотки электромагнита. Чтобы сохранить минимальный порядок модели, в качестве первого приближения импеданс принимаем равным активному сопротивлению обмотки R . Предполагается, что такое дополнение разрешит причинное противоречие модели С.

Новый граф связей показан на рис. 7, где введено сопротивление обмотки R последовательно с источником напряжения S_e .



7. Граф связей для модели Д.

Чтобы полностью определить причинную зависимость для этого графа, вводим определенное давление нагрузки S_p . В этой схеме отсутствуют причинные противоречия, и имеются два независимых элемента накопления энергии (аккумулятора). Значит, этот граф связей приводит к модели второго порядка, что и требуется.

Уравнения, описывающие модель Д, имеют вид

$$d/dt v_3 = K L v_5;$$

$$d/dt v_5 = \frac{1}{M} \left[\frac{H}{R} S_e - \frac{H^2}{R} v_5 - L f_3 - L A S_p \right];$$

$$q_{out} = A L v_5.$$

(D)

Модель Д представляет собой существенное улучшение модели С. Ею можно легко воспользоваться для изучения поведения воздушного насоса в резонансном режиме, но она все еще содержит очевидные недостатки.

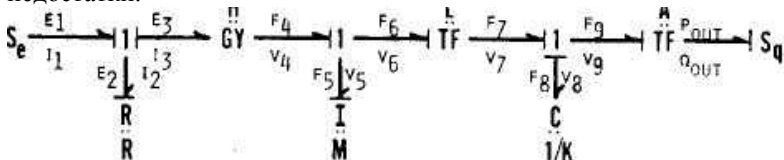


Рис. 8. Граф связей для модели Д с источником потока.

Например, если заменить источник выходного давления источником потока S_q , то граф (8) окажется противоречивым с точки зрения

Кононюк А.Е. Графы

движения массы магнита; это противоречие возникает из-за того, что v_5 определяется выходным потоком: $v_5=0$, если $S_q=0$. Это не соответствует фактическому процессу в насосе. В действительности если поток заперт, то масса продолжает движение с меньшей амплитудой, а собственная частота увеличивается, т. е. имеется линейная зависимость между частотой и увеличением фазового угла между напряжением и скоростью. При закрытом выходном отверстии система оказывается жестче из-за упругости воздуха в сильфоне.

Е. Введение сжимаемости воздуха. Эффекты, связанные со сжимаемостью воздуха, можно моделировать в виде пневматической податливости в сильфоне. Соответствующий граф связей, в котором находит отражение указанное дополнение, приведен на рис. 9.

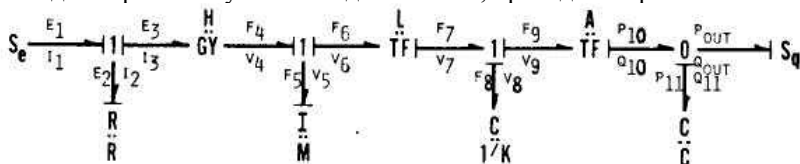


Рис.9. Граф связей для модели Е.

Добавление емкости вводит новую переменную состояния p_{11} . Для ТОГО чтобы избежать причинного противоречия, переменную усилия в новом нулевом узле следует определить через переменную состояния p_{11} , а не p_{out} . В результате соответствующая математическая модель будет представлять собой источник давления, а не источник потока; следовательно, должен быть определен выходной поток. Это легко проверяется по уравнениям, соответствующим модели Е:

$$d/dt f_8 = K L v_5;$$

$$d/dt v_5 = \frac{1}{M} \left[\frac{H}{R} S_e - \frac{H^2}{R} v_5 - L f_8 - L A p_{11} \right];$$

$$d/dt p_{11} = \frac{1}{C} [A L v_5 - S_q];$$

$$p_{out} = p_{11}.$$

(Е)

Введение выходного сопротивления будет делать модель более гибкой в том смысле, что можно будет определить либо p_{out} , либо q_{out} .

Конюк А.Е. Графы

$$d/dt f_s = K L v_s;$$

$$d/dt v_s = \frac{1}{M} \left[\frac{H}{R} S_e - \frac{H^2}{R} v_s - L f_s - L A p_{11} \right];$$

$$d/dt p_{11} = \frac{1}{C} \left[A L v_s - \frac{1}{R_0} (p_{11} - p_{out}) \right];$$

$$q_{out} = \frac{p_{11} - p_{out}}{R_0}$$

(F)

Последние два уравнения можно записать следующим образом:

$$d/dt p_{11} = \frac{1}{C} [A L v_s - R_0 q_{out}];$$

$$p_{out} = p_{11} - R_0 q_{out}.$$

При более сложных условиях присоединения нагрузки нужно детальнее изучать нелинейные свойства клапанов и их влияние на работу системы, в этом случае можно воспользоваться графом связей, показанным на рис. 11.

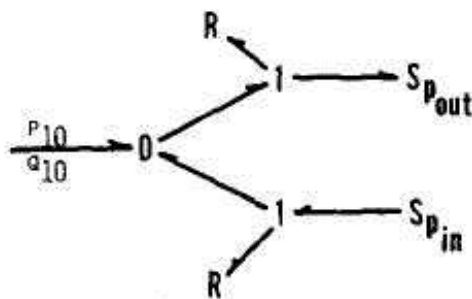


Рис. 11. Нелинейный граф связей для модели F.

Здесь R представляет диодную характеристику.

Математическая модель, заданная уравнениями (F), представляет собой минимально приемлемую модель воздушного насоса. Это модель минимального порядка, которая удовлетворяет всем основным требованиям; она не содержит никаких причинных противоречий, целесообразна с точки зрения выбора входных и выходных

Кононюк А.Е. Графы

переменных, позволяет учесть входной импеданс и, что важнее всего, представлена в форме, пригодной для решения задачи на ЭВМ.

Можно ввести еще два дополнения, которые учитывали бы индуктивность обмотки и собственное демпфирование резинового сильфона. Относительную важность этих двух физических свойств можно оценить из анализа численных значений постоянных I и B , проделанного в следующем разделе.

G. Введение индуктивности обмотки и демпфирования сильфона.

При введении I и B получим новый граф, показанный на рис. 12.

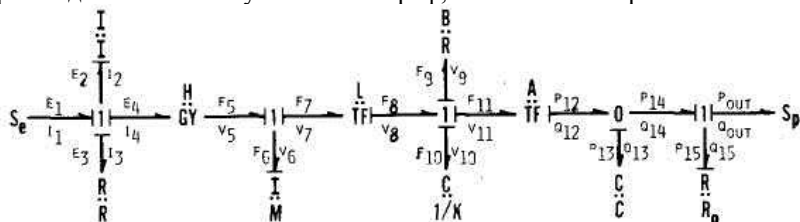


Рис. 12. Граф связей для модели G.

Соответствующие уравнения имеют вид

$$d/dt i_2 = \frac{1}{I} [S_e - R i_2 - H v_{02}];$$

$$d/dt v_0 = \frac{1}{M} [H i_2 - L f_0 - L^2 B v_0 - A L p_{12}];$$

$$d/dt f_0 = K L v_0;$$

$$d/dt p_{12} = \frac{1}{C} (A L v_0 - \frac{1}{R_0} (p_{12} - p_{out}));$$

$$q_{out} = \frac{p_{12} - p_{out}}{R_0}.$$

(G)

В матричном виде модель G записывается следующим образом:

$$\begin{aligned}
 \frac{d}{dt} \begin{bmatrix} i_2 \\ v_0 \\ f_0 \\ p_{13} \end{bmatrix} &= \begin{bmatrix} -\frac{R}{I} & -\frac{H}{I} & 0 & 0 \\ \frac{H}{M} & -\frac{L^2 B}{M} & -\frac{L}{M} & -\frac{AL}{M} \\ 0 & KL & 0 & 0 \\ 0 & \frac{AL}{C} & 0 & -\frac{1}{CR_0} \end{bmatrix} \begin{bmatrix} i_2 \\ v_0 \\ f_0 \\ p_{13} \end{bmatrix} + \\
 &+ \begin{bmatrix} \frac{1}{I} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{CR_0} \end{bmatrix} \begin{bmatrix} S_c \\ S_p \end{bmatrix};
 \end{aligned}$$

$$q_{out} = \frac{1}{R_0} (p_{13} - p_{out}).$$

Исследование модели

Построены две математические модели воздушного насоса: F — модель минимального порядка и G — модель порядка на единицу выше. Условие построения этих моделей состояло в адекватности описания динамики устройства. Другим мотивом может служить оптимизация устройства. Для проверки правильности построения модели и ее надежности следует задаться необходимыми численными значениями параметров. Данное исследование преследует две цели:

- а) из сравнения относительных значений параметров, связанных с физическими свойствами, по которым отличаются модели F и G, определить наиболее пригодную модель;
- б) путем моделирования на ЭВМ с использованием полученных моделей определить переходную и частотную характеристики насоса. Значения параметров, необходимых для анализа полученных моделей, были определены экспериментально и приведены в табл. 1.

Числовые значения

Параметр	Значение
H	13 Н/а [(м/с)/В]
L	0,58
A	$1,71 \cdot 10^{-4} / \text{м}^3$
M	$1,11 \cdot 10^{-3} \text{ Н} \cdot \text{с}^2 / \text{м}$
K	117 кг/м
R	2250 Ом
I	3,9 Гн
C	$18,65 \cdot 10^{-10} \text{ м}^5 / \text{Н}$
R_0	$3,32 \cdot 10^5 \text{ Н} \cdot \text{с} / \text{м}^2$
B	Пренебрежимо мало

При этих численных значениях параметров в модели G уравнения состояния насоса принимают вид

$$d/dt \begin{bmatrix} i_2 \\ v_6 \\ f_9 \\ p_{13} \end{bmatrix} = \begin{bmatrix} -577 & -3,34 & 0 & -0,0907 \\ 1190 & 0 & 902 & -0,0907 \\ 0 & 116 & 0 & 0 \\ 0 & 54000 & 0 & -01700 \end{bmatrix} \times$$

$$\times \begin{bmatrix} i_2 \\ v_6 \\ f_9 \\ p_{13} \end{bmatrix} + \begin{bmatrix} 24,35 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1700 \end{bmatrix} \begin{bmatrix} S_e \\ S_p \end{bmatrix}$$

Удобный способ проверки относительной важности параметров заключается в масштабировании матрицы A уравнений состояния с тем, чтобы урвать все численные значения. Этот процесс в принципе аналогичен масштабированию при вычислениях на аналоговых машинах. Если взять переменные i_5 и p_4 соответственно в масштабах 30 и 10^{-3} , получим уравнения

Кононюк А.Е. Графы

$$d/dt \begin{bmatrix} 30i_2 \\ v_6 \\ f_9 \\ p_{13} \times 10^{-3} \end{bmatrix} = \begin{bmatrix} -577 & -100 & 0 & 0 \\ 39,7 & 0 & -902 & -90,7 \\ 0 & 116 & 0 & 0 \\ 0 & 54 & 0 & -1700 \end{bmatrix} \times \\ \times \begin{bmatrix} 30i_2 \\ v_6 \\ f_9 \\ p_{13} \times 10^{-3} \end{bmatrix} + \begin{bmatrix} 730 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1,7 \end{bmatrix} \begin{bmatrix} S_e \\ S_p \end{bmatrix}.$$

Теперь можно заключить, что в уравнениях состояния достигнут благоприятный численный баланс, означающий, что все физические свойства, отраженные в модели G, одинаково важны. Элементы главной диагонали соответствуют в первом приближении постоянным времени системы: более быстрые процессы определяются пневматической постоянной времени, а более медленные — электрической постоянной времени. Кроме этих действительных корней, имеется пара комплексно сопряженных корней, соответствующих резонансной частоте. Измеримые демпфирующие эффекты обусловлены только сопротивлением обмотки и трением воздуха.

Следует отметить, что при моделировании на ЭВМ было получено максимальное значение скорости выходного потока на частоте 54 Гц, а в реальном воздушном насосе максимум скорости потока достигается на частоте 57 Гц. Коэффициент усиления при резонансе, полученный из результатов моделирования, был равен 12, а в эксперименте 10,5.

Н. Мартенсом и А. Беллом использован простой пример для демонстрации исключительных возможностей графов связей при моделировании систем. Моделирование систем может стать непосредственным логическим процессом при построении соответствующего графа связей.

Процесс моделирования можно начинать с рассмотрения основной структуры связей и функциональных свойств системы, затем, шаг за шагом, можно добавлять другие свойства системы, такие, как накопление энергии, ее диссипацию, а также зависимости вход-выход.

Преимущество графов связей заключается в возможности проведения исчерпывающего анализа взаимодействия между топологическими связями и 1-связными элементами, и на любом этапе процесса моделирования предсказывать результаты.

1.3. Методика построения многосвязных моделей в механике

Задачи механики, включающие конечные движения твердых тел в силовых полях консервативного и неконсервативного типов, рассматриваются с точки зрения многосвязных систем. Ниже приводится, предложенная Р. Розенбергом, методика построения моделей в форме графов связей, основанная на ключевых геометрических переменных и связывающих эти переменные преобразования скоростей. Показано значение таких моделей для улучшения представления и описания структуры системы, составления уравнений движения в пространстве состояний и прямого моделирования сложных механических задач на ЭЦВМ.

Основная задача формулируется просто: по заданному схематическому представлению или прототипу механической системы, а также дополнительным словесным и математическим описаниям найти удобный способ моделирования системы, позволяющий предсказать ее динамическое поведение.

Важность задачи очевидна. Она разрабатывается со времен Ньютона, привлекла внимание многих выдающихся ученых, включая Гамильтона и Лагранжа. С тех же времен разработка основана не на графическом представлении, а на аналитическом аппарате в форме операторов и уравнений различных типов. Поскольку известно, как формулировать задачи механики различными способами, можно утверждать, что основная задача решена. Тем не менее поиск улучшений подхода к составлению моделей в механике всегда будет представлять интерес и существенные улучшения могут иметь важное значение.

Конкретным направлением поиска в данной работе является возможность использования графического представления моделей в механике. Еще недавно лишь аппарат теории электрических цепей в его обобщенной форме открывал возможность подобного рассмотрения задач механики. По ряду причин, в основном группирующихся вокруг сложности явного представления многосвязного элемента в форме цепи, такие диаграммы не нашли широкого применения.

Пейнтер Н. из Массачусетского технологического института предложил другой графический язык для представления многосвязных элементов в явной форме, названный методом графов связей. Этот метод дает явное представление мощностей и позволяет кратко описать многосвязный элемент, например, с пятью связями при

Кононюк А.Е. Графы

помощи изображения, содержащего буквенный символ и пять линий. С тех пор многие исследователи пытались использовать этот метод как способ формирования модели, вывода уравнений и прямого цифрового моделирования. Хотя был достигнут значительный прогресс и стало возможным решать задачи механики с использованием языка графов связей и родственных методов, основная задача разработки полного, систематического и удобного графического подхода не была решена. Тогда Кэрноп предложил ключевое понятие, а именно, графы связей как отображения сохраняющих мощность преобразований, которое тотчас же было признано недостающим элементом в механике.

Итак, в данной разделе предпринята попытка унифицировать все родственные понятия и методы графов связей с целью получения удобной формулировки задачи механики многосвязных систем. Следует отметить, что Р. Розенбергом использовано множество методов, включая выбор исследователем некоторых ключевых переменных, определение требуемых преобразований простыми аналитическими способами, объединение и представление всех частей при помощи графа связей.

Главная цель метода состоит в представлении и иллюстрации применения некоторых форм графов связей как стандартных моделей в механике многосвязных систем. Метод начинается с простейшей формы, включающей взаимосвязанные инерции и податливости, затем дано обобщение на системы, содержащие статически связанные инерции и статически связанные податливости, и далее осуществлен переход к произвольной симметричной модели графа связей. Метод основан на геометрическом подходе, однако другие формулировки становятся очевидными после завершения геометрического подхода.

Наконец, предложены применения этих моделей для представления и передачи информации, исследований по теории нелинейных систем и прямого цифрового моделирования.

Основная методика моделирования

Следуя сложившейся практике решения задач, начнем с рассмотрения результата, который желательно получить. Задавая схематической диаграммой задачи механики, включающей твердые тела и силовые поля, получим связанную систему дифференциальных уравнений первого порядка относительно переменных, имеющих физический смысл.

В частности, желательно иметь физический смысл уравнений (1) — (4), приведенных ниже

Кононюк А.Е. Графы

$$\begin{array}{ll} \text{С-поле} & F_C = \Phi_C(q_C), \\ \text{(податливости)} & \\ \text{I-поле} & V_I = \Phi_I(P_I), \\ \text{(инерции)} & \\ \text{Структура узлов} & \dot{q}_C = T_{CI}(q_C)^* V_I, \\ \text{(связи системы)} & \dot{P}_I = -T_{CI}^t(q_C)^* F_C, \end{array} \quad (1)-(4)$$

где q_C — множество перемещений, определяющих потенциальную энергию; F_C — множество сил, определяющих потенциальную энергию; P_I — множество импульсов, определяющих кинетическую энергию; V_I — множество скоростей, определяющих кинетическую энергию, и q'_C и P'_I — производные по времени от q_C и P_I соответственно.

Эта частная постановка является гибридной и лежит где-то между классической геометрической формой второго порядка методов Лагранжа и формой первого порядка, связывающей силы и скорости, из обобщенной теории цепей. С точки зрения графа связей она наиболее естественна, что будет показывать при его дальнейшем построении. Анализ показывает, что уравнения (1) получаются для податливых элементов непосредственно и фактически составляют их определение. Эти соотношения могут быть линейными или нелинейными в зависимости от природы рассматриваемых податливых элементов.

Уравнения (2) вытекают непосредственно из инерционных свойств системы. Наиболее удобно выразить их через вращения главных осей инерции и движения центров масс. В этом случае инерционные свойства обычно постоянны и система (2) линейна с постоянными коэффициентами. Вектор V_I должен быть определен относительно инерциальной системы координат.

Однако наиболее сложной особенностью задач механики, как правило, является нелинейная связь переменных, вытекающая из геометрического анализа систем. Такая связь представлена в уравнениях (3) и (4) матрицей преобразования $T_{CI}(q_C)$ и транспонированной матрицей $T_{CI}^t(q_C)$. Каждый элемент матрицы T_{CI} в принципе будет скалярной функцией всех элементов вектора q_C . Главная задача механики многополюсных систем состоит в построении преобразования $T_{CI}(q_C)$. Перед тем как продемонстрировать систематический способ решения этой задачи, заметим, что уравнения (1) — (4) можно объединить путем исключения F_C и V_I , в результате чего получаются следующие уравнения:

Кононюк А.Е. Графы

$$\begin{aligned}\dot{q}_C &= T_{C1}(q_C) * \phi_I(P_I); \\ \dot{P}_I &= - T'_{C1}(q_C) * \phi_C(q_C).\end{aligned}\tag{5)-(6}$$

Эти уравнения представляют собой уравнения нелинейной консервативной системы и могут быть легко видоизменены путем учета влияний диссипативных и других неконсервативных сил, а также геометрических связей, не учтенных ранее в структуре преобразования. Модель графа связей обсуждаемой системы показана на рис. 1. Податливости представлены C -полем, имеющим M связей (и не обязательно состоящим из односвязных C -элементов). Инерции представлены I -полем, имеющим P полюсов (также не обязательно состоящим из односвязных элементов). Структура узлов, отражающая связи системы, показана как $(M-P)$ -полюсный модулированный трансформатор (MTF), элементы которого определяются переменными q_C .

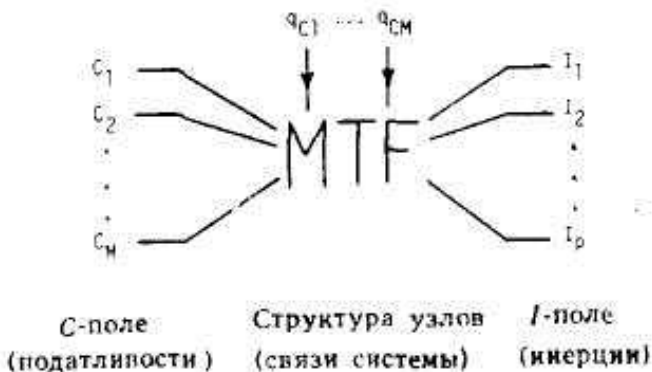


Рис.1. Символическая модель графа связей для основной нелинейной системы (MTF —модулированный трансформатор).

Построение структуры узлов при помощи геометрического подхода

Чтобы построить искомую структуру узлов, определим два дополнительных кинематических вектора, используемых совместно с q_C и V_I . Обозначим через q_K вектор кинематических перемещений, необходимый и достаточный для задания конфигурации системы в некоторый момент времени. Он состоит из совокупности обобщенных

Кононюк А.Е. Графы

координат, используемых в методе Лагранжа. Обозначим через V_K вектор кинематических скоростей, необходимый и достаточный для определения всех движений в некоторый момент времени. Кроме того, предположим, что

$$\dot{q}_K = V_K, \quad (7)$$

т. е. рассматриваемые системы голономны. Это позволяет упростить построение, хотя обобщения допускают рассмотрение неголономных систем.

В качестве первого шага выразим q_C через q_K в общем случае нелинейным преобразованием перемещений

$$q_C = \Phi_{CK}(q_K). \quad (8)$$

Дифференцированием получаем соотношение между векторами скоростей

$$\dot{q}_C = \frac{\partial \Phi_{CK}}{\partial q_K} * \dot{q}_K = T_{CK}(q_K) * \dot{q}_K. \quad (9)$$

Преобразование $T_{CK}(q_K)$ может быть реализовано структурой узлов, состоящей из 0-, 1- и 2-связных модулированных трансформаторов. Уравнения (9) символически представлены графом связей, изображенным на рис. 2.

Кононюк А.Е. Графы

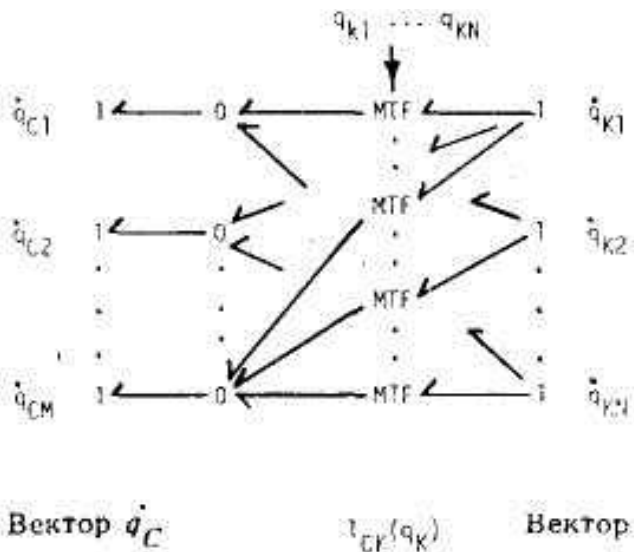


Рис. 2. Структура узлов, представляющая преобразование кинематических скоростей

В качестве следующего шага выразим вектор скоростей V_I , инерционных элементов через вектор V_K скоростей (т. е «движение») системы. Как правило, простейший способ выполнения этой операции состоит в определении вектора q_I представляющего перемещение поля инерционных элементов относительно инерционной системы координат, что дает

$$q_I = \phi_{IK}(q_K). \tag{10}$$

Заметим, что q_I следует выбирать, имея в виду V_I , так чтобы

$$\dot{q}_I = V_I. \tag{11}$$

Дифференцированием каждого из уравнений системы (10) по времени и подстановкой результата в (11) получаем

$$V_I = \frac{\partial \phi_{IK}}{\partial q_K} * \dot{q}_K = T_{IK}(q_K) * \dot{q}_K. \tag{12}$$

Конюк А.Е. Графы

Преобразование, описываемое уравнением (12), символически представлено на рис.3.

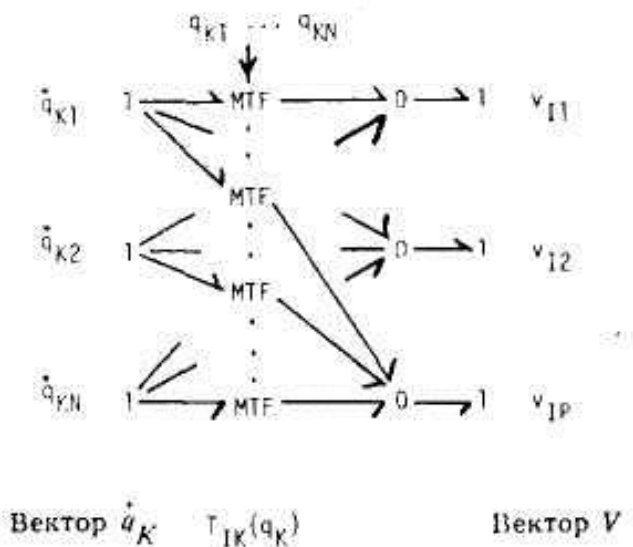


Рис.3. Структура узлов, представляющая преобразование кинематических скоростей в скорости инерции.

Как и прежде, единственно необходимыми элементами графа связей являются 0-, 1-й 2-связные модулированные трансформаторы.

Объединив две части структуры узлов, изображенных на рис. 2 и 3, построим символический конечный граф на рис.4, где для удобства сохранена явная структура узлов.

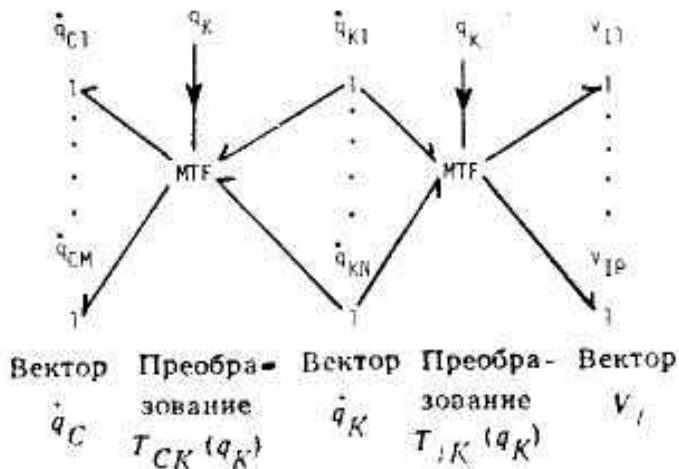


Рис.4. Символическая структура узлов, представляющая связи системы

В качестве конечного шага можно присоединить С- и I-поля к структуре узлов, чтобы получить символический граф связей полной нелинейной консервативной системы, представленный на рис.5.

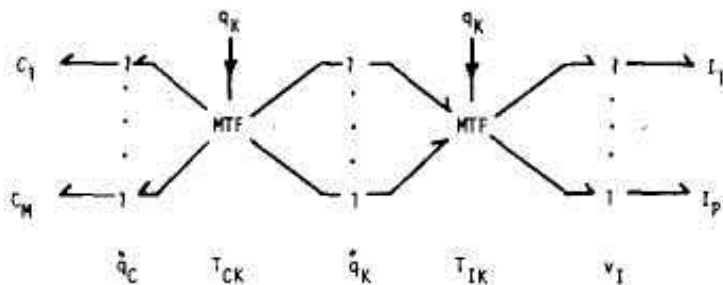


Рис.5. Символический граф связей, представляющий общую нелинейную консервативную систему.

Если для учета влияния диссипации или других неконсервативных сил нужны скорости, отличные от показанных на рис.5, их можно получить построением соответствующих преобразований вектора V_K с использованием вектора q_K в качестве модулирующего. Подобные дополнительные поля можно непосредственно присоединить к структуре узла. Аналогично можно ввести внешние и внутренние связи, налагаемые на скорости.

Кононюк А.Е. Графы

После построения графа связей системы в форме, показанной на рис.5, можно вывести ее уравнения движения с помощью формальных правил метода графов связей. Здесь естественно использовать q_k как геометрическую компоненту вектора состояния совместно с компонентами p_i моментов, поскольку элементы преобразователя выражены через q_k .

Пример применения основной методики моделирования

Рассмотрим пример, показанный на рис. 6.

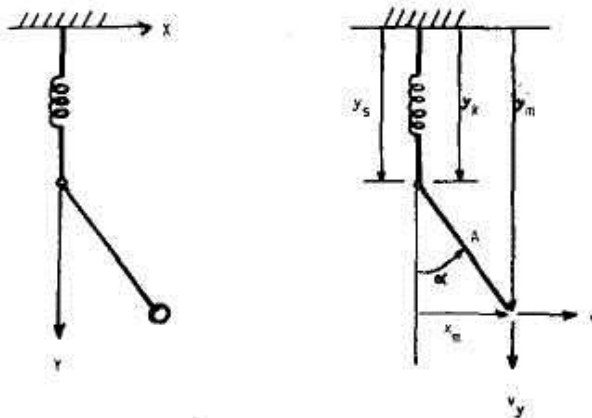


Рис.6. Колебательное звено, состоящее из массы и пружины.

Первый шаг состоит в идентификации трех ключевых векторов, а именно q_s , q_k и V_l (или q_l). Один из вариантов их выбора таков:

$$q_s = [y_s], q_k = \begin{bmatrix} y_k \\ \alpha \end{bmatrix} \quad \text{и} \quad q_l = \begin{bmatrix} x_m \\ y_m \end{bmatrix}.$$

Чтобы получить преобразование $T_{СК}$, используем уравнения (8) и (9), согласно которым

$$y_s = 1 \cdot y_k + 0 \cdot \alpha$$

и

$$\dot{y}_s = 1 \cdot \dot{y}_k + 0 \cdot \dot{\alpha},$$

поэтому

Конюк А.Е. Графы

$$T_{CK} = [1 \quad 0].$$

Для построения преобразования T_{IK} применим уравнения (10) — (12), дающие в рассматриваемом примере следующие соотношения:

$$x_m = A \sin \alpha;$$

$$y_m = y_k + A \cos \alpha;$$

$$v_x = \dot{x}_m;$$

$$v_y = \dot{y}_m$$

$$v_x = 0^* \dot{y}_k + A \cos \alpha^* \dot{\alpha};$$

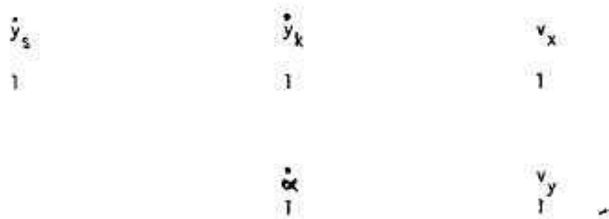
$$v_y = 1^* \dot{y}_k - A \sin \alpha^* \dot{\alpha},$$

поэтому

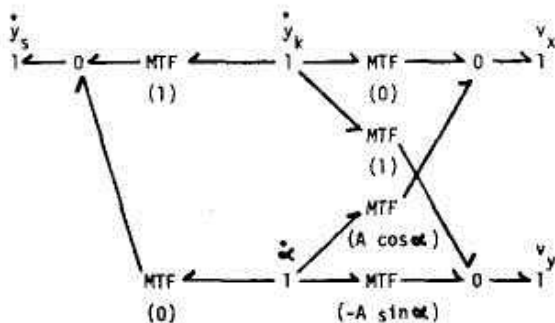
$$T_{IK} = \begin{bmatrix} 0 & A \cos \alpha \\ 1 & -A \sin \alpha \end{bmatrix}.$$

Основной граф связей может быть построен сначала представлением векторов скоростей q'_C , q'_K и V'_I , столбцами 1-связей, как показано на рис.7, а.

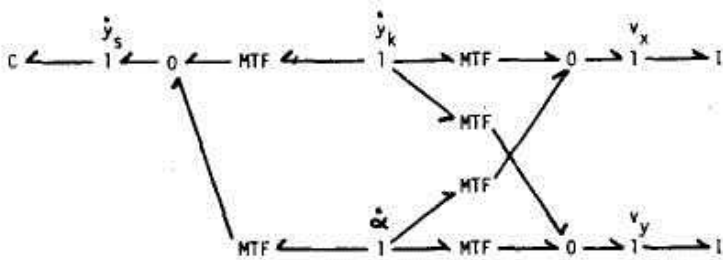
Затем два преобразования T_{CK} и T_{IK} вводят в рис.7, б с использованием модулированных трансформаторов для умножения и 0-узлов для суммирования. Наконец, соотношение для податливости пружины и соответствующие инерционные свойства вводят в рис.7, в.



a



b



c

Рис.7. Модель графа связей колебательного звена из массы и пружины.
a — совокупность ключевых геометрических переменных; *b* — подстановка преобразований $T_{СК}$ и T_{IK} ; *c* — граф связей полной системы.

Кононюк А.Е. Графы

Предположим теперь, что необходимо учесть еще два динамических эффекта, а именно: диссипацию в шарнире маятника и силу тяжести. Момент диссипативных сил возникает непосредственно за счет относительной скорости α ; он показан в виде R-элемента, присоединенного к узлу 1 на рис.8. Сила тяжести создает движение v_y , и ее удобно ввести как постоянный источник силы в соединение v_y (рис.8).

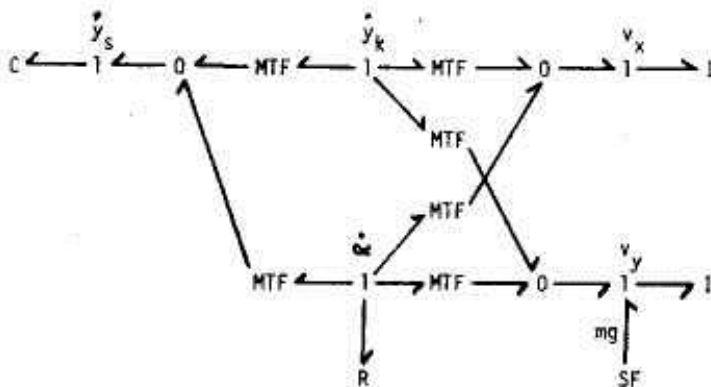


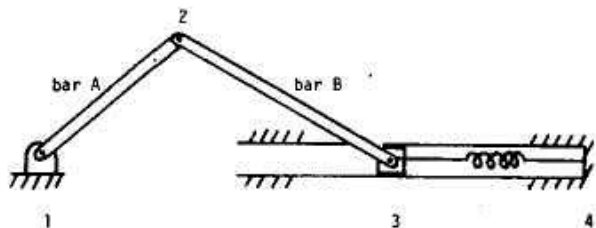
Рис.8. Конечный граф колебательного звена из массы и пружины (фиг. 7) с учетом влияния демпфирования и силы тяжести.

Здесь следует подчеркнуть, что любое из полей или все поля (C , I и R) могут быть нелинейными, но это не вызовет изменений в структуре графа. Конечно, в ньютоновской механике I -поле всегда линейно. Естественно, трудности сведения графа к уравнениям полной системы в пространстве состояний в определенной мере зависят от природы нелинейностей.

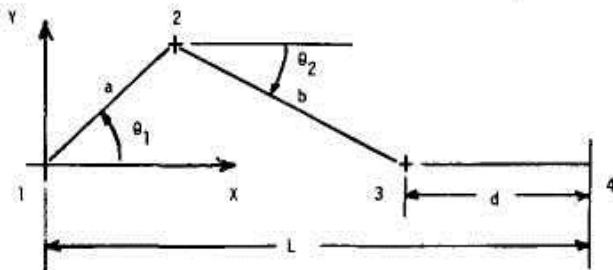
Системы, содержащие статически связанные инерционные элементы

Исследование большого числа часто возникающих задач механики показывает, что обычно вектор скоростей (V_I) инерционного поля имеет больше элементов, чем вектор обобщенных скоростей (V_K). В этом случае на все компоненты V_I , независимы, но вышеописанная методика остается применимой. Тем не менее уместно привести интерпретацию полученной модели.

Рассмотрим в качестве примера кривошипно-шатунный механизм, изображенный на рис.9.



a



b

Рис.9. Кривошипно-шатунный механизм.

a — кривошипно-шатунный механизм; *b* — геометрия механизма.

Идентификация ключевых геометрических переменных приводит к следующим выражениям:

$$q_C = [d], q_K = [\theta_1] \quad \text{и} \quad V_I = \begin{bmatrix} \omega_{12} \\ V_{B2X} \\ V_{B2Y} \\ \omega_{23} \end{bmatrix}$$

где d — длина пружины, θ_1 — угол поворота стержня A вокруг неподвижной шарнирной точки, ω_{12} — угловая скорость стержня A ,

V_{B2X} — составляющая скорости центра масс стержня B вдоль оси x .

V_{B2Y} — составляющая скорости центра масс стержня B вдоль оси y и ω_{23} — угловая скорость стержня B .

Кононюк А.Е. Графы

Неизменный выбор скоростей центров масс дает наиболее систематический подход к моделированию; выбор других скоростей, основанный на интуиции и опыте, иногда может привести к более простым графам связей.

Первый шаг состоит в установлении соотношения между q_C и q_K . Полезно ввести дополнительную кинематическую переменную Θ_2 , рассматриваемую сначала в качестве независимой с последующим наложением связи. Исходный граф связей с геометрическими переменными показан на рис.10, а.

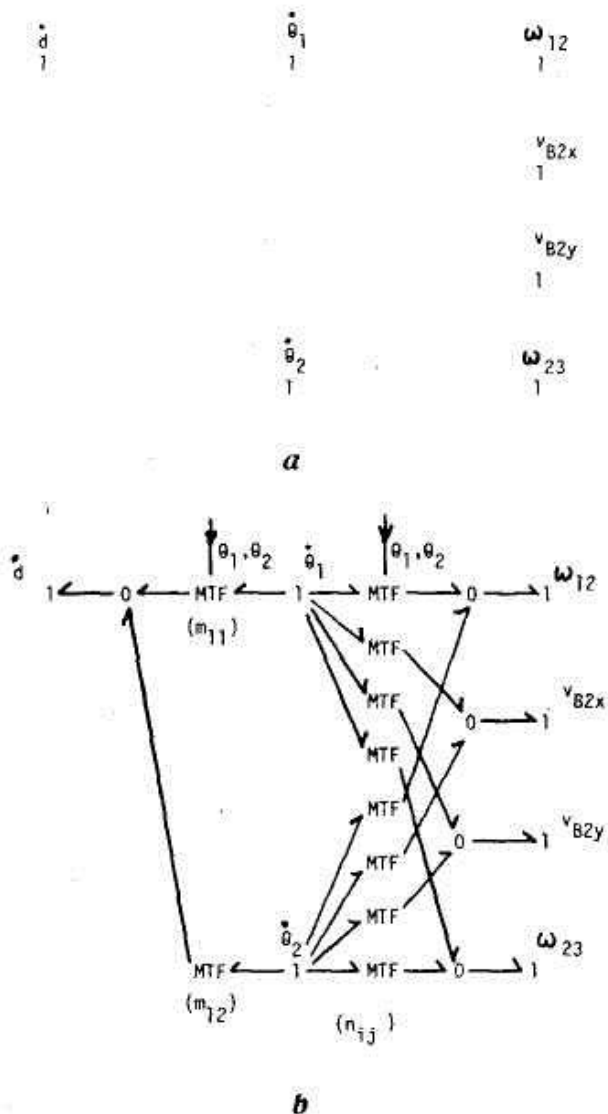


Рис.10. Структура узлов, представляющая связи в кривошипно-шатунном механизме.

a — выявленные ключевые геометрические переменные; *b* — подстановка преобразований скоростей.

Кононюк А.Е. Графы

Далее устанавливается соотношение между q_c и q_k и выводится соответствующее преобразование скоростей:

$$d = -a \cos \theta_1 - b \cos \theta_2 + L;$$

$$\dot{d} = a \sin \theta_1 \cdot \dot{\theta}_1 + b \sin \theta_2 \cdot \dot{\theta}_2.$$

Следовательно,

$$T_{CK}(\theta_1, \theta_2) = [a \sin \theta_1, b \sin \theta_2].$$

Эту матрицу для удобства покажем на рис.10, b в виде

$$T_{CK} = [m_{11}, m_{12}].$$

Компоненты скоростей инерционных элементов можно записать, анализируя систему

$$\omega_{12} = 1 \cdot \dot{\theta}_1;$$

$$V_{B2X} = -a \sin \theta_1 \cdot \dot{\theta}_1 - \frac{b}{2} \sin \theta_2 \cdot \dot{\theta}_2$$

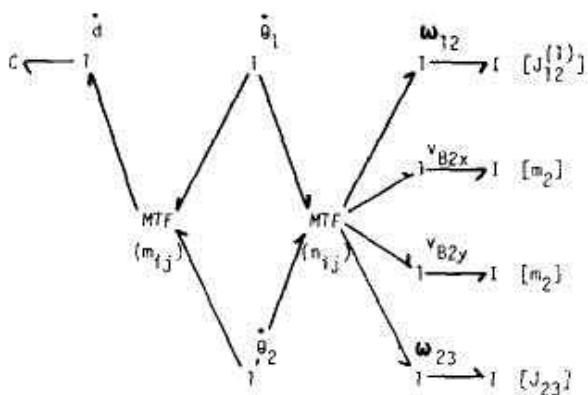
$$V_{B2Y} = a \cos \theta_1 \cdot \dot{\theta}_1 - \frac{b}{2} \cos \theta_2 \cdot \dot{\theta}_2$$

$$\omega_{23} = 1 \cdot \dot{\theta}_2.$$

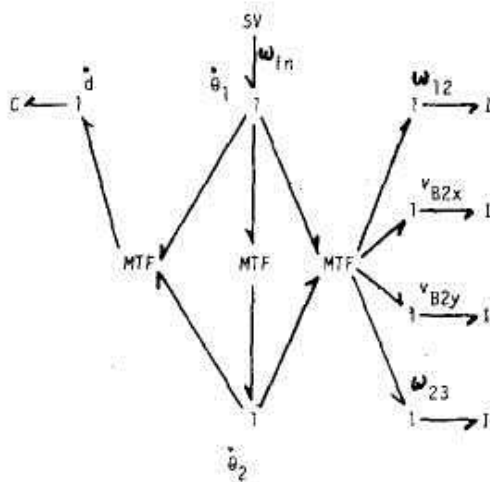
Тогда преобразование будет иметь вид

$$T_{IK} = \begin{bmatrix} 1 & 0 \\ -a \sin \theta_1 & -\frac{b}{2} \sin \theta_2 \\ a \cos \theta_1 & -\frac{b}{2} \cos \theta_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \\ n_{31} & n_{32} \\ n_{41} & n_{42} \end{bmatrix}$$

Структуры, представляющие эти преобразования, показаны на рис.10, b , где элементы m_{ij} представлены, а элементы n_{ij} подразумеваются. Упругие характеристики добавлены к графу связей и инерционные свойства звеньев учтены в структуре, что позволяет получить почти полную модель, изображенную на рис.11, a .



a



b

Рис.11. Модель графа связей для кривошипно-шатунного механизма. *a* — модель графа связей с присоединенными С и I-полями; *b*—модель графа связей с учетом ограничения на движение ползуна и входной скорости.

Кононюк А.Е. Графы

Для удобства явная структура преобразования сведена к компактной форме. Обозначения инерции указывают на различный их характер: здесь $J^{(1)}_{12}$ — момент инерции звена A относительно шарнира 1 и J_{23} — момент инерции стержня B относительно центра масс.

Теперь наложим связь, обусловленную прямолинейным движением ползуна, на углы Θ_1 и Θ_2 , сохранив за Θ_1 роль независимой переменной. Так как

$$a \sin \theta_1 = b \sin \theta_2,$$

или

$$\sin \theta_2 = \frac{a}{b} \sin \theta_1,$$

то

$$\cos \theta_2 \cdot \dot{\theta}_2 = \frac{a}{b} \cos \theta_1 \cdot \dot{\theta}_1,$$

и

$$\cos \theta_2 = \left[1 - \left(\frac{a}{b} \right)^2 \sin^2 \theta_1 \right]^{1/2},$$

поэтому

$$\dot{\theta}_2 = \frac{\frac{a}{b} \cos \theta_1}{\left[1 - \left(\frac{a}{b} \right)^2 \sin^2 \theta_1 \right]^{1/2}} \cdot \dot{\theta}_1,$$

или

$$\dot{\theta}_2 = t_{21}(\theta_1) \cdot \dot{\theta}_1.$$

Это ограничение может быть введено в структуру узлов между Θ'_1 и Θ'_2 , как показано на рис.11, b . Итак, может существовать лишь одно независимое движение (например, Θ'_1 или ω_{12} , или ω_{23} и т. д.).

Если также предположить, что угловая скорость рычага A задана, то в 1-узле для Θ'_1 следует добавить источник скорости, как показано на рис.11, b .

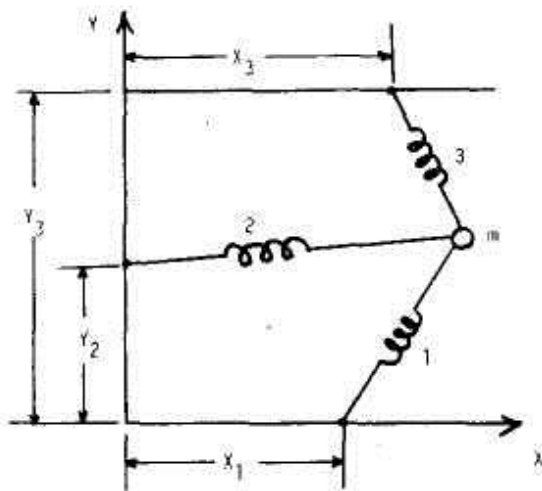
Кононюк А.Е. Графы

Фактически это допущение позволяет векторно определить все остальные скорости по заданной входной скорости, что эквивалентно обычному кинематическому анализу механизма.

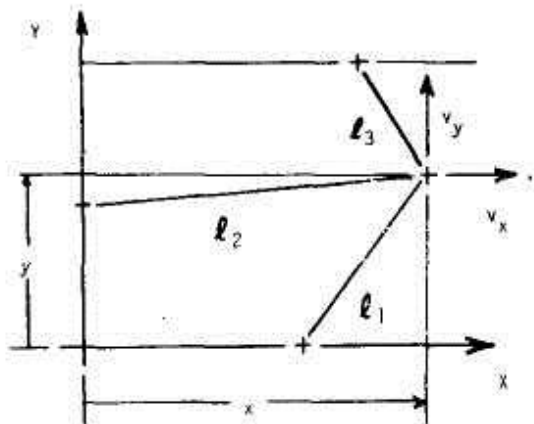
Системы, содержащие статически связанные податливые элементы

Методика построения моделей конечных графов связей для систем, содержащих статически связанные податливые элементы, по существу аналогична рассмотренной выше. Единственное различие состоит в том, что геометрическое преобразование со стороны С-поля графа может стать более сложным.

В качестве иллюстративного примера рассмотрим точечную массу, подвешенную на трех пружинах, как показано на рис.12, *a*. Ключевые геометрические переменные представлены на рис.12, *b*.



a



b

Рис.12. Колебательное звено, состоящее из массы и статически связанных пружин,
a — схема колебательного звена; *b* — геометрические переменные.

Конюк А.Е. Графы

В этом случае преобразование скоростей T_{IK} является простым, чего нельзя сказать о преобразовании T_{CK} . Три ключевых вектора таковы:

$$q_C = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}; q_K = \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{и} \quad V_I = \begin{bmatrix} v_x \\ v_y \end{bmatrix}.$$

Заметим, что $v_x = x'$ и $v_y = y'$, поэтому

$$T_{IK} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

С другой стороны,

$$l_1 = [(x - X_1)^2 + (y)^2]^{1/2},$$

$$l_2 = [(x)^2 + (y - Y_2)^2]^{1/2}$$

и

$$l_3 = [(x - X_3)^2 + (Y_3 - y)^2]^{1/2}.$$

Дифференцируя каждое уравнение по времени, найдем, что

$$\dot{l}_1 = m_{11}(x, y) \cdot \dot{x} + m_{12}(x, y) \cdot \dot{y};$$

$$\dot{l}_2 = m_{21} \cdot \dot{x} + m_{22} \cdot \dot{y}$$

и

$$\dot{l}_3 = m_{31} \cdot \dot{x} + m_{32} \cdot \dot{y},$$

где

$$m_{i,j} = \partial l_i / \partial q_{Kj}.$$

Вся наиболее трудная работа заключается в получении геометрических преобразований скоростей.

Модель графа связей, представляющая полную систему, показана на рис.13.

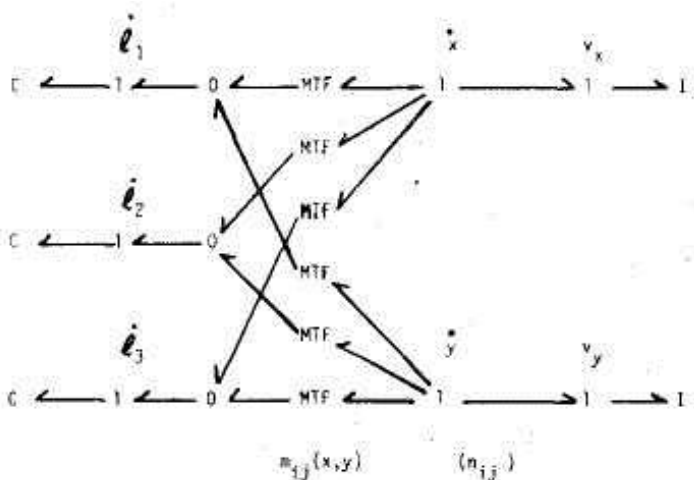


Рис.13. Модель графа связей колебательного звена из массы и статически связанных пружин (n_{ij} подразумеваются).

В силу принятой при построении модели методики статическая связь между l_1 , l_2 и l_3 , не сформулирована в явной форме, но учтена надлежащим использованием x и y .

Симметричная многосвязная модель механических систем

После рассмотрения нескольких конкретных задач по одной и той же методике можно построить модель графа связей для механической системы, симметричную относительно столбца кинематических скоростей, как показано на рис.14.

Кононюк А.Е. Графы

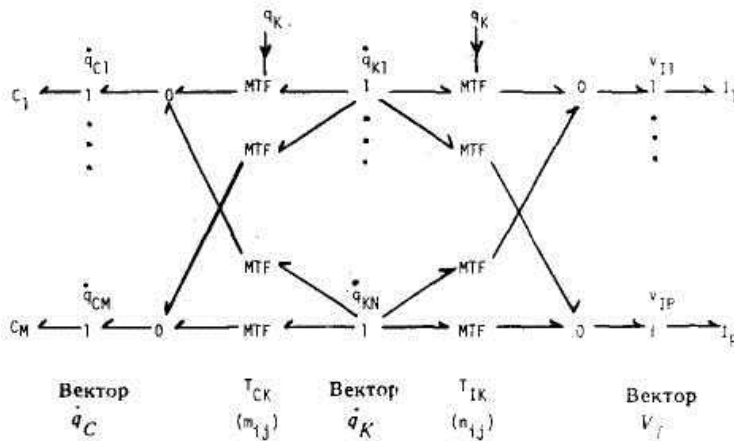


Рис.14. Симметричная модель графа связей многополюсной механической системы.

Если такая модель установлена (по методике, описанной здесь, или по любой другой методике), то в случае необходимости легко ввести диссипационные эффекты, источники сил и различные геометрические связи.

Необходимой информацией для построения является определение трех геометрических векторов q_C , q_K и V_I , и вывод преобразований скоростей $T_{СК}$ и $T_{ТК}$. Элементы С- и I-полей могут быть добавлены затем к основной структуре связей.

Выделим по крайней мере три основных достоинства разработанной здесь методики построения модели объекта с использованием графов связей.

Прежде всего структурное представление сложных проблем механики производится в сжатой и единообразной форме, легко используемой для обмена информацией между исследователями и непосредственной передачи ее на ЭВМ через устройства печати графиков и линий.

Второе достоинство состоит в том, что исследования различных классов нелинейных систем могут основываться на структурной природе систем и их нелинейностей. Подобные исследования могут иметь небольшую отдачу в тех областях, как идентификация нелинейных систем, где «чистая» структура оказывается предпочтительнее системы функций с бесконечным числом членов.

Третье достоинство заключается в том, что представляется возможность прямого моделирования задач нелинейной механики (как и других нелинейных систем) на ЭВМ.

1.4. Метод моделирования гидродинамических систем графами связей

Ниже будет рассмотрен предложенный Д. Кэрном метод, основанный на применении графов связей для моделирования гидравлических систем.

Введение

Основы прикладной механики заложены в ньютоновской механике, и, следовательно, такие понятия, как мощность, энергия, импульс и т.д., которые унифицируют изучение физических систем с помощью графов связей, должны быть применимы к гидравлическим системам, а также к механическим системам твердых тел. По крайней мере внешне, можно установить аналогию между усилием, расходом, перемещением и импульсом для гидравлических систем и соответствующими переменными, характеризующими механические системы твердых тел, как это показано в приведенной ниже таблице.

Обозначение переменных в гидромеханике и механике твердого тела

	Гидромеханика	Механика твердого тела
Усилие	Давление $P \equiv \dot{p}_p$	Сила $f \equiv \dot{p}$
Поток	Объемный расход $Q = \dot{V}$	Скорость $v = \dot{x}$
Импульс	Импульс давления p_p	Импульс p
Перемещение	Объем V	Перемещение x
Мощность	$P(t) \cdot Q(t)$	$f(t) \cdot v(t)$
Энергия	$\int P dV, \int Q dp_p$	$\int f dx, \int v dp$

Несмотря на явные аналогии между переменными, обычно используемыми в гидромеханике и механике твердого тела, до тех пор, пока не будет изучен ряд систем, нельзя считать очевидным, что гидравлические системы можно успешно представлять с помощью таких сосредоточенных элементов, которые используются в механике твердого тела. В большинстве работ по гидромеханике имеют дело с задачами, в которых необходимо определить двух- или трехмерное поле потока. Такие задачи успешно описываются с помощью

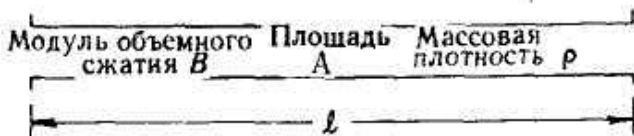
Кононюк А.Е. Графы

дифференциальных уравнений в частных производных. Если неизвестно аналитическое решение задачи, можно обратиться к методам конечных разностей или конечных элементов, при использовании которых сплошную среду представляют с помощью большого числа сходных сосредоточенных элементов. Хотя графы связей могут быть построены для «сосредоточенных микроэлементов», рассматриваемых с помощью дифференциальных уравнений в частных производных или их конечных приближений, однако такое представление обычно дает небольшие преимущества. Все сосредоточенные микроэлементы очень сходны между собой и взаимодействуют друг с другом стандартным образом.

В данном разделе основное внимание будет уделено переходу к сосредоточенным параметрам более крупного масштаба, обычно применяемому в том случае, когда гидравлическая система составляет небольшую часть более крупной системы и поэтому практически не может быть описана достаточно подробно. Типичные примеры таких случаев встречаются, когда гидравлические или пневматические элементы являются частями системы управления. В последующем изложении главным образом будут рассматриваться внутренние течения и взаимодействие жидкости с твердыми телами, осуществляемое под влиянием сил, перемещений, давлений и объемных расходов.

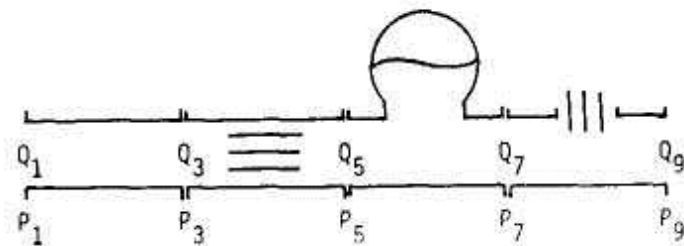
В качестве простого примера того типа приближений, который часто используется при анализе гидравлических систем, рассмотрим участок трубопровода, изображенного на рис.1.

Кононюк А.Е. Графы

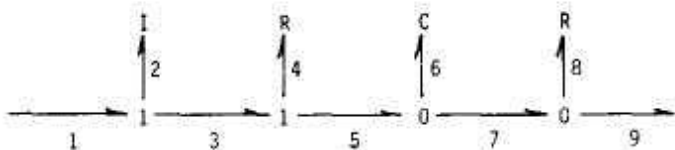


Длина линии с жидкостью, характеризующейся свойствами инерции и сжимаемости, потерями на трение и утечками

a. Схема линии с учетом некоторых параметров



b. Схематическое изображение



c. Граф связей

Рис.1. Представление гидравлической линии с помощью сосредоточенных параметров.

С помощью модели, которая в линейном случае аналогична моделям электрической линии передачи, пытаются отобразить множество явлений, которые присущи реальным трубопроводам. Такая модель возможна для коротких отрезков трубопровода и для линейных элементов, однако она часто используется при большой длине l и для нелинейных элементов. Прежде всего очевидно, что эта модель является одномерной. Объемный расход Q через поперечное сечение и давление P , безусловно, должны рассматриваться как величины,

Кононюк А.Е. Графы

усредненные в поле скоростей и давлений, которые изменяются в поперечном сечении. Поскольку профиль скоростей может меняться в широких пределах при нестационарных условиях, ясно, что переменные P и Q и параметры элементов $-R$, $-C$ и $-I$, изображенных на рис.1 не могут быть точно определены, если характер потока через поперечное сечение заранее неизвестен. Тем не менее при разработке систем редко бывает уверенность в том, как будут вести себя потоки в различных частях системы, поэтому необходимо произвести по крайней мере предварительную оценку параметров системы, прежде чем оценить динамику потока.

Рассмотрим сначала задачу оценки инерции жидкости в трубопроводе. Простейший метод выполнения коэффициента инерции состоит в следующем (рис.1, *c*). Масса жидкости в трубопроводе равна ρAl , сила, сообщающая жидкости ускорение, равна $A(P_1 - P_3)$, скорость жидкости равна Q_2/A . Следовательно

$$\rho Al \frac{d}{dt} Q_2/A = A(P_1 - P_3) \text{ или } Q_2 = \frac{A}{\rho l} \dots \quad (1)$$

где p_{p_2} — импульс давления связи 2 (или интеграл по времени от P_2).

Этот результат показывает, что при использовании переменных P и Q коэффициент инерции, выраженный как «масса», равен $\rho l/A$. Тот факт, что инерция труб с малым поперечным сечением больше, чем инерция труб с большим поперечным сечением, кажется неожиданным.

В связи с приведенным выше простым решением возникает несколько вопросов. Во-первых, в соответствии с предложением об одномерности считалось, что жидкость движется в трубе как твердое тело. Пока не будет известно, как профиль скоростей меняется в пространстве и времени, трудно вносить поправки в это предположение. Тип потоков, рассматриваемых нами, часто называют «квазиодномерными потоками», и при установившемся течении, когда хорошо развитый профиль скорости известен в каждом сечении, можно определить Q/A как *среднюю* скорость. Для последующих вычислений нам понадобится среднее значение квадрата скорости в сечении, поэтому примем эту величину равной $\beta Q^2/A^2$, где поправочный коэффициент $\beta \geq 1$ можно вычислить, если известен профиль скорости. Коэффициент β равен единице только для равномерного распределения скорости, которое может иметь место лишь в безвихревом потоке идеальной жидкости. В случае нестационарных условий часто очень трудно оценить распределение скорости, поэтому в дальнейшем мы будем считать распределение скорости равномерным даже в тех случаях,

Кононюк А.Е. Графы

когда оно строго не является таковым. Точность некоторых результатов можно несколько улучшить введением коэффициентов, подобных β , хотя оценка величин этих коэффициентов при отсутствии экспериментальных данных может быть затруднительной. Во-вторых, в случае отказа от предположения о несжимаемости жидкости соответствующее значение плотности ρ остается неустановленным. Если бы $l=dx \rightarrow 0$, то $\rho(x, t)$ представляло бы текущее значение ρ в сечении x , но в случае конечного l наличие емкости, изображенной на рис.1, несовместимо с допущением о постоянстве ρ . Если ρ изменяется, то, разумеется, два концевых сечения элемента жидкости, вообще говоря, не могут двигаться с одинаковой скоростью. В дальнейшем будет дана более сложная постановка данной задачи. А пока просто считается, что для гидравлических систем, в которых изменения плотности малы, достаточно использовать среднюю плотность ρ .

Что действительно имеет погрешность в этом выводе, так это представление элемента жидкости в виде твердого тела, даже если ясно, что при этом подразумевается контрольный объем, включающий в себя длину трубы, по которой движется масса и импульс. Как будет показано, решение оказывается наиболее корректным, если два концевых сечения трубы имеют одинаковую площадь, так как тогда исчезают члены, определяющие поток импульса через эти сечения. Такое простое решение нельзя непосредственно обобщать на случай трубопровода с переменной площадью сечения.

Элемент R_4 на рис.1 представляет потери давления, помимо тех, которые связаны с ускорением жидкости. Соотношение между P_4 и Q_4 можно было бы легко определить как нелинейное соотношение, если бы была возможность использовать опытные данные для течений в трубопроводах. Но почти все данные о коэффициентах трения соответствуют полностью развитому установившемуся потоку, и их нельзя применять ни для каких нестационарных условий, за исключением тех, которые протекают очень медленно. По этой причине необходимо быть готовым к эксперименту по определению закона трения, при котором реакция модели будет достаточно хорошо согласовываться с результатами экспериментальных данных.

Следующий элемент, изображенный на рис.1, предназначен для моделирования сжимаемости одной только жидкости или совместно с податливостью стенок трубы. Расход Q_6 , который равен разности расходов Q_5 и Q_7 , представляет потери в потоке на участке между концами трубы. Давление P_6 (которое совпадает по величине с P_5 и P_7) можно вычислить путем интегрирования Q_6 при условии, что можно определить модуль объемного сжатия V . Тогда

Кононюк А.Е. Графы

$$P_6(t) = P_6(0) + \frac{B}{V_0} \int_0^t Q_6 dt = P_6(0) + \frac{B}{V_0} V_6(t) \quad , \quad (2)$$

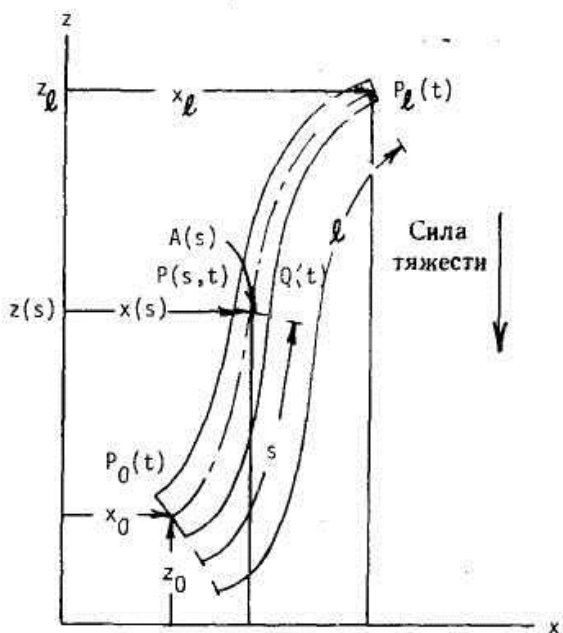
где V_0 — объем жидкости в трубе в момент времени $t=0$ при давлении $P_6(0)$. Может показаться, что достаточно просто заменить линейное соотношение (2) некоторым нелинейным соотношением, связывающим P_6 и V_6 , с тем, чтобы смоделировать эффекты сжимаемости газа, и тогда та же модель будет пригодной в случае больших изменений плотности, однако дело обстоит намного сложнее. Как показано в последующем изложении, если изменения плотности значительны, то следует учитывать термодинамические характеристики процессов.

Наконец, R_8 представляет потери потока в сечении трубы, обусловленные утечками. Ясно, что если P является абсолютным, а не манометрическим давлением, то R_8 реагирует на разность между P_8 и давлением снаружи трубы, а не на само давление P_8 . Это требует, чтобы R_8 было связано с 1-узлом, где вычисляется разность между внутренним и внешним давлениями, и накладывает условие равенства между потоком, вытекающим из трубы, и потоком, поступающим во внешнее пространство.

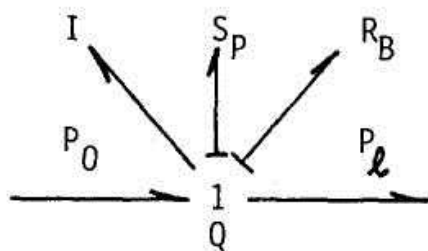
Из сказанного выше ясно, что при обосновании процесса перехода к сосредоточенным параметрам встречаются существенные трудности. Многие из них исчезают, если необходима лишь линеаризованная сосредоточенная модель, поскольку в этом случае можно пренебречь малыми изменениями параметров, которые имеют конечные средние значения. Для того чтобы ясно представить природу наиболее часто используемых гидродинамических моделей, необходимо рассмотреть дифференциальные уравнения гидромеханики в частных производных и нелинейные эффекты.

Одномерный поток несжимаемой жидкости в движущихся контрольных объемах

Для того чтобы выяснить связь между типовыми методами гидромеханики и методами представления систем с сосредоточенными параметрами, использующими элементы теории графов связей, дадим сначала вывод уравнения Бернулли. Рассмотрим задачу, представленную на рис.2, а.



а. Схема системы



б. Граф связей

Рис. 2. Движение несжимаемой жидкости в криволинейном канале.

Пусть s соответствует расстоянию вдоль центральной линии криволинейного трубопровода длиной l . Тогда $A(s)$ — площадь поперечного сечения трубопровода, $x(s)$ и $z(s)$ описывают соответственно горизонтальное и вертикальное положение центральной линии, $v(s, t)$ — средняя скорость жидкости в точке s в момент времени t и $P(s, t)$ — давление.

Кононюк А.Е. Графы

В соответствии с законом Ньютона можно написать

$$\rho \frac{Dv}{Dt} = \rho \frac{\partial v}{\partial t} + \rho v \frac{\partial v}{\partial s} = - \frac{\partial P}{\partial s} - \rho g \frac{dz}{ds} \quad (3)$$

Заметим, что в силу несжимаемости жидкости объемный расход $Q(t)$ не зависит от s , а v определяется следующим соотношением:

$$v(s, t) = Q(t)/A(s) \quad (4)$$

Подставляя (4) в уравнение (3), получим

$$\rho \left(\frac{\dot{Q}(t)}{A(s)} + \frac{Q^2(t)}{A(s)} \frac{\partial 1/A(s)}{\partial s} \right) = - \frac{\partial P}{\partial s} - \rho g \frac{dz}{ds}, \quad (5)$$

Интегрируя (5) по s от $s=0$ до $s=l$, будем иметь

$$I\dot{Q} + \frac{\rho Q^2}{2} \left(\frac{1}{A_l^2} - \frac{1}{A_0^2} \right) = P_0(t) - P_l(t) - \rho g(z_l - z_0), \quad (6)$$

где

$$I = \rho \int_0^l ds/A(s) \quad (7)$$

и вместо члена Q^2/A^2 можно подставить $\beta Q^2/A^2$ в соответствии с приведенными выше рассуждениями.

Следует отметить, что уравнение Бернулли (6) можно точно представить с помощью сосредоточенных элементов, изображенных на рис.2, *b*. Линейный инерционный коэффициент, определенный с помощью (7), сводится к виду, который был найден не строгим методом, используемым при выводе уравнения (1), при условии, что $A(s)=\text{const}$, однако в уравнении (6) появился ряд новых членов, которых не было в (1). Наличие гравитационного члена, который представлен на рис.2, *b* с помощью источника постоянного давления, легко понять, в то время как член, содержащий Q^2 , требует пояснений.

Прежде всего, если $A(s)=\text{const}$, то $A_l=A_0$ и член с Q^2 исчезает. Это свидетельствует о том, что уравнение (1) по существу корректно в случае постоянного поперечного сечения. Однако при выводе уравнения (1) поток жидкости через концевые сечения отрезка трубопровода не был рассмотрен надлежащим образом. Применяя закон Ньютона в его простейшей форме, необходимо «мысленно следовать» за потоком жидкости, т. е. необходимо использовать описание Лагранжа. Тем не менее практически стремятся

Кононюк А.Е. Графы

рассматривать трубопровод как некий контрольный объем, через который течет жидкость, т. е. пытаются применить описание Эйлера. Член уравнения (6), содержащий Q^2 можно рассматривать как поправочный член на динамическое давление, который был опущен при выводе уравнения (1).

Другой способ интерпретации уравнения (6) основан на том факте, что поток мощности $P(t) \cdot Q(t)$ не является общим потоком энергии через стационарную точку трубопровода. Члены вида $\rho Q^2 / 2A^2$ определяют динамическое давление, обусловленное кинетической энергией жидкости. В рассматриваемой системе P и Q содержат всю информацию, требуемую для вычисления динамического давления, поскольку скорость равна Q/A . Член уравнения (6) с Q^2 можно рассматривать как сопротивление, представленное на рис. 2, б, так как он определяет связь между Q и давлением (динамическим давлением). Если начертить характеристики сопротивления, как это изображено на рис.3, видно, что при рассмотрении только мощности PQ динамическое давление может быть преобразовано в статическое и обратно, как это происходит в соплах и диффузорах.

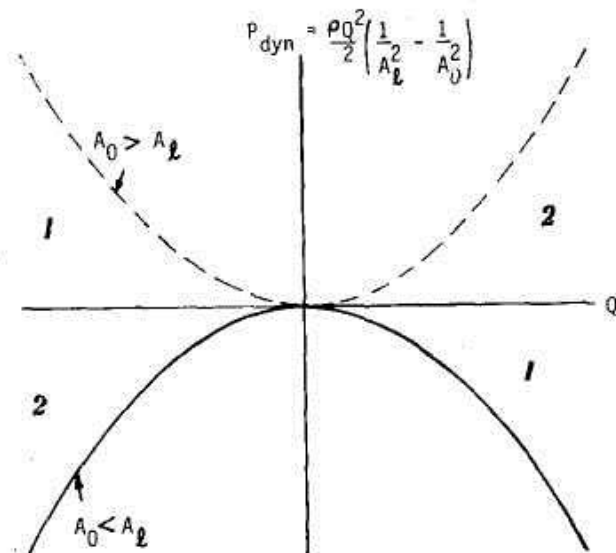


Рис.3. Характеристики сопротивления, соответствующего динамическому давлению. 1 — восстановление статической мощности; 2 — потери статической мощности.

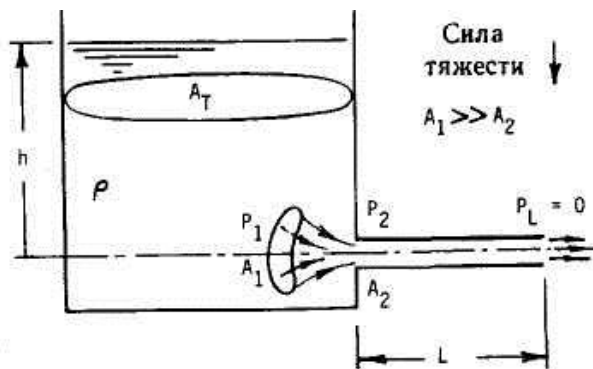
Кононюк А.Е. Графы

Сопротивление, удовлетворяющее уравнению Бернулли, является необычным в силу двух обстоятельств. Во-первых, не зависимо от того, выполняется условие $A_0 > A_1$ или условие $A_0 < A_1$ существуют режимы работы, при которых сопротивление является источником мощности. Физически это означает, что динамическое давление частично преобразуется в статическое давление и в силу этого в кажущийся поток мощности $P-Q$. Во-вторых, хотя давление в выражении для сопротивления определяется однозначно для любого расхода, однако если задаться давлением, то нельзя будет определить однозначно расход. Таким образом, для этого сопротивления существует сильная причинная обусловленность.

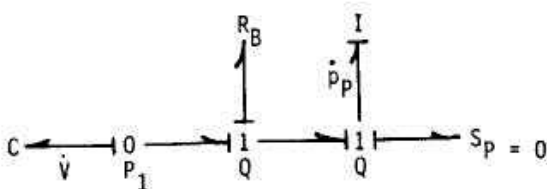
Последнее свойство легко объяснить, заметив, что в основе наших рассуждений лежало допущение о заполнении жидкостью трубопровода и, следовательно, $v=Q/A$. Для некоторых условий, определяющих давление в концевых сечениях трубопровода, это допущение не соответствует действительности — в трубопроводе может образоваться струйное течение. Другая возможность состоит в том, что выведенное уравнение справедливо только для одного направления движения. При исследовании, например, истечения воды в атмосферу через насадок нет никакого смысла рассматривать обратное течение, так как насадок может заполниться воздухом и уравнения уже не будут справедливы в этом случае. Достаточно сказать, что при любом применении уравнения Бернулли, включая представление в виде графов связей, приведенное на рис.2, следует быть очень внимательным, чтобы использовать соответствующую ветвь графика сопротивления и избежать тем самым бессмысленных результатов. Такие трудности, связанные с сопротивлением, редко встречаются в физических системах другого типа.

В качестве иллюстрации использования представления с помощью графов связей рассмотрим простую классическую задачу оценки времени, необходимого для истечения воды из сосуда через трубу (рис.4).

Кононюк А.Е. Графы



а. Схема системы



б. Граф связей

Рис.4. Задача истечения жидкости из резервуара.

Для простоты не учитываются потери на трение, однако сопротивление Бернулли можно рассматривать как показатель потерь кинетической энергии жидкости, вытекающей из системы. Емкость сосуда равна $A_T/\rho g$ где A_T — площадь сосуда, гораздо большая по сравнению с площадью трубы. Как и при классическом решении, представим, что жидкость входит в трубу при давлении P_1 через некоторое сечение площадью A_1 (большей, чем A_2) практически с нулевой скоростью. Сопротивление Бернулли в этом случае определяет давление $\rho Q^2/2A_2^2$, где Q связано с импульсом давления соотношением, аналогичным по форме с уравнением (1). Используя обозначения на рис.4 и замечая с помощью графов связей, что требуются две переменные состояния $V(t)$ и $p_p(t)$, находим

$$\dot{V} = - \frac{A_2}{\rho L} p_p ; \tag{8}$$

Конюк А.Е. Графы

$$\dot{p}_p = \frac{-\rho}{2A_2^2} \left(\frac{A_2}{\rho L} p_p \right)^2 + \frac{\rho g V}{A_T} \quad (9)$$

или

$$-\frac{L}{A_2} \ddot{V} = \frac{-\rho}{2A_2^2} (\dot{V})^2 + \frac{\rho g V}{A_T}, \quad (10)$$

пока $V \geq 0$. Заметим, что, несмотря на отсутствие статического потока мощности, направленного к источнику нулевого давления, представляющему собой атмосферу, существует поток, из которого можно извлечь мощность с помощью диффузора.

Чтобы проиллюстрировать сложность задач, решаемых с использованием понятия контрольного объема, и показать возможности графов связей при исследовании сложных систем, рассмотрим систему, изображенную на рис.5, а.

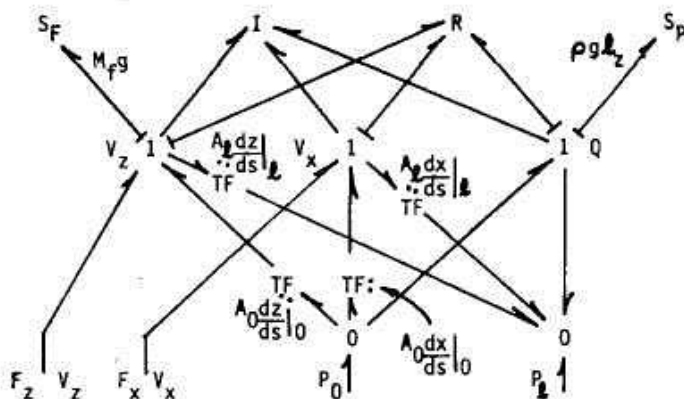
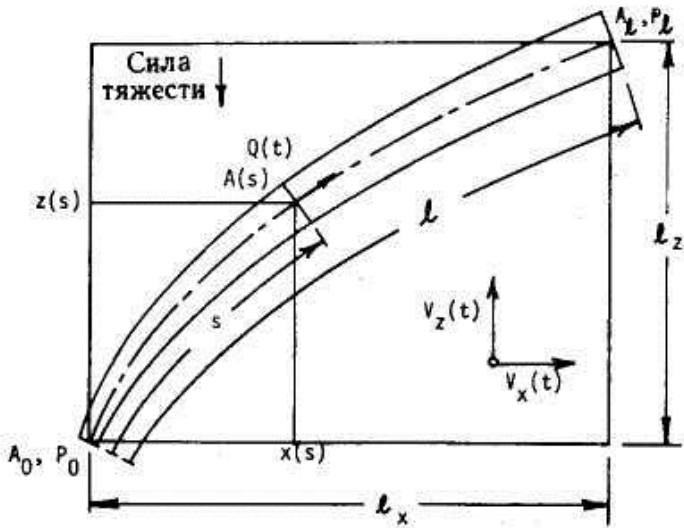


Рис.5. Одномерный поток несжимаемой жидкости в движущемся контрольном объеме.

Конюк А.Е. Графы

Различие между рис.2 и рис.5 состоит в том, что в последнем случае трубопровод движется в плоскости x, z (без вращения) со скоростями $v_x(t), v_z(t)$ под действием внешних сил $F_{x \text{ ext}}(t)$ и $F_{z \text{ ext}}(t)$, равных

$$F_{x \text{ ext}} = P_0 A_0 \left. \frac{dx}{ds} \right|_0 - P_l A_l \left. \frac{dx}{ds} \right|_l + F_x(t); \quad (11)$$

$$F_{z \text{ ext}} = P_0 A_0 \left. \frac{dz}{ds} \right|_0 - P_l A_l \left. \frac{dz}{ds} \right|_l + F_z(t) - M_f g, \quad (12)$$

где $F_x(t)$ и $F_z(t)$ — любые силы, действующие на контрольный объем в дополнение к силам давления и гравитационным силам, а общая масса жидкости M_f в этом объеме равна

$$M_f \equiv \int_0^l \rho A(s) ds \quad (13)$$

Пусть $v_a(s, t)$ обозначает абсолютную скорость жидкости в направлении оси трубы

$$v_a(s, t) = V_x \frac{dx}{ds} + V_z \frac{dz}{ds} + v_{rel}(s, t), \quad (14)$$

где $v_{rel}(s, t)$ — скорость жидкости относительно трубопровода, равная

$$v_{rel}(s, t) = Q(t)/A(s) \quad (15)$$

Тогда абсолютное ускорение жидкости в направлении оси трубы будет равно

$$\frac{Dv_a}{Dt} = \frac{\partial v_a}{\partial t} + v_{rel} \frac{\partial v_{rel}}{\partial s} \quad (16)$$

Основное уравнение, аналогичное уравнению (3) для неподвижного контрольного объема, примет вид

$$\begin{aligned} \rho \frac{Dv_a}{Dt} &= \rho \left(\dot{V}_x \frac{dx}{ds} + \dot{V}_z \frac{dz}{ds} + \frac{\dot{Q}}{A} + \frac{Q^2}{A} \frac{d}{ds} \left(\frac{1}{A} \right) \right) = \\ &= \frac{\partial P}{\partial s} - \rho g \frac{dz}{ds}. \end{aligned} \quad (17)$$

Интегрируя (17) от $s=0$ до $s=l$, получим

Кононюк А.Е. Графы

$$\begin{aligned} \rho l_x \dot{V}_x + \rho l_z \dot{V}_z + I \dot{Q} + \frac{\rho Q^2}{2} \left(\frac{1}{A_1^2} - \frac{1}{A_0^2} \right) = \\ = P_0 - P_1 - \rho g l_x \quad , \quad (18) \end{aligned}$$

где I определено, как и раньше, выражением (7), а l_x и l_z показаны на рис.5.

Уравнение (18) по существу характеризует связь между давлениями на концах трубопровода и ускорениями контрольного объема и жидкости в нем. Для того чтобы связать внешние силы, действующие на контрольный объем, с ускорениями, нужно написать закон сохранения импульса

$$\frac{d[\bar{p}]_{vol}}{dt} = F_{ext} + \left. \frac{d\bar{p}}{dt} \right|_{in} - \left. \frac{d\bar{p}}{dt} \right|_{out} \quad , \quad (19)$$

где $[\bar{p}]_{vol}$ — вектор количества движения жидкости, содержащейся в объеме, а члены в правой части появляются в том случае, когда закон Ньютона для фиксированного количества вещества преобразуется к виду, используемому при рассмотрении переноса массы и импульса через контрольный объем.

Составляющая по оси x для левой части уравнения (19) легко вычисляется

$$\begin{aligned} \frac{d}{dt} [p_x]_{vol} &= \frac{\partial}{\partial t} \int_0^l \rho A v_x ds = \frac{\partial}{\partial t} \int_0^l \rho A \left(V_x + \frac{Q}{A} \frac{dx}{ds} \right) ds = \\ &= V_x \int_0^l \rho A ds + \rho Q l_x = M_f \dot{V}_x + \rho l_x \dot{Q} \quad , \quad (20) \end{aligned}$$

где M_f — масса жидкости, определенная в (13), а l_x приведено на рис.5. Аналогичные результаты получаются и для составляющей по оси z :

$$\frac{d}{dt} [p_z]_{vol} = M_f \dot{V}_z + \rho l_z \dot{Q} \quad . \quad (21)$$

Проекция на ось x потоков импульса в уравнении (19) имеют вид

$$\left. \frac{dp_x}{dt} \right|_{in} = \rho Q \left(V_x + \frac{Q}{A_0} \frac{dx}{ds} \Big|_0 \right);$$

$$\left. \frac{dp_x}{dt} \right|_{out} = \rho Q \left(V_x + \frac{Q}{A_1} \frac{dx}{ds} \Big|_1 \right), \quad (22)$$

а проекции на ось

$$\left. \frac{dp_z}{dt} \right|_{in} = \rho Q \left(V_z + \frac{Q}{A_0} \frac{dz}{ds} \Big|_0 \right);$$

$$\left. \frac{dp_z}{dt} \right|_{out} = \rho Q \left(V_z + \frac{Q}{A_1} \frac{dz}{ds} \Big|_1 \right). \quad (23)$$

Подставляя выражения(11), (12) и (20)-(23) в уравнение (19), получим два уравнения

$$M_f \dot{V}_x + \rho l_x \dot{Q} = P_0 A_0 \frac{dx}{ds} \Big|_0 - P_1 A_1 \frac{dx}{ds} \Big|_1 +$$

$$+ \rho Q^2 \left(\frac{1}{A_0} \frac{dx}{ds} \Big|_0 - \frac{1}{A_1} \frac{dx}{ds} \Big|_1 \right) + F_x \quad (24)$$

$$M_f \dot{V}_z + \rho l_z \dot{Q} = P_0 A_0 \frac{dz}{ds} \Big|_0 - P_1 A_1 \frac{dz}{ds} \Big|_1 +$$

$$+ \rho Q^2 \left(\frac{1}{A_0} \frac{dz}{ds} \Big|_0 - \frac{1}{A_1} \frac{dz}{ds} \Big|_1 \right) + F_z - M_f g. \quad (25)$$

С помощью уравнений (18), (24) и (25) можно определить связь между скоростями и расходом, с одной стороны, и силами и давлениями — с другой. Однако эти уравнения не так легко интерпретировать.

Первый этап заключается в интерпретации этих уравнений в виде графов связей, для чего необходимо представить левые части уравнений в матричной форме и обозначить правые части как результирующие силы [уравнения (24) и (25)] и результирующее давление [уравнение (18)]:

$$\begin{bmatrix} M_f & 0 & \rho l_x \\ 0 & M_f & \rho l_z \\ \rho l_x & \rho l_z & I \end{bmatrix} \begin{bmatrix} \dot{V}_x \\ \dot{V}_z \\ \dot{Q} \end{bmatrix} = \begin{bmatrix} F_x \text{ net} \\ F_z \text{ net} \\ P_{\text{not}} \end{bmatrix}, \quad (26)$$

Кононюк А.Е. Графы

В уравнении (26) сформулированы основные соотношения для линейного 3-связного I -поля. Это уравнение аналогично линейной электрической цепи с взаимной индукцией и механическим системам с движущимися твердыми телами. Разумеется, если трубопровод обладает массой, то в уравнении (26) или в (24) и (25) к величине M_f следует добавить массу трубопровода. Соотношения (26) соответствуют I - полю, так как симметрия матрицы обеспечивает сохранение энергии.

Остальные члены рассматриваемой системы уравнений легко интерпретировать, если попытаться определить R -поле, чтобы учесть члены, содержащие Q^2 , которые соответствуют давлению в уравнении (18) и силам в уравнениях (24) и (25). R -поле, которое учитывает конвективный перенос энергии и импульса, имеет предпочтительную причинность, а именно, если задано Q , то силы и давления определяются однозначно, однако нельзя найти однозначное Q для заданных сил и давлений. На графе связей, приведенном на рис. 5, b , предпочтительная причинность соответствует R - полю, а другие связи изображены без указания причинных соотношений.

Граф связей дает дополнительное представление о системе уравнений. Система имеет четыре внешних входа, соответствующих силам F_x и F_z и давлениям P_0 и P_l и направляющих мощность через линейную узловую структуру к линейному 3-связному I - полю и нелинейному 3-связному R - полю специального вида, которое возникает вследствие конвекции. Дополнительные потери мощности на трение могут быть включены в R -поле. Свойство сохранения мощности узловой структуры дает дополнительную информацию, которую нельзя получить из системы уравнений. Например, объемный расход, который следует умножить на $P_0(t)$ чтобы вычислить входную мощность при $s=0$, равен не $Q(t)$, а

$$Q(t), \text{ а } Q(t) + V_z(t) A_0 \left. \frac{dz}{ds} \right|_0 + V_x(t) A_0 \left. \frac{dx}{dt} \right|_0$$

Этот «эквивалентный расход» правильно учитывает работу сил давления при движении контрольного объема и возникает, естественно, при рассмотрении двухсторонних характеристик преобразователей. В этом смысле граф связей является более полной характеристикой системы, чем система уравнений, которую по существу можно обоснованно использовать только в случае причинной связи по силе и давлению на входе. Чтобы применять причинную связь по расходу, необходимо знать расходы, сопряженные P_0 и P_l . В отличие от системы уравнений граф связей содержит такую информацию. Часто случается, что простая попытка построить граф

Кононюк А.Е. Графы

связей для системы уравнений вынуждает понять природу системы более глубоко, чем при использовании других методов. Это обусловлено тем, что многие уравнения по своей природе являются причинными, а графы связей должны быть в своей основе не причинными и, следовательно, должны содержать все возможные причинные соотношения.

Представление эффектов сжимаемости

Легко показать, что при малых изменениях плотности эффекты сжимаемости моделируются с помощью линейных конденсаторов, как это упоминалось в введении. При больших изменениях плотности, однако, лучше всего начать с изучения термодинамики чистого вещества. Затем можно применить метод макроскопической термодинамики состояния равновесия, обобщить его на случай, когда изменения происходят с конечной скоростью, но достаточно медленно, чтобы можно было применить анализ псевдоравновесия.

Вообще говоря, внутренняя энергия единицы массы жидкости u зависит от таких двух независимых свойств, как плотность ρ и энтропия S . Изменения этих величин связаны с помощью известного уравнения Гиббса:

$$du = TdS - Pd(1/\rho), \quad (27)$$

где T — термодинамическая температура, P — давление, а $1/\rho$ — удельный объем.

Из уравнения (27) получаем, что S и $1/\rho$ являются собственными переменными внутренней энергии u , т. е.

$$u = u(S, 1/\rho); \quad (28)$$

$$T = \partial u / \partial S \quad (29)$$

$$-P = \partial u / \partial (1/\rho) \quad (30)$$

Энергия u сохраняется, так как любые циклические изменения S и $1/\rho$ всегда возвращают u к ее исходному значению, определяемому выражением (28); в результате мы имеем соотношение Максвелла:

$$\partial T / \partial (1/\rho) = \partial^2 u / \partial (1/\rho) \partial S = \partial -P / \partial S \quad (31)$$

а характеристики жидкости можно определить, используя выражение (28) совместно с эквивалентным графом связей, изображенным на рис.6.

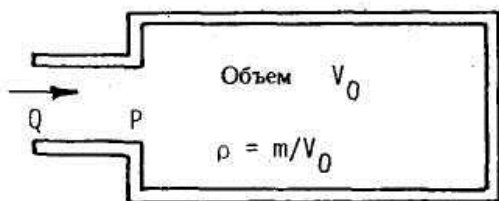
Кононюк А.Е. Графы

$$\begin{array}{ccc}
 T & & P \\
 \xrightarrow{\hspace{2cm}} & C & \xrightarrow{\hspace{2cm}} \\
 \frac{dS}{dt} & & \frac{d(1/\rho)}{dt} \\
 \\
 T = \frac{\partial u}{\partial S} & & -P = \frac{\partial u}{\partial (1/\rho)}
 \end{array}$$

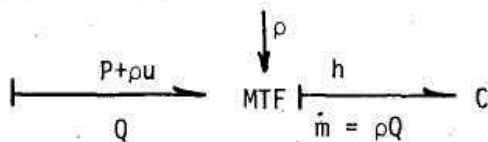
Рис.6. Представление уравнения состояния жидкости с помощью поля.

Отметим, что на рис. 6 «энергетическая схема» использована таким образом, что произведения усилий и расходов определяют мощности, как это имеет место в большинстве графов связей. Вторая возможная схема — «энтропийная» - не в полной мере совместима с другими моделями графов связей, основанными на потоке мощности и энергии. Кроме того, экстенсивные переменные S и $1/\rho$ отождествляются в графах связей с *перемещениями*, а интенсивные переменные T' и P — с *усилиями*. В этом случае консервативные свойства u представляются с помощью C -поля. Очевидно, что усилия и перемещения другой природы, как, например, химические потенциалы или мольные числа, можно присоединить к C -полю, если в жидкости, кроме сжатия и нагрева, происходят другие изменения. Аналогично, если жидкость сжимается адиабатически (и если мы пренебрегаем влиянием механического трения и вязкостью), то температурно-энтропийный входной поток является неактивным и мы сталкиваемся с изэнтропическим случаем, когда u и P зависят только от удельного объема $1/\rho$. Другим случаем, имеющим практическое значение, является *изотермический* случай, в котором сжатие происходит настолько медленно, что температура газа всегда равна температуре данного резервуара. Этот случай соответствует введению причинности по производной для TS' -входа C -поля.

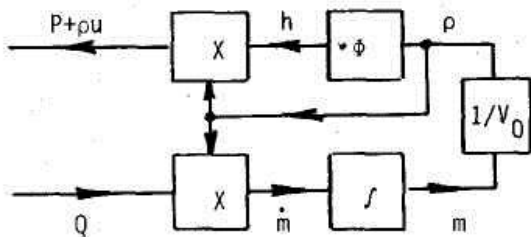
Представление с помощью C -поля позволяет непосредственно моделировать системы со сжатием, расширением и нагревом заданного количества жидкости. Однако случай, когда жидкость течет через контрольный объем, является более сложным. На рис.7 приведен пример сжатия жидкости в заданном объеме, когда жидкость может втекать и вытекать через отверстие.



а. Втекание жидкости в резервуар постоянного объема



б. Граф связей



с. Блок-схема

Рис.7. Изэнтропическое сжатие жидкости.

В этом случае общая энергия жидкости U , содержащейся в объеме V_0 , зависит от массы m в этом объеме и от внутренней энергии u на единицу массы. Изменения U происходят не только в результате работы $P \cdot Q$, совершаемой потоком, но также в результате переноса энергии. Если использовать в качестве переменной массовый расход $m' = \rho Q$ вместо Q , то для изэнтропического случая

$$dU = u dm + (P/\rho) dm \quad (32)$$

Определяя энтальпию h с помощью соотношения

Кононюк А.Е. Графы

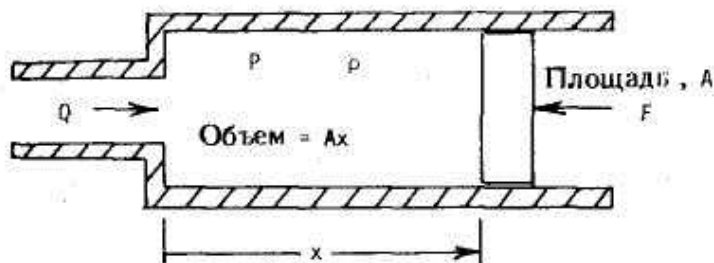
$$h = u + P/\rho \quad (33)$$

находим, что поток мощности через отверстие определяется следующими выражениями:

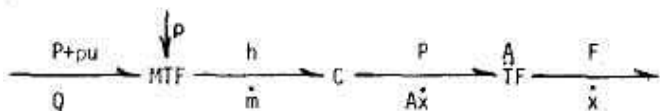
$$\begin{aligned} h \cdot \dot{m} &= u\dot{m} + (P/\rho)\dot{m} = \\ &= u\rho Q + (P/\rho)\rho Q = \\ &= (\rho u + P) \cdot Q, \end{aligned} \quad (34)$$

которые показывают, что PQ является только гидростатической мощностью. Как и в предыдущем разделе, где оказалось необходимым добавить к статическому давлению составляющую, которая характеризует динамическое давление, чтобы учесть реальный поток мощности, так и теперь необходимо добавить к P составляющую ρu , чтобы учесть конвективную внутреннюю энергию.

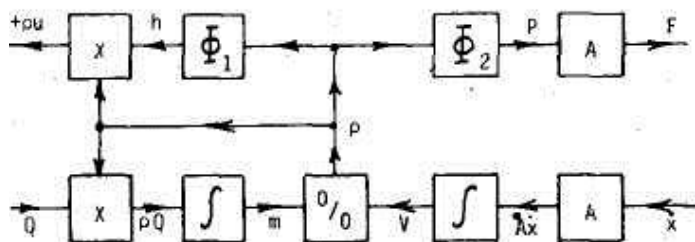
На рис.7, *b* изображен граф связей, основанный на выражениях (33) и (34), а блок-схема, соответствующая причинности по интегралу, приведена на рис.7, *c*. Заметим, что динамическое давление не было включено в рассмотрение, поэтому данная модель применима только для потоков с низкими расходами (этому случаю соответствует граф связей на рис.6). Кроме того, в изэнтропическом случае давление и внутренняя энергия зависят только от плотности, поэтому энтальпия также определяется условиями внутри сосуда. В более сложных случаях энтальпия на входе определяется условиями внутри сосуда при вытекании жидкости ($m' < 0$) и внешними условиями, если жидкость втекает в сосуд ($m' > 0$). Известно, что такого рода переключение причинности, связанное с изменением направления потока, встречается в теплообменниках, однако этот вопрос подробно не изучался с применением графов связей или других методов анализа систем.



(a) Схема системы



(b) Граф связей



(c) Блок-схема

Рис.8. Сжатие в силовом цилиндре.

На рис.8 показано, как изменяется граф связей, когда изэнтропическое сжатие осуществляется частично за счет изменения объема, а частично за счет притока и оттока жидкости, как это происходит в силовом цилиндре пневматического сервомеханизма. С-поле, изображенное на рис.8, b, имеет одну hm' -связь и одну PV' -связь. В результате полная энергия U зависит от m и V следующим образом:

Кононюк А.Е. Графы

$$h = \frac{\partial U}{\partial m}; \quad -P = \frac{\partial U}{\partial V}. \quad (35)$$

Можно убедиться в том, что С-поле действительно консервативно, проверив справедливость соотношения взаимности Максвелла:

$$\frac{\partial h}{\partial V} = \frac{\partial -P}{\partial m}. \quad (36)$$

В соответствии с предположением об изэнтропичности внутренняя энергия на единицу массы является функцией только удельного объема:

$$u = u(1/\rho); \quad -P = \frac{du(1/\rho)}{d(1/\rho)}. \quad (37)$$

Выразив удельный объем как

$$1/\rho = V/m, \quad (38)$$

получим

$$\begin{aligned} P &= P(V/m), \quad u = u(V/m); \\ h &= u + \frac{PV}{m}, \end{aligned} \quad (39)$$

Тогда

$$\begin{aligned} \frac{\partial h}{\partial V} &= u' \cdot \frac{1}{m} + P' \frac{V}{m^2} + P \cdot \frac{1}{m} = \\ &= P' \frac{V}{m^2}, \end{aligned} \quad (40)$$

где штрихи означают производные функции u и P , и для исключения двух членов в (40) было использовано уравнение (31).

Дифференцирование P по m дает

$$\frac{\partial P}{\partial m} = -\frac{V}{m^2} P', \quad (41)$$

Сравнивая (41) с (40), убеждаемся в правильности уравнения (36).

Кононюк А.Е. Графы

Граф связей на рис.8 и соответствующие ему уравнения представляют сложный вариант модели эффектов сжимаемости в гидравлических сервомеханизмах с переменным объемом жидкости. Практически можно просто включить в систему уравнений линеаризованный параметр сжимаемости, однако часто оказывается фактически невозможным перейти от линеаризованной модели к более точной нелинейной модели, не возвращаясь снова к физическим принципам, лежащим в основе решаемой задачи.

Свойства, обусловленные инерцией и сжимаемостью одномерного потока

Простая модель, изображенная на рис. 1, приближенно описывает с помощью сосредоточенных параметров свойства инерционности и сжимаемости, присущие длинным линиям, пренебрегая при этом элементами, характеризующими потери. Если последовательно соединить большое количество таких моделей с параметрами, соответствующими длине Δl , а затем потребовать, чтобы $\Delta l \rightarrow 0$, то уравнения состояния для такой модели дадут аппроксимацию дифференциального уравнения в частных производных, называемую *одномерным волновым уравнением*. В предыдущих разделах были разработаны модели, учитывающие инерционные свойства несжимаемого потока и сжимаемость жидкости при отсутствии инерции. Возникает вопрос, можно ли построить модель распределенной линии с помощью простого наращивания ряда элементов с сосредоточенными параметрами, попеременно выбирая те из них, которые учитывают инерционные свойства или сжимаемость, как это показано на рис. 1. Ответ будет положительным для следующих важных случаев:

1. *Акустическая аппроксимация.* В этом случае малые изменения давления и плотности моделируются с помощью линейных уравнений. Коэффициенты инерции и сжимаемости вычисляются при средних значениях давления и плотности. Такая модель полезна при изучении влияния сжимаемости масел, в задачах о гидравлическом ударе, а также в акустике.

2. *Описание Лагранжа.* Если пытаться следовать за движением группы частиц, то инерция этой группы остается постоянной. Система уравнений может быть нелинейной только за счет нелинейных упругих или диссипативных свойств, обычно рассматриваемых в ньютоновской механике. Для исследования, например, колебаний стержней и струн лагранжево описание вполне естественно, так как частицы никогда далеко не перемещаются. Для балок и пластин дифференциальные уравнения сложнее, чем волновое уравнение, но и

Кононюк А.Е. Графы

в этом случае лагранжево описание позволяет использовать конечные модели, учитывающие инерцию и сжимаемость с помощью полей I и S . При этом узловая структура более сложная, чем для простого случая 0-1-0-1 струны на рис.1. Для систем с жидкостью лагранжево описание используется очень редко.

Исходя из стандартных компонент графов связей, можно для рассмотренных выше случаев легко построить модели с конечными элементами, которые, будучи уменьшены до элементарных объемов, приводят к дифференциальным уравнениям движения в частных производных для моделей с непрерывно изменяющимися параметрами. Такие характеристики системы, как инерционность, сжимаемость и сопротивление, можно рассматривать по отдельности с помощью конечных элементов, даже если эти характеристики относятся к одной точке при стремлении к нулю объема элемента модели с непрерывно распределенными параметрами. Как будет показано, обычное для гидромеханики эйлерово описание привлекает проблемы инерционных свойств и сжимаемости настолько основательно, что во многих случаях, за исключением акустических аппроксимаций, становится затруднительным построить последовательность конечных элементов, сходящихся к системе с непрерывно распределенными параметрами, не используя при этом большого числа активных связей.

Рассмотрим изэнтропический одномерный поток жидкости через трубу с единичной площадью поперечного сечения. Обозначив через s пространственную координату, через $v(s, t)$ —скорость, через $\rho(s, t)$ — плотность и через $P(s, t)$ — давление, получим следующие уравнения, описывающие систему:

$$\rho \frac{\partial v}{\partial t} + \rho v \frac{\partial v}{\partial s} = - \frac{\partial P}{\partial s} , \quad (42)$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v}{\partial s} = 0 , \quad (43)$$

$$P = P(\rho) \quad (44)$$

Уравнение (42) является записью закона Ньютона для элемента жидкости, ускорение которого выражено в форме Эйлера. Уравнение (43) формулирует закон сохранения массы, а выражение (44) можно использовать для определения влияния сжимаемости. Плотность $\rho(s, t)$ фигурирует как в законе инерции (42), так и в законах сжимаемости. В противоположность этому при лагранжевом описании инерцию можно

Кононюк А.Е. Графы

выразить в функции постоянной плотности жидкости в исходном состоянии.

Очевидно, нет смысла пытаться строить граф связей для уравнений (42) — (44), если только ρ в (42) не является константой (в уравнениях (43) и (44), величина ρ может изменяться) и если vdv/ds представляет собой величину второго порядка малости. По существу уравнение (42) представляет закон Ньютона для потока, разделенного на бесконечно малые элементы. В каждый последующий момент времени этот закон применим к другому бесконечно малому элементу жидкости. Таким образом, в случае меняющихся во времени инерционных свойств энергия не будет постоянной, в то время как инерция, учитываемая графом связей, обеспечивает постоянство энергии. К тому же член dv/ds показывает, что скорости на двух границах элементарного контрольного объема различны. Поэтому в уравнении (42) учтены одновременно влияния как инерции, так и сжимаемости. Наконец, давление и скорость (или объемный расход, поскольку мы рассматривали единичную площадь поперечного сечения) в этих уравнениях не дают в результате перемножения истинный поток мощности через сечение s в момент времени t . Как мы видели, конвекционная внутренняя энергия в этих уравнениях не учитывается. Не следует поэтому ожидать, что эти (правильные) уравнения могут быть представлены с помощью графов связей, которые основаны на сохранении мощности и энергии. Для представления гидродинамических линий с помощью графов связей, по-видимому, существуют следующие возможности:

- a) использование описания Лагранжа,
- b) использование описания Эйлера, но с ограничением по существу до акустической аппроксимации,
- c) рассмотрение только несжимаемых или безынерционных жидкостей, как это сделано выше. Таким образом, схема, изображенная на рис.1, будучи интуитивно привлекательной, является, по-видимому, более ограниченной, чем это могло показаться при общей оценке даже без учета таких явлений, как касательные напряжения, пограничные слои и т. д., которые усложняют моделирование гидравлических передающих линий.

Заключение

Показано, что многие гидромеханические системы можно описать с помощью элементов графов связей такого же типа, который обычно используется в электрических и механических системах. В некоторых случаях встречаются элементы, необычные для других видов систем.

Кононюк А.Е. Графы

Например, сопротивление, которое соответствует уравнению Бернулли, представляет уменьшение или увеличение связанной со статическим давлением мощности в зависимости от уменьшения или увеличения динамического давления. Сопротивление в действительности представляет инерционные свойства, как это видно из описания Эйлера или из представления с помощью контрольных объемов. Если проявляются оба свойства — инерционность и сжимаемость — и движение жидкости описывается в форме Эйлера, то процесс перехода к сосредоточенным параметрам является более сложным. Наиболее известные системы с сосредоточенными параметрами представлены в форме Лагранжа, поэтому нет ничего удивительного в том, что в открытых системах, через которые может протекать поток вещества или энергии, наблюдаются неожиданные эффекты. В некоторых важных случаях (например, при наличии несжимаемости) модели, очень близкие к лагранжевым системам с сосредоточенными параметрами, могут быть обнаружены для эйлеровых систем. Это обстоятельство объясняет, почему многие исследователи никогда даже не задумывались над различием между описаниями Эйлера и Лагранжа, создавая грубые модели с сосредоточенными параметрами для систем с распределенными параметрами.

1.5. Модели с распределенными параметрами сердечно-сосудистой системы человека

В настоящем разделе графы связей использованы для построения модели с распределенными параметрами сердечно-сосудистой системы человека. Модель включает сердце, артериальную, капиллярную и венозные части как системного, так и легочного кровообращения. Представлено сравнение результатов моделирования с данными физиологических экспериментов. Вследствие использования универсального языка графового моделирования для систем с распределенными параметрами, модель обладает большой функциональной гибкостью и может быть модифицирована в зависимости от цели применения. Одним из таких применений, которое исследовалось детально в данной работе, является моделирование движений тела, вызванных течением крови, у человека, механически изолированного от земли. Регистрация этого движения, называемая баллистокardiографией, обеспечивает возможность косвенного определения характеристик работы сердца.

Введение

В этом разделе изложен метод построения крупномасштабной модели сердечно-сосудистой системы человека, предложенный Д.Ауслендером, Т. Лобделом, Д. Чжуном, демонстрирующий использование графов связей, которые обладают всеми своими традиционными достоинствами: универсальностью, алгоритмической представимостью систем уравнений, эффективностью процесса вычисления и т. д. Авторы применили их для описания системы с распределенными параметрами, а не к системе с сосредоточенными параметрами, для анализа которой графы связей обычно применяются. Хотя распространение волны давления по системе кровообращения млекопитающих указывает на то, что мы имеем дело с системой с распределенными параметрами, предыдущие работы по моделированию были посвящены созданию больших и малых моделей с сосредоточенными параметрами, в которых участки системы кровообращения моделировались *RLC*-цепочками. Причина такого усиленного интереса к моделям с сосредоточенными параметрами заключалась в том, что исследователи применяли для их создания хорошо знакомую им методику решения систем линейных дифференциальных уравнений.

Реализация крупномасштабной модели на ЭВМ позволяет достигнуть наибольшей гибкости и практичности; кроме того, любое цифровое решение независимо от типа модели требует, чтобы численный алгоритм обеспечивал решение в дискретные моменты времени. Для решения задачи моделирования использовалась цифровую модель, работающую в дискретном времени и включающую динамические элементы с распределенными параметрами, в результате чего несколько повысилась эффективность вычислений и адекватность моделирования участков сосудов.

При конструировании модели использовались графы связей, что позволило отказаться от составления и решения систем уравнений. Такой подход позволил сохранить максимальную структурную гибкость. Например, ввиду наличия многочисленных данных, описывающих систему кровообращения собак, проверялись некоторые начальные концепции на основании этих данных, что позволило затем перейти к моделированию системы кровообращения человека без перепрограммирования. Для этого было достаточно изменить данные, описывающие топологию системы, и специфические физические и геометрические параметры. Объяснение такой простоты в работе заключается в универсальности программы графового моделирования. Данная программа, которая названа *DBOND*, во многом сходна с более известной программой *ENPORT* по структуре, но не содержит ни

Кононюк А.Е. Графы

индуктивных, ни емкостных элементов. Здесь вся динамика системы описывается идеальными двусторонними линиями передач. Программа DBOND вырабатывает переходную матрицу, основанную на топологических данных и параметрах системы. Поэтому потери в эффективности расчетов при использовании этой программы сводятся к минимуму. В этом смысле программа работает скорее как компилятор, а не как интерпретатор, что важно для решения задач, подобных изложенных в методе, требующих множества цифровых манипуляций.

Целью при разработке модели системы кровообращения было не только показать, что возможно получить «правильный» ответ. Показано, что можно построить физиологически достаточно обоснованную модель, обладающую большой вычислительной гибкостью, которая могла бы стать не только средством исследования, но и, возможно, орудием клинической практики. Основной предпосылкой для достижения хорошего физиологического соответствия является то, что линия передачи характеризуется длиной, скоростью распространения волны и характеристическим импедансом, т. е. теми параметрами, которые близко соответствуют нормальным анатомическим и физиологическим параметрам, особенно в артериальной системе. В соответствии с этой целью при составлении программы в качестве входных использовались только основные данные, полученные в предыдущих исследованиях, за исключением тех данных, которые не были найдены. В настоящее время имеется достаточно данных для моделирования артериальной части сердечно-сосудистой системы, но трудно найти данные для моделирования сердца и венозной части системы кровообращения. Действительно, механизмы венозного кровотока до сих пор еще полностью не выяснены.

Проблема, которая требовала для своего решения такого подхода, привлекла внимание авторов вскоре после создания основной модели кровообращения. Эта проблема связана с попыткой определения клинического значения баллистокардиографических записей. Баллистокардиограмма (БКГ) записывается с пациента, лежащего на очень легком столе, который смонтирован так, что может двигаться в горизонтальном направлении почти без трения. Для записи движений стола при отсутствии каких-либо внешних сил, действующих в горизонтальном направлении, используется акселерометр или другой прибор. Так как в таком положении движущимися массами в теле являются только кровь и сердце, то акселерометр регистрирует некий параметр, определяющий общее поведение сердечно-сосудистой системы. Таким образом, теоретически возможно применять БКГ для

Кононюк А.Е. Графы

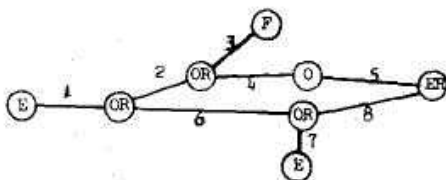
косвенного измерения производительности сердца, т. е. такого параметра, который в настоящее время можно измерить только при проникновении внутрь тела. Баллистокардиография известна более ста лет (фактически еще в 1877 г. было установлено, что движение крови и сердца вызывает перемещение тела), но все еще остается только перспективным методом для косвенного определения производительности сердца.

Задача авторов заключалась в проведении исследования с целью показать возможность использования крупномасштабного моделирования БКГ для физиологической интерпретации различных пиков и впадин, типичных для БКГ человека. Использование крупномасштабного моделирования для расшифровки БКГ вызвано тем, что в настоящее время нельзя получить экспериментальной зависимости между большим числом внутренних состояний сердечно-сосудистой системы и БКГ у человека.

Программа DBOND

Здесь приводится краткий обзор этой работы; DBOND использует несколько модифицированную форму графов связей, как это показано на рис.1. Узловые элементы, взятые в кружок, представляют собой статические узлы и обычно являются комбинациями нескольких стандартных статических элементов графов связей.

Конюк А.Е. Графы



Код : E 1; OR 1,2,6; OR 2,3,4; F 3; O 4,5; ER 5,8; OR 6,7,8; E 7

(а) **Пример простого графа связей-задержек и его кодовая форма**



(б) **Статические узлы и их эквиваленты**

Рис.1. Характеристические графы для программы DBOND.

Некоторые примеры узловых элементов показаны на фиг. 1, б. Линии в отличие от стандартных графов связей, где они отображают идеальные связи, используются для характеристики элементов линии передачи, которые отображают динамику системы. В отличие от стандартных графов связей здесь должны быть введены данные как для линий, так и для узловых элементов, а именно время задержки волн, и характеристический импеданс для каждой линии (или задерживающие связи, как

Кононюк А.Е. Графы

они называются) и сопротивления, напряжения источников и т. д. для статических узловых элементов. Работа программы DBOND во многом схожа с работой программы ENPORT. Описание топологии системы вводится с помощью кода, полученного путем нумерации линий и использования характерных названий для узлов. Пример этого кода показан на рис.1.а. После считывания в закодированном графе топологическая информация используется для заполнения списков в памяти, которые при помощи перекрестных указателей отображают структуру внутренних данных, содержащую все необходимое для представления топологии системы. Далее вводятся параметры линий и узлов и эти записи в памяти используются с целью организации данных для последующего использования. На этапе настройки программы или «компилирования» вся эта информация используется для составления переходных матриц и установки сдвиговых регистров для моделирования распространения волны по линии передачи. Затем выбирается желаемый графический или табличный выход и программа готова к запуску. Для моделирования нелинейных элементов во время работы по программе соответствующие элементы переходных матриц рассчитываются во времени на каждом шаге.

Дискретизация времени достигается наложением условия синхронизации на линии передач. Линии передач, которые используются программой DBOND, идеализированы в том смысле, что они представляют собой линии без потерь и без дисперсии. При распространении волн по этим линиям их амплитуда и форма остаются неизменными и они не взаимодействуют с волнами, распространяющимися в другом направлении. Волны взаимодействуют друг с другом только на концах линий через статические узлы. Здесь волны отражаются и передаются согласно обычным законам волнового рассеяния. Синхронизация обеспечивает расчет волн таким образом, что все волны прибывают в узлы одновременно. Если бы этого не было, то образовывались бы новые волны, которые передавались бы по линиям, связанным с этим узлом всякий раз по прибытии волны в данный узел. Таким образом, синхронизация обеспечивает постоянство числа волн в системе. Практическим следствием этого является то, что времена распространения волн (времена задержки) должны быть целыми кратными величинами шага вычислений по времени. Если требуется лучшее разрешение, то необходимо использовать меньшую величину шага. Переходная матрица представляет собой упорядоченный набор всех коэффициентов рассеяния; она описывает множество новых волн, возникших в результате прибытия всех волн в системе в соответствующие узлы.

Кононюк А.Е. Графы

Сосудистая модель

Граф связей-задержек для сердечно-сосудистой модели показан на рис.2, где учтены некоторые важные сосуды и анатомические особенности.

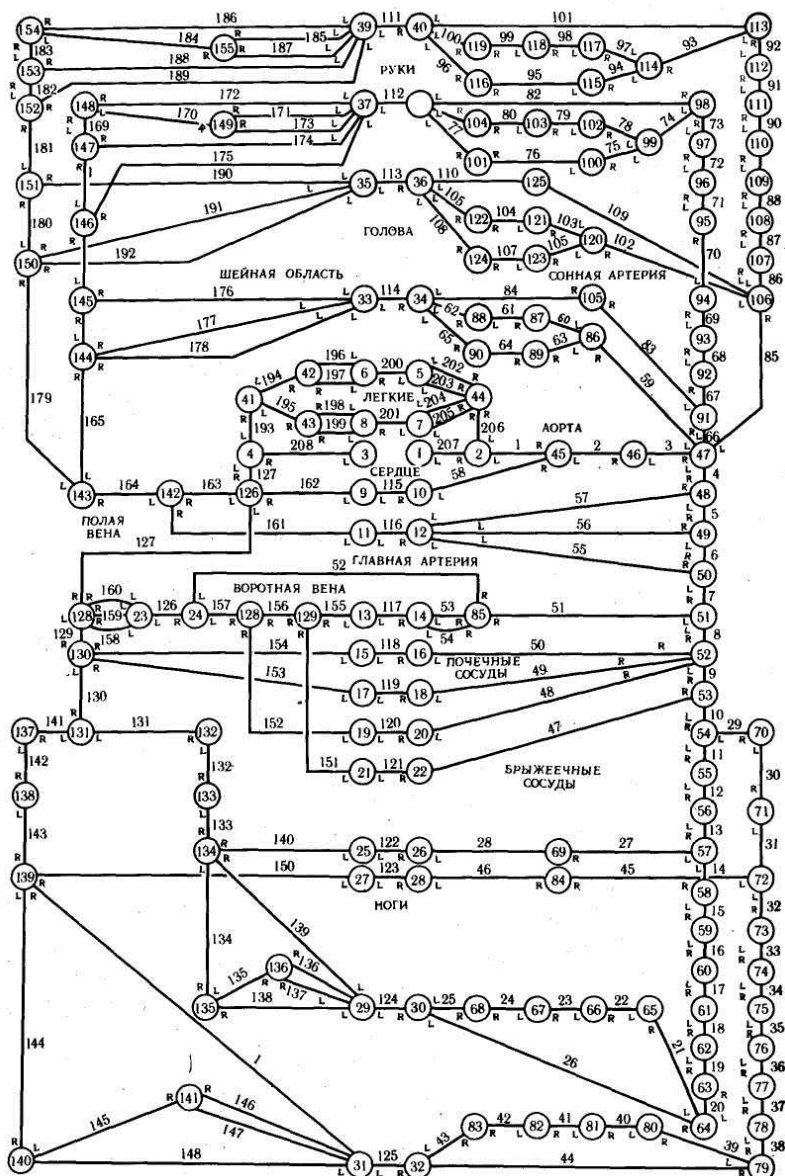


Рис.2. Граф связей-задержек модели сердечно-сосудистой системы

Кононюк А.Е. Графы

В большинстве случаев каждая линия задержки отображает динамику одного кровеносного сосуда. В некоторых случаях используются две линии задержек или более для моделирования изменений поперечного сечения или свойств по длине сосуда. Сопротивления отображены в свойствах узлов, которые в общем случае являются *OR*-узлами. Выражения для времени распространения волны T , характеристического импеданса Z и сопротивления R , характеризующие свойства отрезка кровеносного сосуда постоянным сечением, предложенные Лобделлом, приводятся ниже:

$$T = \sqrt{IC} = l(a + 1) \left[\frac{3\rho}{E(2a + 1)} \right]^{1/2} \quad (1)$$

$$Z = \sqrt{I/C} = \frac{l}{S(a + 1)} [1/3\rho E(2a + 1)]^{1/2} \quad (2)$$

$$R = \frac{8\pi\mu l}{S^2}, \quad (3)$$

где I — общая распределенная инертность, C — общая распределенная податливость, (μ — вязкость крови (принятая равной $3 \cdot 10^{-2}$ пз), ρ — плотность крови ($1,05 \text{ г}\cdot\text{см}^{-3}$), S — площадь внутреннего сечения сосуда, a — отношение внутреннего радиуса сосуда к толщине стенки, l — длина отрезка, E — модуль Юнга.

63 системных (термин «системная» относится к циркуляции от сердца к остальным частям тела и обратно в отличие от «легочной» циркуляции, которая относится к движению крови от сердца к легким и обратно) артериальных сосуда внутренним диаметром более 0,1 см моделировались 110 отрезками. Характеристики 49 из этих сосудов взяты из анатомических данных, а характеристики остальных сосудов (двух коронарных и двенадцати грудных артерий) были оценены по параметрам сосудов подобного же размера. Зависимость упругости от размеров сосуда моделировалась при помощи введения различных значений модуля Юнга. Величина модуля Юнга бралась для крупных сосудов равной $4,0 \cdot 10^6$, для средних сосудов $8 \cdot 10^6$ и для самых мелких сосудов $16 \cdot 10^6 \text{ г}\cdot\text{см}^{-1}\cdot\text{с}^{-2}$.

53 системные вены в модели были представлены 65 отрезками. Так как данных о венозных сосудах было явно недостаточно, их характеристики оценивались с использованием следующих

Кононюк А.Е. Графы

допущений: вена представляет собой сосуд, имеющий круглое сечение; отношение внутреннего диаметра больших вен к внутреннему диаметру артерий того же названия равно 1,5; отношение внутреннего диаметра вены к толщине ее стенки в 1,5 раза больше отношения внутреннего диаметра к толщине стенки артерии того же диаметра; длина вены равна длине артерии того же наименования; значение модуля Юнга для всех вен равно $1,2 \cdot 10^6 \text{ г} \cdot \text{см}^{-1} \cdot \text{с}^{-2}$. Длины и диаметры вен конечностей оценивались по анатомическим рисункам и из предположения, что площади поперечного сечения пропорциональны принятым относительным расходам через эти вены.

Системные сосуды внутренним диаметром менее 0,1 см были объединены с соответствующим капиллярным руслом. 60 капиллярных русел смоделировались 60 отрезками. Каждое капиллярное русло моделировалось камерой смещения, не имеющей сопротивления, податливость которого определялась по объему крови и давлению в капиллярном русле. Давление в русле было принято равным 25 мм рт.ст. для всех системных русел, кроме печеночного, где оно было принято равным 8 мм рт. ст. Величины объемов в руслах оценивались приложением принятого процентного распределения капиллярного кровотока к общему принятому объему крови во всех сосудах, сгруппированных в русла. Податливость рассчитывалась из следующего соотношения:

$$C = \frac{V}{P},$$

(4)

где V представляет собой общий объем крови в русле и P — давление в русле. Этот метод расчета дает приемлемую аппроксимацию функции податливости в узком диапазоне давлений. Среднее время нахождения кровяных клеток в моделируемом капиллярном русле было принято равным 11 с. Эта величина довольно точно соответствует времени 7 с, требующемуся кровяным клеткам для прохождения через действительное капиллярное русло. Модель легочного кровообращения была разработана на основании анатомических данных, взятых из литературы. Моделировались только 11 главных сосудов отдельно от капиллярных русел. Длины и поперечные сечения семи артерий и четырех вен были оценены на основании данных, полученных на собаках. Величины податливости сосудов были взяты прямо из литературных данных, а не рассчитывались аналитически, как для случая системных артерий. Капиллярные русла моделировались двумя податливыми резервуарами. Общая податливость для этих резервуаров была рассчитана по формуле (4) для объема 75 мл и давления 9 мм рт.

Кононюк А.Е. Графы

ст. Время пребывания кровяных клеток в капиллярном русле, равно 0,75 с, хорошо согласуется со временем пребывания частиц крови в капиллярном русле, обычно указываемом в литературе.

Периферические сопротивления рассчитывались для определения давления и кровотока в русле. В расчетах давление в венах, подходящих к левому и правому отделам сердца, принималось равным 7,5 и 5,5 мм рт. ст. соответственно.

Дополнительно к этим детальным особенностям отдельных сосудов модель должна воспроизводить некоторые общие свойства сосудистой системы в целом. Например, общие геометрические характеристики (основанные на данных для собаки весом 13 кгс) показаны на рис.3.

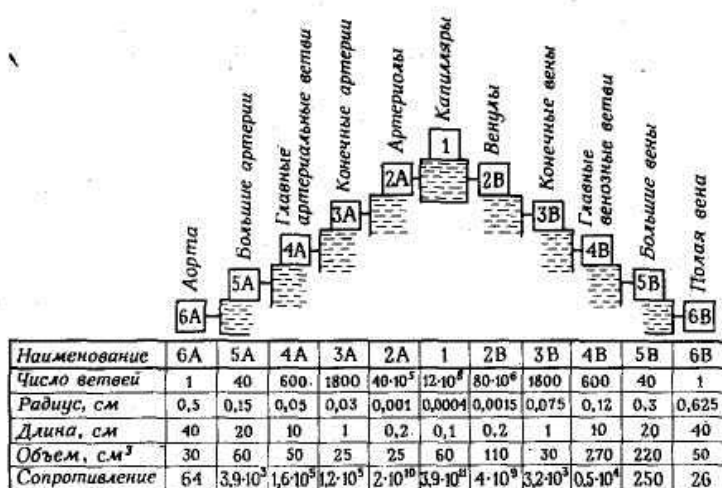


Рис.3. Геометрия сосудистого русла.

Аорта представляет собой трубку с площадью поперечного сечения 0,8 см², постепенно разветвляющуюся на $12 \cdot 10^8$ капилляров, площадь поперечного сечения каждого из которых равна $5 \cdot 10^{-7}$ см². Общая площадь поперечного сечения всех капилляров равна 600 см², а их общая поверхность равна 30 м². Поэтому скорость кровотока должна уменьшаться от 38 см/с в аорте до 0,05 см/с в капилляре.

Распределение давления в системе представлено на рис.4.

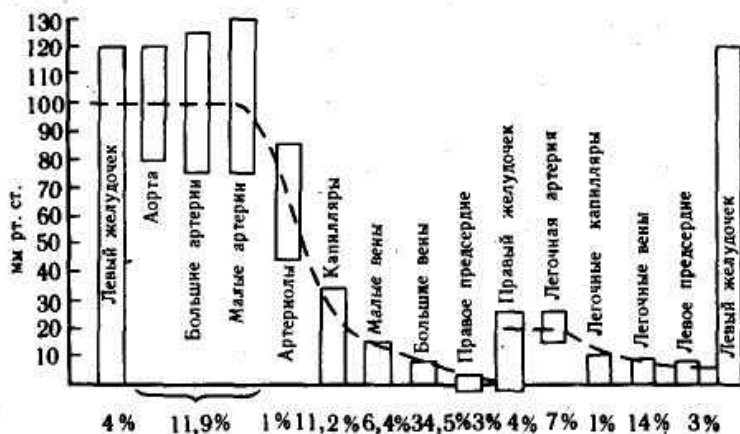


Рис.4. Кровяное давление и распределение объемов крови.

Пунктирная линия показывает среднее давление, а высоты прямоугольников показывают пульсационные величины давления. Следует отметить, что среднее давление уменьшается незначительно после аорты, а пульсации давления возрастают. Основное падение давления в артериальной системе приходится на артериолы. Распределение объема крови в различных частях системы показано на рис.4 под прямоугольниками пульсационного давления.

Модель сердца

Желудочки левого и правого отделов сердца моделировались двумя линиями задержки, податливость которых менялась со временем. Такое моделирование работы желудочков основано на методе, предложенном Бенценом, при котором податливость, связывающая объем желудочка с давлением, уменьшается во время сокращения сердечной мышцы и увеличивается при ее расслаблении. Для моделирования предсердий использовались пассивные линии. Клапаны моделировались резисторными узлами *OR*, сопротивление которых изменялось в соответствии со знаком разности давлений на этих узлах. Значения параметров желудочков устанавливались из условия получения заданных соотношений между давлением и расходом в сосудах, непосредственно связанных с сердцем.

Каждый желудочек при моделировании считался податливым цилиндром, закрытым с одного конца.

Кононюк А.Е. Графы

Комбинируя уравнения (1) и (2), можно убедиться, что податливость меняется по длине трубы прямо пропорционально площади ее поперечного сечения, поэтому податливость цилиндра уменьшается при уменьшении его объема во время фазы выброса при работе сердца. Податливость в конце фазы выброса рассчитывалась путем подстановки объема и давления в конце системы в уравнение (4). Кривая изменения податливости в течение фазы выброса определялась эмпирически и несколько изменялась по сравнению с первоначальной формой в ходе баллистокардиографических исследований.

В соответствии с использованной методикой расчета моделируемое давление в полностью закрытой линии обратно пропорционально квадратному корню из ее адмиттанса. Поэтому фазы изохорического сокращения и расслабления при работе сердца моделировались изменением адмиттансов линий задержки желудочков таким образом, что давление росло или уменьшалось во времени в соответствии с заданными кривыми. Во время фазы заполнения поддерживался постоянный импеданс, причем давление в полости желудочков во время этой фазы оставалось почти постоянным.

Работа модели

Моделируемые давления и кровотоки в сердце и аорте показаны на рис.5.

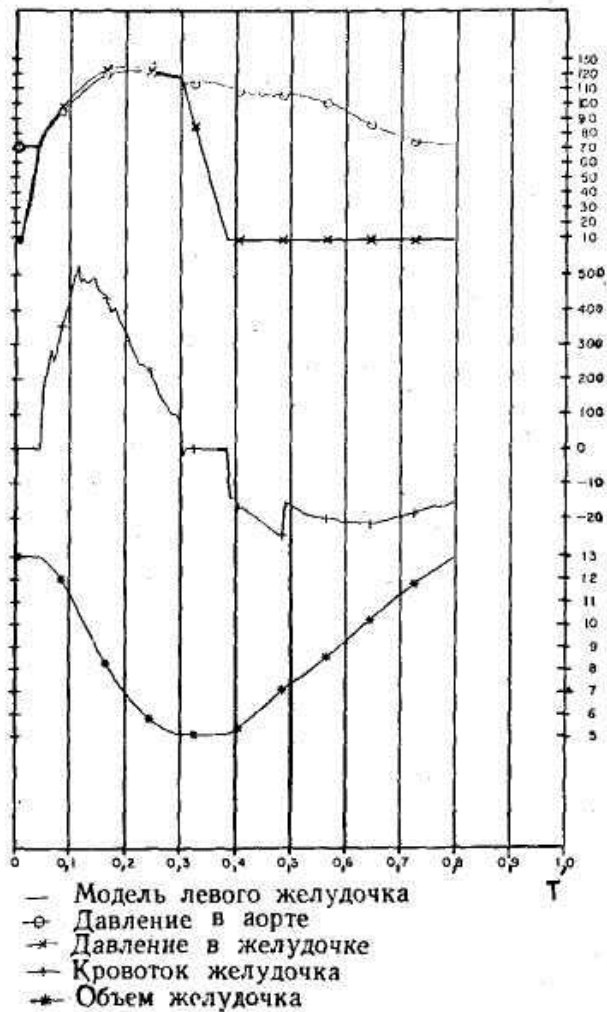


Рис.5. Моделируемые давление и кровоток в сердце и аорте.

Кононюк А.Е. Графы

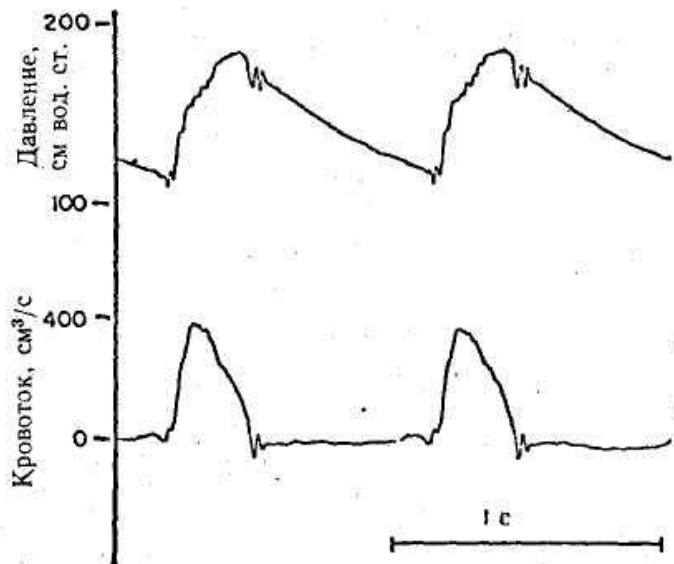


Рис.6. Измеренные давление и кровоток в аорте.

Их можно сравнить с данными физиологического эксперимента, представленными на рис.6.

Подобное же сравнение можно провести и для артериальных пульсовых давлений в различных точках тела. Моделируемые давления приведены на рис.7, а измеренные давления для тех же самых точек тела приведены на рис.8.

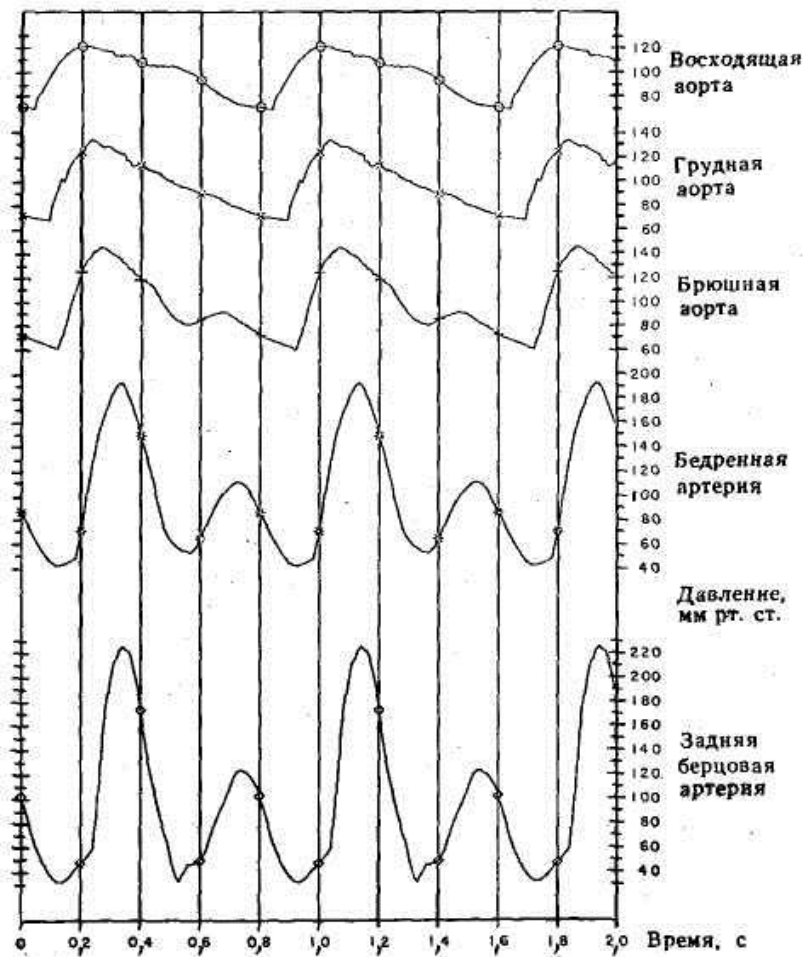


Рис.7. Моделируемые артериальные пульсовые давления.

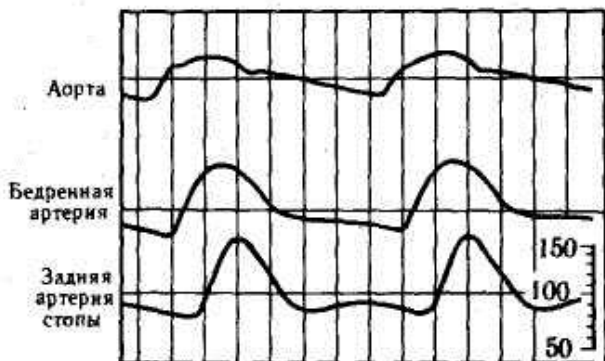


Рис.8. Измеренные артериальные пульсовые давления

Причины увеличения пульсового давления и амплитуды диастолического давления вдоль аорты, полученного на модели, не известны. Это может быть следствием неучета дисперсии в некоторых больших артериях при моделировании. Волновые скорости в различных участках артериальной системы, полученные на модели, отличаются в пределах 15% от значений, полученных в эксперименте. Одной из самых больших трудностей при исследовании сердечно-сосудистой системы человека *in vivo* является получение надежной информации о величине кровотока. На модели эту информацию можно получить легко. Моделируемые артериальные кровотоки показаны на рис.9.

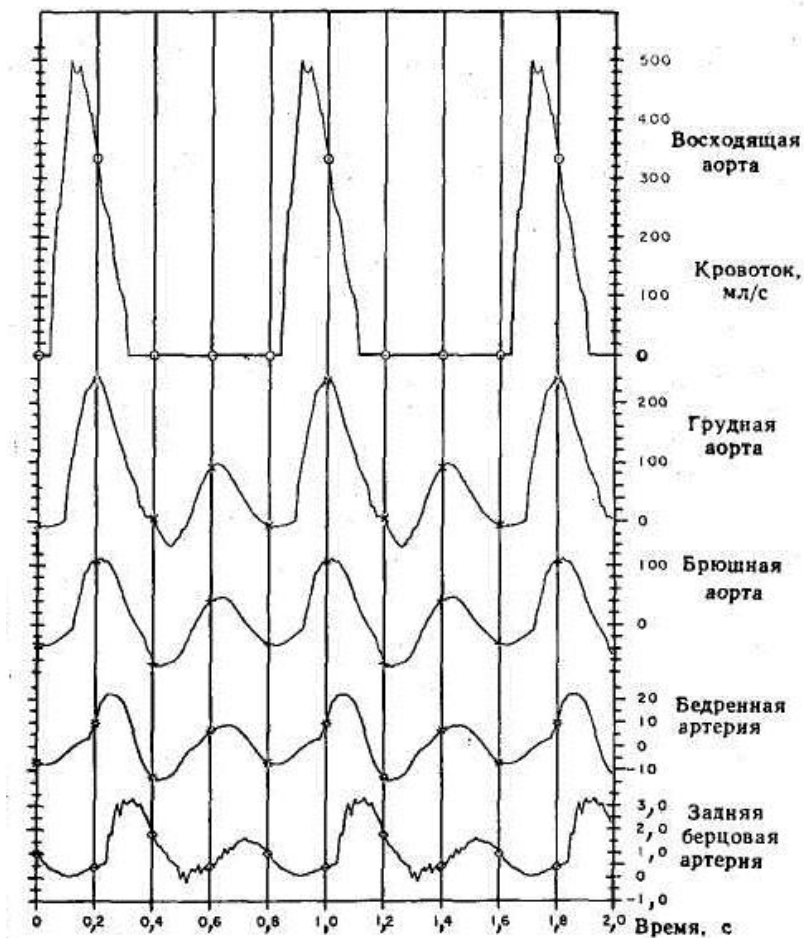


Рис.9. Артериальные кровотоки.

Эти данные могут быть сопоставлены с давлениями в тех же самых точках, представленными на рис.7. Доказательством экспериментальных трудностей при измерении кровотока может служить то, что не были найдены данные о кровотоке в литературе для сравнения с результатами моделирования.

Кононюк А.Е. Графы

В целях демонстрации универсальности модели и возможности ее использования в клинических исследованиях модель использовалась для изучения обморочного состояния, эмболии и гипертензии.

Применение модели для баллистокардио-графических исследований

В дальнейшем осуществлялось моделирование процесса баллистокардиографии (БКГ) человека, заключающейся в определении перемещений тела, вызванных течением крови у человека, механически изолированного от земли. При моделировании предполагалось, что БКГ учитывает только движения тела, вызванные током крови, а другие эффекты, такие, как мышечные сокращения, движение сердечной стенки и другие движения тела, не учитываются. БКГ рассчитывается путем применения закона сохранения количества движения к модели кровообращения; поскольку мы предполагаем, что основной массой, находящейся в движении, является масса крови, количество ее движения в данном направлении обуславливает равное, но противоположно направленное количество движения самого тела (влиянием малых перемещений тела на кровоток мы пренебрегаем). Баллистокардиографическая кривая скорости может быть рассчитана непосредственно по значениям этого количества движения и массы тела (принимавшейся равной 75 кг). Перемещение рассчитывалось путем интегрирования, а ускорение путем дифференцирования кривой изменения скорости. Так как использованный баллистокардиограф основан на измерении количества движения, а не перемещения тела, то для расчета БКГ-кривой ускорения требуется только одна операция дифференцирования, а не две операции дифференцирования, требующиеся для расчета ускорения при методе баллистокардиографии, основанном на использовании фиксированного центра массы.

Для каждой линии задержки модели продольная компонента количества движения крови рассчитывалась исходя из следующего уравнения:

$$\begin{aligned} \text{Количество движения в продольном направлении} &= \\ &= (\text{масса жидкости}) \times (\text{скорость крови}) \times \\ &\times (\text{характеристика направления}) = (\rho S l) \cdot (Q/S) (d) = \rho l Q d, \end{aligned}$$

где ρ — плотность крови, г/см³; l — длина участка сосуда, см; S — площадь поперечного сечения участка, см²; Q — расход крови, см³/с (средний расход в конце каждой линии); d — характеристика на-

Кононюк А.Е. Графы

правления линии, равная косинусу угла между направлением данной линии и линией положительного направления, идущей от ног к голове. Таким образом, суммарное количество движения крови вдоль продольной оси тела определяется следующим выражением:

$$\begin{aligned} \text{Количество движения в продольном направлении} = \\ = (\text{масса жидкости}) \times (\text{скорость крови}) \times \\ \times (\text{характеристика направления}) = (\rho S l) \cdot (Q/S) (d) = \rho l Q d, \end{aligned}$$

где i — номер отрезка (имеется 208 линий в модели). После этого баллистокardiографическая кривая скорости может быть рассчитана легко на основании закона сохранения количества движения из уравнения:

$$v = \frac{\text{суммарное количество движения}}{\text{масса тела}},$$

где v — баллистокardiографическая скорость (положительное направление принято от ног к голове). При использовании первоначальной модели в качестве базовой для баллистокardiографического исследования в программу необходимо было ввести только геометрическую информацию — длины и характеристики направлений сосудистых участков. Эта новая информация вводилась в программу и затем производился расчет баллистокardiографических кривых с помощью новой подпрограммы. Для моделирования баллистокardiографии часто используется другой метод, основанный на использовании центра масс. Этот метод основан на том, что общий центр масс тела и крови остается неизменным в условиях механической изоляции от земли и что при этих условиях смещение центра массы крови будет вызывать соответствующее смещение центра массы остальных частей тела. Недостаток этого метода состоит в том, что непосредственно регистрируется смещение тела и для получения БКГ-кривой ускорения необходимо двойное дифференцирование БКГ-кривой перемещения. Двойное дифференцирование приводит к значительным погрешностям БКГ-кривой ускорения.

Так как моделирование полностью проводится на цифровой машине, то можно использовать методы экспоненциального сглаживания для расчета БКГ-кривой ускорения и получать производные по времени для различных давлений и кровотоков в модели. Этот метод сглаживания и дифференцирования имеет следующее преимущество: в память машины необходимо вводить только некоторые статистические данные, характеризующие все прошлые наблюдения, и предписывать геометрически уменьшающиеся веса более старым данным. Кроме того, производные более высокого порядка могут быть рассчитаны для

Кононюк А.Е. Графы

данной переменной одновременно. Это необходимо для расчета БКГ-кривой ускорения из БКГ-кривой смещения при методе центра масс.

Важной целью моделирования БКГ является сравнение получаемых при таком моделировании результатов с изменением рабочих характеристик сердца и сосудистой системы. Поэтому еще до моделирования БКГ исследовалось влияние внутренних изменений на характеристики сердца, чтобы создать лучшую базу для сравнения. Так как на сердечно-сосудистую систему в гораздо меньшей степени влияют дефекты в работе правого желудочка, чем соответствующие дефекты в работе левого желудочка, то указанные выше внутренние изменения производились в левом желудочке. Причина этого заключается в том, что в норме правый желудочек начинает сокращаться раньше и наращивает свое внутреннее давление более медленно по сравнению с левым желудочком, и в результате отток крови из правого желудочка происходит более медленно. Поэтому пиковый кровоток правого желудочка намного меньше пикового кровотока левого желудочка.

Особое внимание, в частности, было уделено величине производной давления в левом желудочке, так как Мэзон установил, что этот показатель отражает главные свойства сократимости желудочка и что он очень полезен для оценки направленных изменений в сократимости при терапевтических или других воздействиях на пациента. Кроме того, исследования Гаррисона показали, что существует зависимость между максимальным значением производной давления левого желудочка и *HI*-волной (0,65) БКГ-кривой ускорения (эти «волны» характеризуются условными обозначениями пиков баллистокardiограммы), *IJ*-волной (0,81) и *HI*-пиком (0,63), который является производной БКГ-кривой ускорения (рис.15). Однако не было обнаружено значительной корреляции между максимальным значением давления в левом желудочке с предыдущими тремя волнами. Этот результат согласуется с экспериментальными данными, которые показывают, что системная гипертензия мало сказывается на форме баллистокardiографической кривой. На основании вышеизложенного большое внимание было уделено при моделировании изменению формы кривой производной давления левого желудочка при сохранении величины давления в приемлемых пределах.

Первоначальная кривая давления в левом желудочке (штриховая линия) и ее модифицированная форма (сплошная линия) приведены на рис.10.

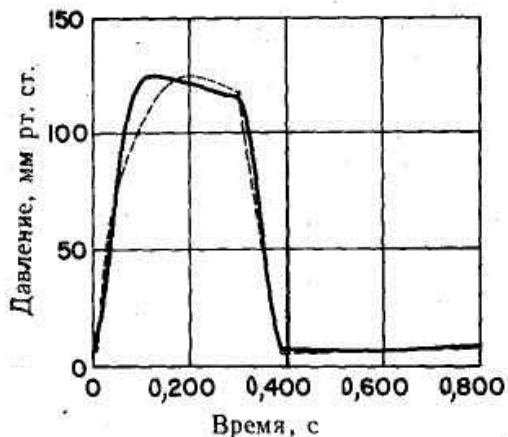


Рис.10. Первоначальная (штриховая линия) и модифицированная (сплошная линия) кривые давления в левом желудочке.

На рис.11 показаны производные давления для первоначальной и модифицированной форм.

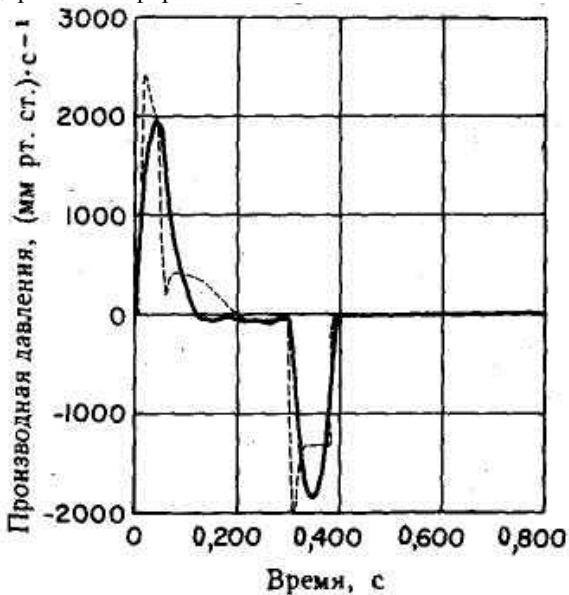


Рис.11. Первоначальная (штриховая линия) и модифицированная (сплошная линия) кривые производных давления в левом желудочке.

Кононюк А.Е. Графы

Модифицированная кривая производной давления состоит из положительного пика, соответствующего фазе выброса, и отрицательного пика параболической формы, соответствующего фазе расслабления. Такая форма кривой производной давления в левом желудочке обеспечивает хорошую аппроксимацию. Другим важным показателем работы сердца, отражающимся на баллистокardiографической кривой, является производная или ускорение кровотока в восходящей аорте. Эта кривая представлена на рис.12 для средней точки восходящей аорты.

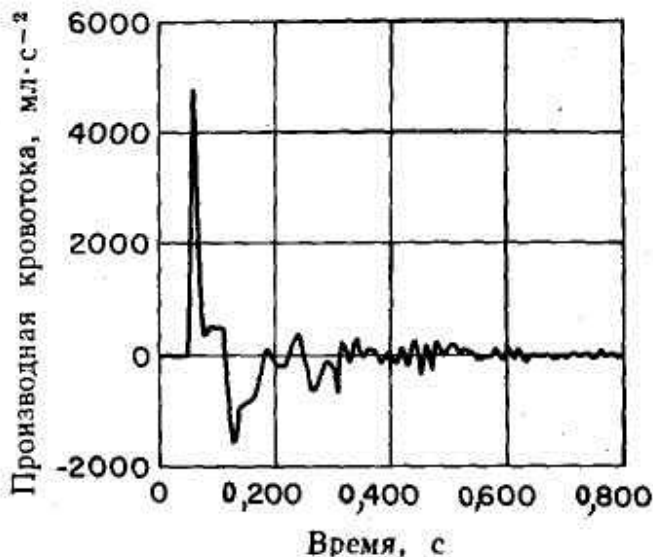


Рис. 12. Производная кровотока в середине восходящей аорты.

Во время фазы выброса кривая производной кровотока имеет резкий положительный всплеск, который является самым чувствительным показателем, характеризующим состояние сердца, хотя измерить этот показатель у пациента нелегко. Однако при моделировании этот показатель легко получить для сравнения с БКГ и изменениями в ней. На рис. 13—15 представлены графические результаты моделируемых баллистокardiограмм, основанные на законе сохранения количества движения.

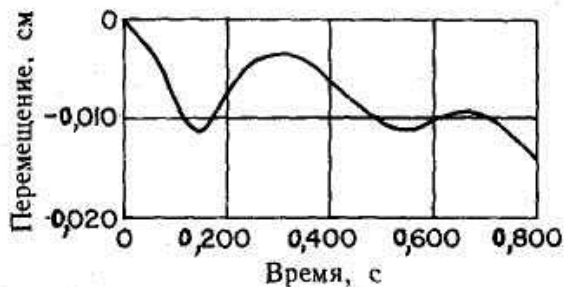


Рис. 13. Моделируемое баллистокардиографическое смещение, полученное с помощью закона сохранения количества движения.

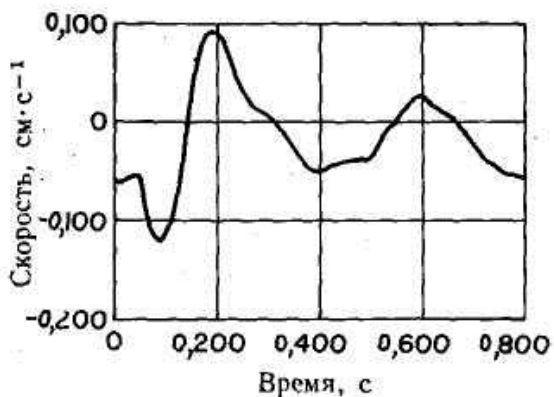


Рис.14. Моделируемая баллистокардиографическая кривая скорости.

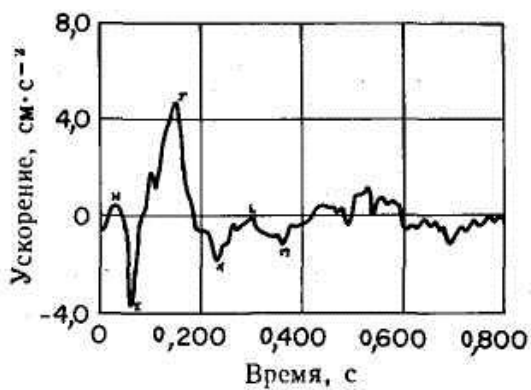


Рис.15. Моделируемая баллистокардиографическая кривая ускорения или силы.

При данном методе моделирования БКГ-кривая перемещения не возвращается к исходному значению в конце сердечного цикла. Исследования также показали, что если для моделирования БКГ используется только артериальная система, то происходит положительное смещение кривой перемещения. Но, когда моделирование БКГ дополняется включением влияния венозного возврата, смещение кривой становится отрицательным. Если провести новую базовую линию из начального положения к значению в конце сердечного цикла для предыдущих двух случаев, то формы кривых перемещения будут одинаковыми для этих новых базовых линий. Из этого следует, что венозный кровоток мало влияет на БКГ-кривую ускорения. Если для определения БКГ-кривой перемещения при помощи той же самой модели использовался бы метод, основанный на постоянстве центра тяжести, то смещение кривой в конце сердечного цикла не наблюдалось бы вследствие самой природы этого метода независимо от учета венозного возврата или от того, что рассматривается только часть артериальной системы.

БКГ-кривая ускорения представлена на рис.15. Моделирование показало, что волна *HIJ* четко представлена, хотя *K*-, *L*-, *M*- и *N*-волны выражены менее явно. Во время первоначального моделирования, однако, оказалось, что амплитуда волны *I* была наполовину меньше амплитуды волны *J*; этот первоначальный дефект был скорректирован путем модификации модели левого желудочка. Причина заключалась в слишком быстром уменьшении производной артериального давления в желудочке после открытия аортального клапана. Этот недостаток приводил к слишком медленному росту соответствующего давления, что отрицательно сказывалось на БКГ. Модифицированная модель сердца позволяет получить БКГ, у которой амплитуда волны *I* немного меньше амплитуды волны *J*, что хорошо согласуется с результатами физиологических исследований.

Заключение

Результаты проведенного исследования ясно продемонстрировали, что с помощью модели сердечно-сосудистой системы, реализованной на программе DBOND, можно смоделировать адекватные баллистокардиограммы. Начальной целью при создании модели сердечно-сосудистой системы являлось применение модели в качестве общего исследовательского инструмента; эта цель была достигнута. При разработке модели вопросы моделирования БКГ не рассматривались, но легкость, с которой модель модифицировалась

Кононюк А.Е. Графы

для расчета БКГ, продемонстрировала ее универсальность и физиологический гомоморфизм.

Дальнейшие исследования на этой базовой модели должны быть сосредоточены в двух следующих направлениях: улучшение модели сердца и улучшение представления венозного кровотока. Следует отметить, что ни одна из этих проблем не является проблемой чистого моделирования. Необходимо провести еще значительные физиологические и инженерные исследования для лучшего понимания происходящих явлений. Моделирование баллистокardiограммы в данном случае преследовало цель выяснения возможностей создания такой модели. Успех этой модели указывает направления дальнейших исследований для выявления физиологической значимости индивидуальных БКГ.

1.6. Метод графов связей для описания нелинейных процессов диффузии и химической реакции в однородных или неоднородных средах

В этом разделе рассматривается метод графов связей для описания нелинейных процессов диффузии и химической реакции в однородных или неоднородных средах, предложенный Д. Ауслендером, Ж. Остером, А. Перелсоном и Ж. Клиффордом.

Введение

Взаимосвязь химических реакций и явлений переноса интенсивно изучалась в последние годы, поскольку она имеет большое значение в промышленности и биологии. В самом деле, сложность организации биологических систем в значительной степени обусловлена взаимным влиянием динамики химических реакций, явлений переноса и особенностей структуры биологических молекул

При исследовании явлений, связанных с одновременно протекающими диффузией и реакциями, применялся один из следующих подходов:

а) Химические реакции описывались членами, учитывающими источники и стоки в континуальных или макроскопических уравнениях баланса .

б) Использовался формализм термодинамики необратимых процессов. В первом случае при выборе формы члена, описывающего источник в уравнении реакции, использовали закон действующих масс, употребляемый в химической кинетике. Кроме определенных теоретических недостатков, о которых речь пойдет ниже, этот подход

Кононюк А.Е. Графы

позволяет получить аналитические решения только в случае постоянных коэффициентов переноса и реакций первого порядка. Более сложная кинетика неизбежно требует численного решения, и полученные результаты не обладают большой общностью. Кроме того, во многих системах реакции протекают в чрезвычайно неоднородных средах, например в большинстве биологических реакций катализаторами являются ферменты, связанные со структурными составляющими системы. Решение континуальных уравнений для таких сложных конфигураций, как правило, невозможно. Поэтому исследователи в области биологического переноса, особенно специалисты по биологии мембран, используют формализм термодинамики необратимых процессов. Это позволяет непосредственно исследовать влияние термодинамических связей, которым в химико-технологической литературе придается второстепенное значение. Однако указанный формализм ограничен линейными стационарными системами довольно простого структурного типа.

Оказалось, что методы теории цепей и теории управления можно без особых трудностей применить для описания сложных нелинейных динамических систем. Более того, оказалось, что **электрическую цепь можно рассматривать как необратимую термодинамическую систему**. В настоящей работе предлагается обобщить методы теории цепи таким образом, чтобы можно было учесть нелинейные явления в системах с сопряженными диффузией и химическими реакциями. Для этой задачи естественно использовать язык теории графов связей поскольку при этом все энергетические процессы рассматриваются с единой точки зрения и имеется возможность использовать наглядные представления и обозначения графов для сложных систем с преобразованиями энергии и связями. Графы связей не только служат для описания энергетических и информационных потоков: они позволяют получить алгоритм для вывода уравнений движения и дают удобный способ общения с ЭВМ при моделировании.

Общие положения

Поскольку теория графов связей довольно полно изложена в начале раздела 1.1, в данном примере будут рассмотрены главным образом те аспекты теории, которые относятся к термодинамическим системам.

Приведем данное Шоттки классическое определение **термодинамической системы** как системы взаимодействия которой с окружающей средой сводится только к обмену механической работой и теплотой. На языке теории цепей компонента термодинамической системы представляет собой n -связный элемент, т.

Кононюк А.Е. Графы

е. «черный ящик», состояние которого может быть задано конечным числом внешних параметров, т. е. конечным числом связей с внешней средой. Каждой такой связи отвечает пара сопряженных переменных: переменная типа тока τ_i и переменная типа напряжения α_i . Выбор по крайней мере одного множества $x = (\tau_1, \dots, \tau_k; \alpha_{k+1}, \dots, \alpha_n)$ из n этих величин достаточен для определения сопряженного множества $y = (\alpha_1, \dots, \alpha_k, \tau_{k+1}, \dots, \tau_n)$. Отображение $\tau \rightarrow \alpha$ называется определяющим соотношением n -связного элемента. Например система цилиндр — поршень, показанная на рис.1а, имеет одну тепловую и одну механическую связи характеризуемые термодинамическими параметрами (T, S) и (p, V) соответственно.

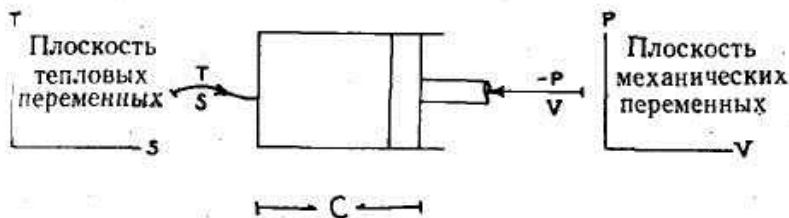


Рис.1а. Термомеханическая 2-связная емкость.

Определяющее соотношение (уравнение состояния) для такой системы обычно задается следующим образом: $(S, V) \rightarrow (T(S, V), -p(S, V))$.

Основная проблема при исследовании термодинамических систем методами теории графов связей заключается в построении алгоритма структуры связей и последующем представлении n -связных элементов в виде набора элементарных «стандартных» 1-, 2- и 3-связных элементов (т. е. $R, L, C, \theta, 1, TD$; подробнее об этом будет сказано ниже).

Прежде чем обсуждать вопрос о соединениях систем из n -связных элементов, необходимо наложить следующие ограничения на переменные (τ_i, α_i) в узлах связей. Во-первых, допускаются соединения только связей одинаковой энергетической природы (механических с механическими, тепловых с тепловыми и т. д., рис.1б).

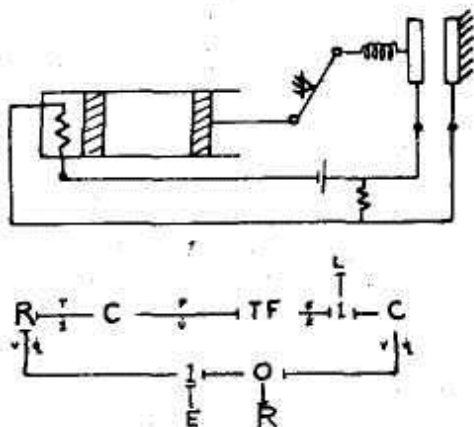


Рис.1б. Соединение тепловых, механических и электрических N-связных элементов.

Во-вторых, на переменные типа тока τ_i накладываются ограничения в виде линейного уравнения сохранения (законы Кирхгофа для токов, ЗКТ) и, в-третьих, на переменные типа напряжения α_i накладываются ограничения в виде линейного уравнения непрерывности (закон Кирхгофа для напряжений, ЗКН). (Целесообразно, хотя и не обязательно, выбирать переменные τ_i , α_i таким образом, чтобы их произведение имело размерность энергии или мощности.)

Термодинамические связи

Каждой связи графа сопоставлены сопряженные переменные типа тока (поток) и типа напряжения (усилие), относящиеся к рассматриваемой энергетической области. В качестве переменных типа тока и типа напряжения выбраны соответственно электрохимический потенциал μ_i и молярный (массовый) поток J_i . Диффузия одного электрически нейтрального вещества через мембрану показана на рис.2, а.

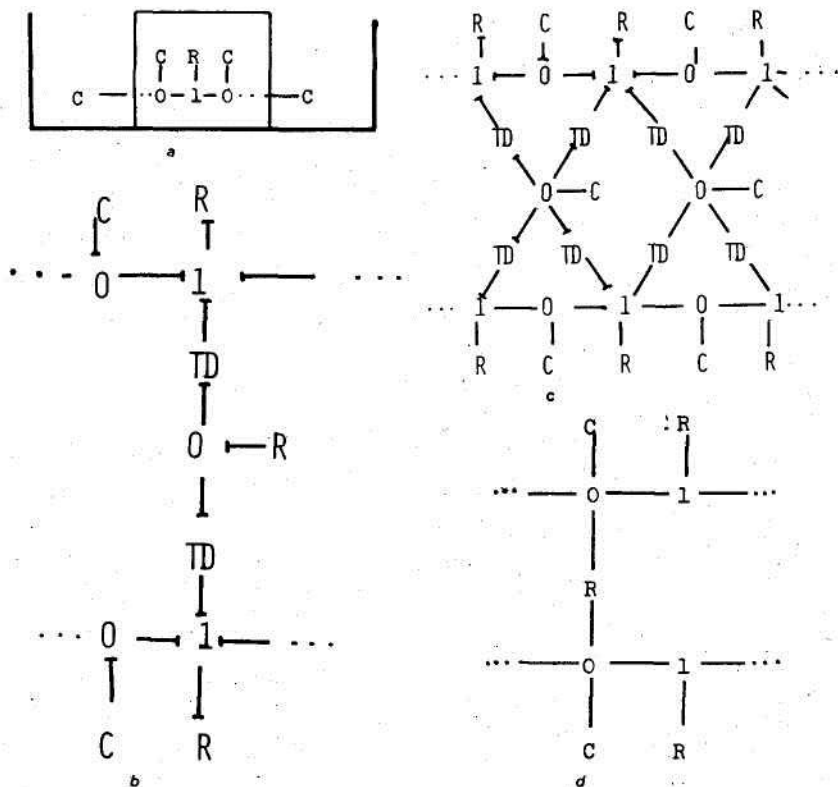


Рис.2. Структуры связей. *a* — простая диффузия; *b* — фрикционная связь; *c* — связи в электрическом поле; *d* — связи реакции.

В приложениях обычно используют три основных типа термодинамических связей: а) непосредственная диффузионная связь, б) электромеханическая связь между диффундирующими ионами и в) связь при реакции.

Диффузионную связь обычно рассматривают как простое гидродинамическое воздействие, т. е. как фрикционное торможение при даффузии одного компонента через другой. Другой тип связи обусловлен действием интенсивных короткодействующих сил, стремящихся обеспечить электронейтральность в электролитических растворах. Третий тип связи возникает при локальных взаимных превращениях компонентов в ходе химической реакции. Во многих случаях все три взаимодействия могут существовать одновременно,

Кононюк А.Е. Графы

что приводит к появлению довольно сложных и неожиданных профилей концентрации и определяет особенности динамики системы. Связи а) и б) представляют собой взаимодействие потоков, зависящее от относительного движения диффундирующих веществ. По этой причине можно описывать каждое из них 2-связным элементом, соединяющим потоки химических компонентов. Структуры графов связей для этих взаимодействий приведены на рис. 2, *b, c*, где 2-связный элемент — TD — обозначает преобразователь графа связей. Эти структуры описаны в литературе и здесь подробно рассматриваться не будут.

Взаимодействие химически реагирующих веществ имеет совершенно другую природу. Это не связь в смысле структур а) и б), а скорее локальный эффект сохранения. Рассмотрим простейший случай реакции $A \rightleftharpoons B$, протекающей в изобаро-изотермических условиях. Вещество *A*, диффундируя и вступая в реакцию, непрерывно превращается в вещество *B* в каждой точке пространства. Реакция сопровождается уменьшением свободной энергии, совершенно не связанным с уменьшением свободной энергии в результате диффузии каждого компонента в направлении градиента химического потенциала. Решетчатую структуру связей при реакции можно представить в виде резистивного 2-связного элемента, соединяющего 0-узлы каждой цепи диффузии, как показано на рис. 2, *d*. Чтобы понять, какая структура графа связей описывает термодинамические аспекты химических реакций, исследуем вначале случай гомогенных реакций без диффузии.

Химические реакции

Термодинамические основы. Рассмотрим однородную (хорошо «перемешанную») систему, в которой протекает единственная реакция при постоянных температуре и давлении. Соответствующим термодинамическим потенциалом для такой системы является свободная энергия Гиббса *G* :

$$dG = \sum_{i=1}^N \mu_i dn_i \quad (T, P \text{ constant}), \quad (1)$$

где μ_i — химический потенциал компонента *i* и n_i — масса (или количество молей) компонента *i*. Условие химического равновесия имеет вид

$$dG = \sum \mu_i dn_i = 0. \quad (2)$$

Кононюк А.Е. Графы

При протекании реакции происходит необратимое уменьшение свободной энергии со скоростью

$$\dot{G} = \sum \mu_i \dot{n}_i. \quad (3)$$

Однако не все n'_i независимы. Условие стехиометрии реакции накладывает следующее ограничение:

$$\frac{\dot{n}_i}{\nu_i} = \frac{\dot{n}_j}{\nu_j} \quad \forall i, j, \quad (4)$$

где стехиометрические коэффициенты реакции ν_i положительны для продуктов и отрицательны для реагентов. Назовем отношение

$$\frac{\dot{n}_i}{\nu_i} \approx J^R \quad (5)$$

скоростью реакции и перепишем уравнение (3) в виде

$$\dot{G} = J^R \sum \nu_i \mu_i. \quad (6)$$

Обозначим

$$- \sum \nu_i \mu_i \approx A \quad (7)$$

и назовем сродством реакции .

Выражение (3) можно теперь записать в виде

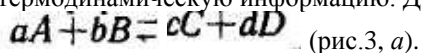
$$- \frac{dG}{dt} = J^R A. \quad (8)$$

Скорость реакции J^R и сродство A являются соответственно переменными типа тока и типа напряжения. Заметим, что условие равновесия (2) принимает вид —

$$A = \sum \nu_i \mu_i = 0.$$

Представление графа связей химической реакции

Определим структуру графа связей, содержащую всю указанную выше термодинамическую информацию. Для простоты рассмотрим реакцию



Кононюк А.Е. Графы

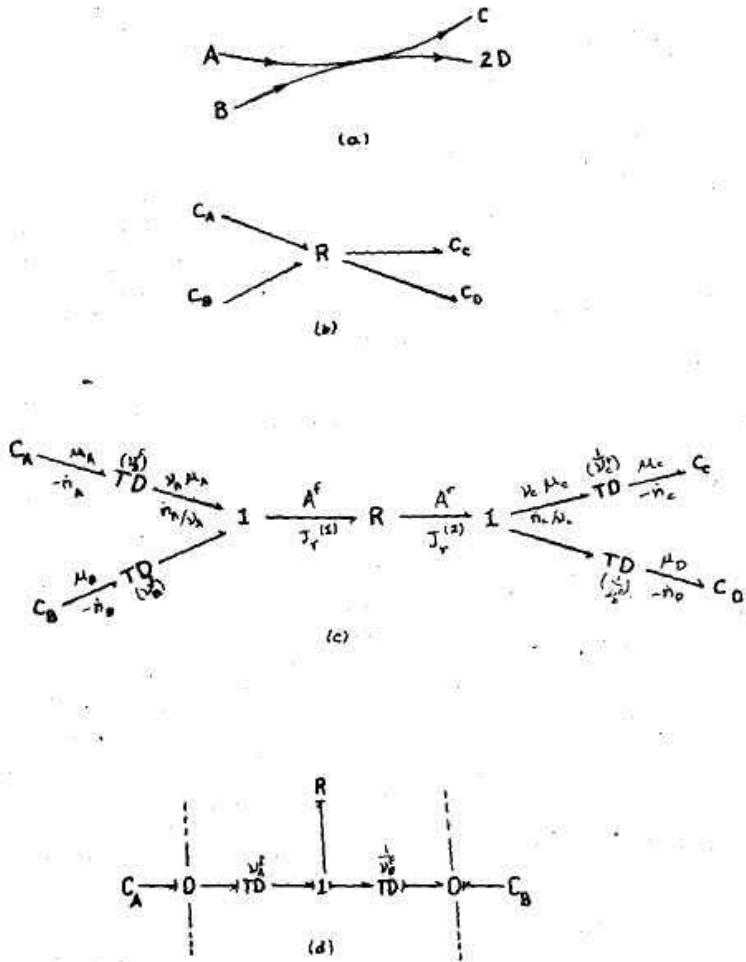


Рис.3.

Как уже отмечалось, с феноменологической точки зрения химическая реакция может быть представлена многообразным полем потерь со входными реагентами и выходными продуктами в стехиометрических пропорциях. На рис.3, *b* эта реакция представлена 4-связным резистивным соединением химических емкостей компонентов *A*, *B*, *C* и *D*. Требование стехиометрии представляет собой незнергическое ограничение, соответствующее линейному преобразованию переменных

Конюк А.Е. Графы

$$(\mu_i, \dot{n}_i) \rightarrow (v_i \mu_i, \dot{n}_i / v_i) \approx (A_i, J^R). \quad (9)$$

Величину

$$v_i \mu_i = A_i$$

иногда называют «парциальным средством», так что

$$A = - \sum A_i. \quad (10)$$

Выражение (10) представляет собой определяющее соотношение для преобразования в графе связей по модулю v_i :

$$\frac{\mu_i}{\dot{n}_i} \text{ TD } \frac{A_i}{J^R}.$$

Вводя преобразователи, как показано на рис.3, с, можно учесть стехиометрию реакции. Диссипация вследствие реакции моделируется резистивным 2-связным элементом, связь которого представляют собой средство прямой и обратной реакций:

$$A_f \approx \sum_{i=1}^N v_i^f \mu_i; \quad (11a)$$

$$A_r \approx \sum_{i=1}^N v_i^r \mu_i, \quad (11b)$$

где v_i^f и v_i^r — прямые и обратные стехиометрические коэффициенты (положительные целые числа);

$$v_i = v_i^f - v_i^r.$$

Интегральное средство A , определенное уравнением (7), запишем в виде

$$A = A_f - A_r. \quad (12)$$

Физическое ограничение, состоящее в том, что для реакции выполняется закон сохранения вещества, требует, чтобы переменные типа тока J^R_1 и J^R_2 , соответствующие 2-связному сопротивлению реакции, были равны, т. е.

Кононюк А.Е. Графы

$$J_1^R = J_2^R.$$

Полная диссипация, соответствующая этому 2-СВЯЗНОМУ элементу, равна

$$A_f J_1^R - A_r J_2^R = (A_f - A_r) J^R = A J^R.$$

Эта диссипация может быть также описана 1-связным резистором с переменными типа тока и типа напряжения (J^R , A). Кроме того, уравнения (10) и (13) оказываются уравнениями 1-узла, налагающего ограничения типа закона Кирхгофа для токов на J^R и закона Кирхгофа для напряжений на A_i . Следовательно, с чисто термодинамической точки зрения можно формально преобразовать 2-связное сопротивление реакции в 1-связный резистор, соединенный с 1-узлом, как показано на рис.3, *d*. Однако при этом преобразовании 2-связного резистора не учитывается один важный аспект кинетики химических реакций. Чтобы убедиться в этом, проанализируем некоторые экспериментальные факты.

Рассмотрим химическую реакцию, находящуюся в неравновесном установившемся состоянии за счет поддержания на постоянном уровне концентрации реагентов и продуктов. Пусть проведены два эксперимента и измерены

$$\left(\frac{\partial J^R}{\partial A_f}\right)_{A_r} \text{ и } \left(\frac{\partial J^R}{\partial A_r}\right)_{A_f}.$$

Тогда

$$\left(\frac{\partial J^R}{\partial A_f}\right)_{A_r} \neq \left(\frac{\partial J^R}{\partial A_r}\right)_{A_f}. \quad (14)$$

Если использовать представление графа связей, показанное на рис.3, *c*, то из этого неравенства можно заключить, что 2-связное сопротивление необратимо и, например, матрица приращений проводимости не симметрична. Однако для представления этого 2-связного сопротивления в виде, показанном на рис.3, *d*, должно существовать единственное соотношение между A и J^R , характеризующее 1-связный резистор. Кроме того, 1-связный элемент всегда имеет тривиальный обратный (определяющее соотношение для приращений — скалярное), а это противоречит условию необратимости. Известно, что все химические реакции, описываемые кинетикой взаимодействия масс, необратимы, но вблизи равновесия структура 1-связного элемента является хорошей аппроксимацией, и только в точке равновесия она в точности описывает процесс.

Кононюк А.Е. Графы

Правило знаков. Выберем знаки для преобразователей (TD) и 2-связного сопротивления реакции таким образом, чтобы представление графа связей как можно точнее соответствовало обозначениям химических реакций, принятым в биохимии (рис.3, а). Различие состоит лишь в том, что граф связей представляет собой также вычислительный алгоритм.

Сопряженные реакции. Эта структура может быть легко распространена на более сложные схемы реакций, в которых компоненты могут принимать участие более чем в одной реакции. На рис. 4 введены соответствующие 0-узлы, моделирующие простую химическую систему с обратной связью.

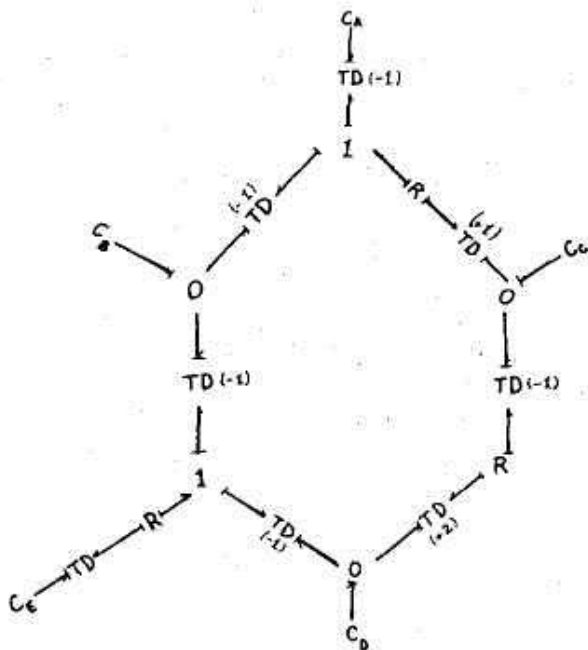


Рис. 4. Представление сопряженных реакций $A+B \rightleftharpoons C$, $C \rightleftharpoons D$, $D+B \rightleftharpoons E$ в виде графов связей.

Сопряженные диффузия и реакция. Принимая, что химический потенциал представляет собой движущую силу как для диффузии, так и для реакции, можно свести структуру реакции к рассмотренной ранее диффузионной цепи. На рис. 5 показан простейший случай диффузии и реакции в мембране.

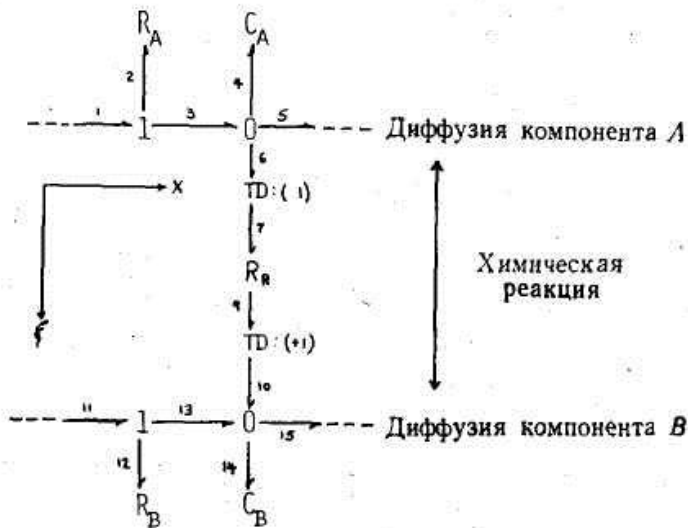


Рис. 5. Сотряженные реакция и диффузия.

Определим смещение реакции ξ следующим образом:

$$\xi(t) \triangleq \xi(0) + \int_0^t J^R dt$$

(Из рис. 5 видно, что пространственную зависимость течения реакции удобно представлять на графике в координатах $\xi-x$.)

Известно, что при линейных определяющих соотношениях цепь диффузии, показанная на рис.2, а, в предельном случае бесконечно большого числа участков описывается обычным уравнением диффузии

$$\partial c / \partial t = -D \nabla^2 c.$$

Это утверждение легко распространить на случай сопряженной диффузии

$$c = -D \nabla^2 c;$$

здесь c — вектор концентрации и D — матрица коэффициентов диффузии. Структура связей (рис. 5) также обладает соответствующим предельным поведением, что приводит к точным уравнениям сопряженных диффузии и реакции.

Кононюк А.Е. Графы

Причинная связь. Одной из наиболее полезных особенностей графа связей как средства моделирования является введенное Пейнтером обозначение так называемого «причинного хода», позволяющего различать входные (независимые) и выходные (зависимые) переменные на энергетических полюсах. Причинная согласованность позволяет произвольно задавать начальные условия для всех динамических элементов, не нарушая теорем Кирхгофа, и в этой ситуации потребовались бы бесконечно большие значения мощности (например, когда две емкости соединяются параллельно).

Присоединим к графу связей, представленному на рис. 3, *d*, обозначения причинного хода. Как и следовало ожидать, сопротивление реакции регулируется движущей силой и задает скорость реакции в зависимости от химического средства.

Системы ферментов. На реакцию $A \rightleftharpoons C$ может оказывать катализирующее влияние фермент *E*, который может быть либо равномерно распределен, либо закреплен на некотором структурном образовании, например на мембране. Поскольку катализатор может изменить не точку равновесия, а только скорость достижения этого равновесия, реакция, катализируемая ферментом, должна описываться шунтом с малым сопротивлением для реакции без катализатора. Типичную реакцию можно записать в виде $A + E = (AE) = E + C$, где (AE) — промежуточный «активированный комплекс». Граф связей для такой системы реакций показан на рис. 6.

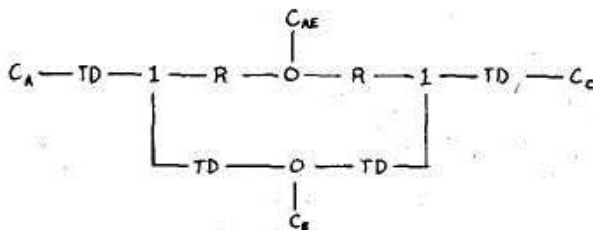


Рис.6. Реакция с катализатором $A + E \rightleftharpoons AE \rightleftharpoons E + C$.

Если фермент может диффундировать или испытывать структурные изменения (например, так называемые аллостерные переходы), то для молекулярных компонентов контур фермента может образовывать «проход» через некоторый диффузионный барьер с малым сопротивлением. В этом и состоит явление «облегченного» переноса, обнаруженное в некоторых биологических системах (например,

Кононюк А.Е. Графы

перенос галактозы через мембрану красной кровяной клетки, перенос кислорода через растворы гемоглобина). Такой механизм переноса носителя, когда фермент либо диффундирует, либо действует как химикомеханический преобразователь, показан на рис. 7.

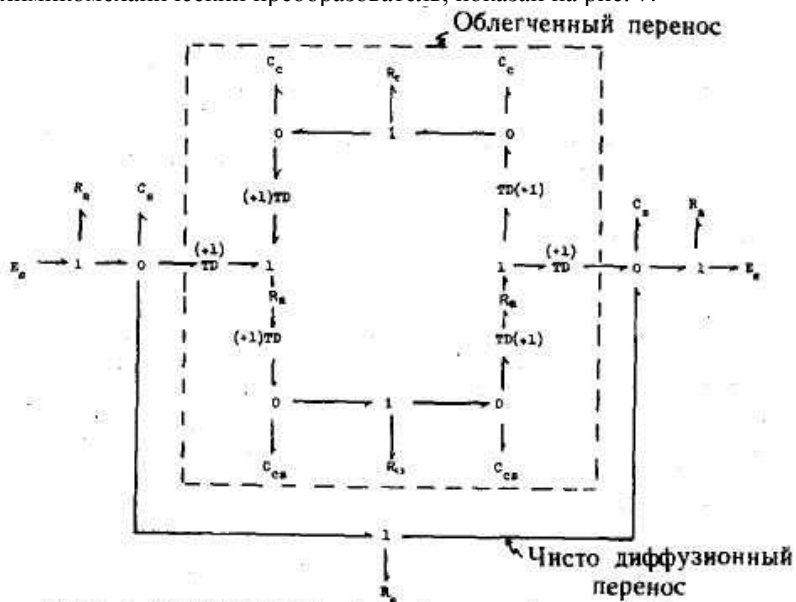


Рис.7. Сопряженные реакции и облегченный перенос. Описание облегченного переноса (обозначения: s — переносимый субстрат, c — носитель, cs — комплекс субстрат — носитель): 1) комплекс субстрат — носитель образуется в левой стороне; 2) cs переносится слева направо за счет изменений структуры или переноса; 3) cs диссоциирует на c и s в правой части; 4) c восстанавливается и получает следующий s .

(Заметим, что в последнем случае координата смещения реакции ξ имеет компоненту вдоль пространственного направления x .)

Анализ систем такого рода здесь не будет представлен. Здесь мы ограничимся реакциями без катализатора.

Определяющие соотношения. *Сопротивление реакции.* Строго говоря, определяющие соотношения следует рассматривать как часть простейших экспериментальных данных, необходимых при моделировании цепи. Тем не менее краткое обсуждение природы

Кононюк А.Е. Графы

определяющих соотношений для графа связей сопряженных реакций и диффузии было бы уместным.

Появление многих идей химической кинетики вызвано успешным применением «закона» действующих масс для реакций, протекающих в газовой фазе. Закон действующих масс имеет вид

$$J^R = v_F - v_R,$$

где v_F и v_R — скорости прямой и обратной реакций соответственно. Непосредственно может быть измерена только суммарная скорость реакции J^R . Представление ее в виде разности двух членов не вызвано требованиями термодинамики: оно имеет смысл только для интерпретации величин в кинетической теории. Для реакций, в которых участвуют разбавленные растворы газов, представление суммарной скорости в таком виде может быть оправдано, если исходить из вероятностной модели столкновений. Например, экспериментально установлено, что большинство реакций, протекающих в газовой фазе, можно приближенно описать уравнением вида

$$J^R = k_f(T) \prod_{\text{реагенты}} \left(\frac{n_i}{V}\right)^{\alpha_i} - k_r(T) \prod_{\text{продукты}} \left(\frac{n_j}{V}\right)^{\beta_j}, \quad (15)$$

де α_i и β_j — малые положительные числа, называемые порядком реакции (иногда они равны стехиометрическим коэффициентам), и n_i/V — молярная концентрация. Такое кинетическое соотношение, естественно, не укладывается в рамки теории графов связей, поскольку поток J^R нельзя выразить в виде функции некоторой величины, имеющей смысл потенциала, т. е. соотношение (15) является скорее «кинематическим», нежели динамическим.

Для случаев реакций в растворе, более характерных для биологии, начинают действовать другие факторы, которые приводят к тому, что закон действующих масс не выполняется; в первую очередь к таким факторам относятся молекулярные силы и эффекты раствора. До сих пор не существует удовлетворительной теории скорости реакций в растворе, поэтому немало изобретательности было потрачено на то, чтобы приближенно описать взаимодействие масс в неразбавленных газах. Вводимые при этом эмпирические поправки должны удовлетворять следующему требованию: скорость реакции должна быть положительной в направлении убывания свободной энергии и обращаться в нуль в точке равновесия. Обычно используют выражение вида

Кононюк А.Е. Графы

$$J^R = \zeta \left\{ k_r(T) \prod [\gamma_i]^{\alpha_i} \left[\frac{n_i}{V} \right]^{\alpha_i} - k_r(T) \prod [\gamma_j]^{\beta_j} \left[\frac{n_j}{V} \right]^{\beta_j} \right\},$$

где γ_i — коэффициент активности компонента i и ζ — некоторая функция концентраций n_i/V .

С точки зрения моделирования динамической системы, по-видимому, мало оснований стремиться к выполнению закона действующих масс, поскольку для реакций в растворах скорость не пропорциональна полиному от объемной концентрации реагентов. Можно также рассматривать определяющее соотношение между скоростью реакции и сродством прямой и обратной реакции как исходные экспериментальные данные и придавать этому соотношению такую экспериментальную форму, которую диктует конкретная реакция.

В общем случае сформулированным ранее термодинамическим требованиям можно удовлетворить, используя определяющее соотношение вида:

$$J^R = \theta \left(\frac{n_i}{v} \right) \cdot A, \tag{16}$$

где θ — любая положительная функция концентраций. В приложении показано, что уравнение (15) для скорости реакции эквивалентно следующему термодинамическому соотношению:

$$J^R = v_F (1 - e^{-A/RT}).$$

Однако, поскольку условие необратимости не может быть выполнено в уравнении, подобном соотношению (16), приходится использовать представление в виде 2-связного элемента. Определяющее соотношение имеет вид

$$\begin{aligned} J_1^R &= \Phi_1(A_f, A_r) \\ J_2^R &= \Phi_2(A_f, A_r), \end{aligned} \tag{17}$$

где Φ_1 и Φ_2 должны быть одной и той же функцией в силу закона сохранения масс. В приложении показано, что модель действующих масс можно аппроксимировать выражениями вида

$$\begin{aligned} J_1^R &= \kappa (e^{A_f/RT} - e^{A_r/RT}) \\ J_2^R &= \kappa (e^{A_f/RT} - e^{A_r/RT}), \end{aligned} \tag{18}$$

где κ — неотрицательная константа.

Кононюк А.Е. Графы

Сопrotивление диффузии. Перенос массы посредством диффузии можно феноменологически описать при помощи первого закона Фика:

$$J_s^d = - D_s \frac{dc_s}{dx}, \quad (19)$$

где J_s^d — поток вещества и D_s — коэффициент диффузии компонента s . Однако с точки зрения термодинамики движущей силой для диффузии является градиент химического потенциала, а не градиент концентрации. Для простоты рассмотрим идеальный раствор, т. е.

$$\mu_s = \mu_s^\circ + RT \ln c_s \quad (20)$$

и, следовательно,

$$J_s^d = - \frac{D_s c_s}{RT} \frac{d\mu_s}{dx} \approx \quad (21)$$

$$\approx \left(\frac{D_s c_{s,l}}{RT w_l} \right) \Delta\mu_s, \quad (22)$$

где $\Delta\mu_s$ — уменьшение химического потенциала компонента s на участке l шириной w_l . Сопrotивление диффузии $R_{s,l}^d$ для компонента s в этой области равно

$$R_{s,l}^d = \frac{RT w_l}{D_s C_{s,l}}. \quad (23)$$

Химическая емкость. Соотношение, подобное уравнению (20), представляет собой определяющее соотношение для химических емкостей в графе связей диффузии и реакции. В общем случае определяющее соотношение для емкости имеет вид $n_s = \varphi(\mu_s)$. После дифференцирования получаем

$$\frac{dn_s}{dt} = \frac{\partial \varphi(\mu_s)}{\partial \mu_s} \frac{d\mu_s}{dt}. \quad (24)$$

Обозначим

$$[\partial_\varphi / \partial \mu_s](\mu_s) = C_s.$$

Тогда уравнение (24) можно привести к

$$f_s = C_s(\mu_s) \mu_s, \quad (25)$$

где $f_s = dn_s/dt$ — поток, поступающий в емкость.

Моделирование систем диффузии— реакции

Модель. Рассмотрим мембрану, имеющую единичную площадь и толщину t и расположенную между двумя бесконечными резервуарами, которые содержат $N-q$ химических компонентов с фиксированными концентрациями. Предположим, что в этих резервуарах не происходит химических реакций и что каждый из $N-q$ компонентов может проникать через мембрану. Предположим также, что имеются q химических компонентов (называемых молекулами носителя) $q \geq 0$, которые могут диффундировать внутри мембраны, но не переходят через ее границы. Предположим также, что внутри мембраны возникают M химических реакций и что любая стехиометрическая комбинация N имеющихся компонентов может вступать в реакцию. Эта мембрана может быть представлена графом связей диффузии и реакции, состоящим из L участков каждый шириной w_l , $l=1, \dots, L$. Чтобы рассматривать системы со свойствами, изменяющимися в пространстве, мы допускаем, что w_l и значения параметров R и C графа связей меняются от участка к участку. В тех случаях, о которых здесь будет идти речь, на каждом участке протекала одна и та же реакция, но это ограничение не обязательно. Для иллюстрации рассмотрим модель графа связей, приведенную на рис. 8 и соответствующую системе с компонентами A и B , участвующими в одной реакции.

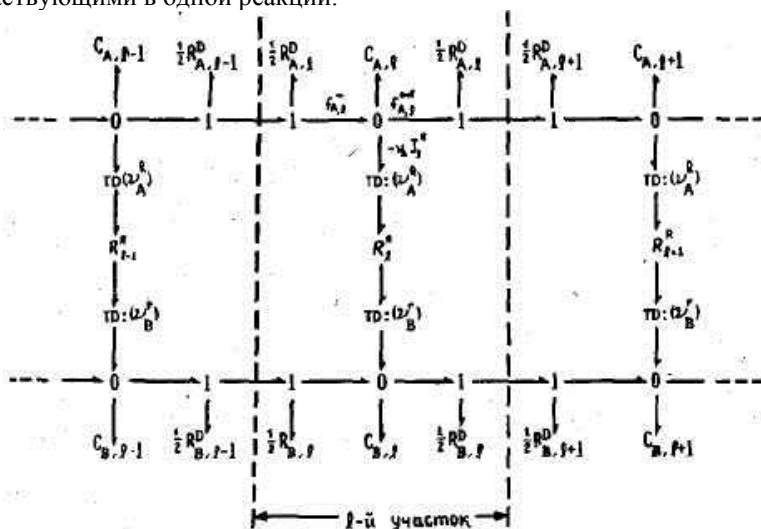


Рис. 8

Кононюк А.Е. Графы

Если бы были и другие компоненты, то каждой из них соответствовала бы своя R — S -цепочка диффузии, связанная на каждом участке через преобразователи с каждой реакцией, в которой она участвует.

Симметричная структура участка была выбрана для того, чтобы каждая цепочка диффузии начиналась и заканчивалась сопротивлением. Это позволяет избежать трудностей, связанных с причинностью, на границах мембраны (рис. 9).

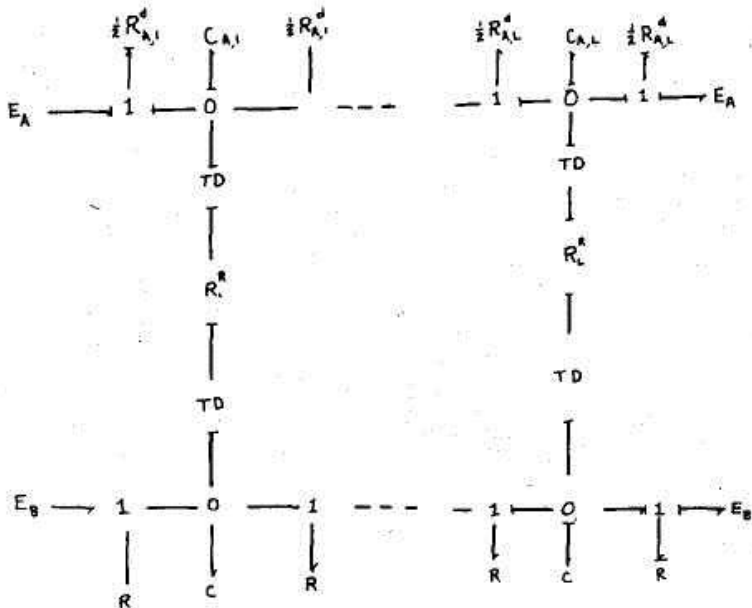


Рис. 9.

Полное сопротивление диффузии для компонента s на l -м участке поровну распределяется между двумя диффузионными сопротивлениями на каждом симметричном участке. Из уравнения (23) следует

$$R_{s,l}^d = \frac{RTw_l}{D_s c_{s,l}} \quad (26)$$

где w_l — ширина участка l в см и $c_{s,l}$ — концентрация компонента s на участке l . Для молекул-носителей, не проникающих через границы мембраны, левое диффузионное сопротивление на участке l и правое диффузионное сопротивление на участке l бесконечно велики, что

Кононюк А.Е. Графы

препятствует потоку молекул-носителей через границы системы. (При вычислениях необходимо только выбирать определяющее соотношение для граничного сопротивления таким образом, чтобы оно было на несколько порядков больше, чем внутреннее сопротивление.) Как видно из рис. 8, при таком выборе симметричной структуры участка между каждой парой емкостей располагается пара сопротивлений. Следовательно, диффузионные потоки в нулевой узел из нулевого узла, с которым соединяется емкость $C_{s,l}$ равны

$$f_{s,l}^{\text{in}} = \frac{\mu_{s,l-1} - \mu_{s,l}}{1/2R_{s,l-1}^d + 1/2R_{s,l}^d}; \quad (27)$$

$$f_{s,l}^{\text{out}} = \frac{\mu_{s,l} - \mu_{s,l+1}}{1/2R_{s,l}^d + 1/2R_{s,l+1}^d}. \quad (28)$$

Уравнения состояния. Динамические уравнения системы, описываемой графом связей, можно вывести из определяющих соотношений для элементов и законов Кирхгофа. Поскольку, согласно предположению, эффекты конвекции пренебрежимо малы, граф связей реакции и диффузии не содержит индуктивностей. Следовательно, уравнения состояния можно записать только через потенциалы емкостей. Для емкости $C_{s,l}$ из уравнения (25) и закона Кирхгофа получаем

$$f_{s,l} = f_{s,l}^{\text{in}} - f_{s,l}^{\text{out}} + \sum_{r=1}^M \nu_{s,r} J_{r,l}^R, \quad (29)$$

где $\nu_{s,r}$ — стехиометрический коэффициент компонента s в реакции r и $J_{r,l}^R$ — скорость реакции r на участке l . Подставляя уравнения (11), (17), (25), (27), и (28) в уравнение (29), получаем уравнения состояния

$$\dot{\mu}_{s,l} = \frac{1}{C_{s,l}} \left[\frac{\mu_{s,l-1} - \mu_{s,l}}{1/2R_{s,l-1}^d + 1/2R_{s,l}^d} - \frac{\mu_{s,l} - \mu_{s,l+1}}{1/2R_{s,l}^d + 1/2R_{s,l+1}^d} \right] +$$

$$+ \sum_{r=1}^M \nu_{s,r} \phi \left(\begin{array}{cc} -\sum_{\text{react.}} \nu_{s,r} \mu_s & \sum_{\text{prod.}} \nu_{s,r} \mu_s \end{array} \right) \quad s = 1, \dots, N; \quad (30)$$

$$l = 1, \dots, L.$$

Распределение химического потенциала в установившемся состоянии. Полагая $\dot{\mu}'_{s,l}=0$ в уравнении (30), получим систему нелинейных алгебраических уравнений, определяющих распределение химического потенциала в установившемся состоянии. Использование определяющего соотношения для емкости [см. уравнение (20)

Кононюк А.Е. Графы

позволяет перейти от распределения потенциала к распределению концентрации.

Программа для ЦВМ. Была разработана стандартная программа, позволяющая моделировать граф связей диффузии и реакции с произвольным числом реагентов и реакций. Для простоты использовался закон действующих масс, что позволяло характеризовать реакции константами скорости реакции в прямом и обратном направлениях. Однако при желании можно использовать любые другие подходящие уравнения скорости реакции. Для каждого химического компонента следует задать его концентрацию в резервуаре, коэффициент диффузии, стехиометрические коэффициенты и указать, может ли он проходить через мембрану.

Наша главная цель заключалась в разработке программ, которые можно было бы эффективно и легко использовать для вычисления установившейся концентрации и распределения химического потенциала. Динамические характеристики были получены из уравнения (30) при помощи стандартных методов численного интегрирования. В случае установившегося состояния требуется решить систему $N \times L$ нелинейных алгебраических уравнений. Любое множество химических потенциалов, удовлетворяющее уравнению

$$f_{s,l}^{in} - f_{s,l}^{out} + \sum_{r=1}^M \nu_{r,s} J_{r,l}^R = 0 \quad (31)$$

(где $f_{s,l}^{in}$ и $f_{s,l}^{out}$ определяются уравнениями (27) и (28), а $J_{r,l}^R$ — законом действующих масс), является решением для установившегося состояния при всех значениях $s=1, \dots, N, l=1, \dots, L$.

Следовательно, можно методом последовательных приближений решить аппроксимирующую систему линейных алгебраических уравнений, пока не будет найдено множество химических потенциалов, удовлетворяющих уравнению (31). Хотя сопротивление реакции не является односвязным элементом, можно для итеративной схемы аппроксимировать $J_{r,l}^R$ выражением

$$J_{r,l}^R \approx \frac{A_{r,l}}{R_{r,l}^R} + J_{r,l}^{R,0}, \quad (32)$$

где $J_{r,l}^{R,0}$ — скорость реакции r при $A_{r,l}=0$ и где $R_{r,l}^R$ — приращение сопротивления реакции.

Подставляя уравнения (27), (28) и (32) в уравнение (31), получим систему нелинейных алгебраических уравнений в виде, удобном для вычислений

Кононюк А.Е. Графы

$$\mathbf{Q}(\boldsymbol{\mu})\boldsymbol{\mu} = \mathbf{F}(\boldsymbol{\mu}), \quad (33)$$

где $\boldsymbol{\mu}$ — $NL \times 1$ -вектор химических потенциалов каждого компонента на каждом участке,

$\boldsymbol{\mu} = (\mu_{1,1}, \dots, \mu_{N,1}, \mu_{1,2}, \dots, \mu_{N,2})^T$, \mathbf{Q} — $NL \times NL$ -матрица

коэффициентов, зависящая от $\boldsymbol{\mu}$, поскольку сопротивление реакции и диффузии зависит от концентрации, и \mathbf{F} — $NL \times 1$ -вектор движущих сил, содержаний химических потенциалов и потоки реакций при

нулевом сродстве $J_{r,l}^{R,0}$. Матрица \mathbf{Q} симметрична и содержит ненулевые элементы в N -наддиагоналях и N -поддиагоналях. Приведем пример использования уравнения (33) в случае реакции $A \rightleftharpoons B$.

Уравнения для участка l , где $1 < l < L$, имеют вид

$$\begin{aligned} & \left[\frac{2}{R_{A,i-1}^d + R_{A,l}^d} \right] \mu_{A,l-1} - \\ & - \left[\frac{2}{R_{A,l-1}^d + R_{A,l}^d} + \frac{2}{R_{A,l}^d + R_{A,l+1}^d} + \frac{\nu_A^2}{R_l^R} \right] \mu_{A,l} - \\ & - \left[\frac{\nu_A \nu_B}{R_l^R} \right] \mu_{B,l} + \left[\frac{2}{R_{A,l}^d + R_{A,l+1}^d} \right] \mu_{A,l+1} = -\nu_A J_{r,l}^{R,0} \end{aligned}$$

Аналогичное уравнение выполняется для компонента B .

При $l=1$ члены, соответствующие $\mu_{A,l-1}$ и $\mu_{B,l-1}$, располагаются в правой части уравнения, а $\mu_{A,l-1}$ и $\mu_{B,l-1}$ заменяются постоянными значениями для левого резервуара. При $l=L$ аналогичная процедура выполняется над членами, соответствующими $\mu_{A,l+1}$ и $\mu_{B,l+1}$.

Уравнение (33) разрешается итеративно путем повторяющейся оценки $\mathbf{Q}(\boldsymbol{\mu})$ и $\mathbf{F}(\boldsymbol{\mu})$ с использованием в качестве первого приближения линейного распределения концентрации и решения системы линейных уравнений

$$\mathbf{Q}\boldsymbol{\mu} = \mathbf{F} \quad (34)$$

методом исключения Гаусса. Каждое решение уравнения (34) подставляется затем в уравнение (31) для проверки выполнения критерия стационарности. Уравнение (33) в действительности не решается; оно используется только для построения последовательности кривых распределения, сходящейся к установившимся распределениям.

Кононюк А.Е. Графы

Для систем с числом уравнений, меньших 100, этот метод решения оказался удовлетворительным. Для систем с большим числом уравнений можно использовать особенности структуры симметричной матрицы и сократить либо используемую оперативную память, либо время счета в зависимости от метода решения линейных уравнений.

Результаты. Для проверки достоверности модели графа связей и вычислительной схемы сравним вычисленные установившиеся распределения для реакции $A \rightleftharpoons B$ с точными аналитическими решениями континуальных уравнений:

$$D_A \frac{d^2 c_A}{dx^2} = -k_f c_A + k_r c_B;$$

$$D_B \frac{d^2 c_B}{dx^2} = k_f c_A - k_r c_B.$$

Обозначив концентрации компонентов A и B на левой границе мембраны через c_A^0 и c_B^0 и концентрации A и B на правой границе через c_A^1 и c_B^1 соответственно, найдем

$$c_A(X) = \frac{\eta_1 \operatorname{sh}(hX) + \eta_0 \operatorname{sh}(h(1-X))}{(\gamma + K_{eq}) \operatorname{sh}(h)} + \frac{c_1 X + c_0(1-X)}{\gamma + K_{eq}}, \quad (35)$$

где K_{eq} — константа равновесия, X — нормированная переменная смещения, т.е. $X=x/t$, t — толщина мембраны, γ — отношение коэффициентов диффузии D_A/D_B ; h — модуль Тиле,

$k = t [(k_f/D_A) + (k_r/D_B)]^{1/2}$, $\eta_1 = K_{eq} c_A^1 - c_B^1$, $\eta_0 = K_{eq} c_A^0 - c_B^0$,
 $c_1 = \gamma c_A^1 + c_B^1$ и $c_0 = \gamma c_A^0 + c_B^0$ — константы. Поскольку граф связей представляет собой конечномерное приближение к континуальным уравнениям, получаемые распределения концентрации приближаются к кривым, описываемым уравнением (35). Для случая, когда $D_A=D_B=2,0 \cdot 10^{-5}$, $k_f=k_r=10^{-3}$, $t=1,0$, модуль Тиле $h=10$, распределения концентрации, удовлетворяющие уравнению (35), построены на рис. 10 для двух систем граничных условий $c_A^0=1,0$, $c_A^1=1,0$, $c_B^0=1,5$, $c_B^1=1,5$ и $c_A^0=1,0$, $c_A^1=1,0$, $c_B^0=1,1$, $c_B^1=1,1$.



Рис.100. Распределение концентраций.

-----концентрация компонентов В; _____ концентрация компонентов А; случай I: отношение концентраций на границе 1,5 : 1; случай II: отношение концентраций на границе 1.1 : 1.

Программа для сопряженных реакции и диффузии, описанная в разделе «Программа для ЦВМ», вычисляет распределения концентрации с любой степенью точности. На рис.11 построен график зависимости логарифма относительной ошибки

$$\frac{|c_A(\text{аналитическое}) - c_A(\text{граф связей})|}{c_A(\text{аналитическое})}$$

вычисленной на первом участке и в средней точке графа связей, от общего числа участков в графе связей для двух систем граничных условий, указанных выше.

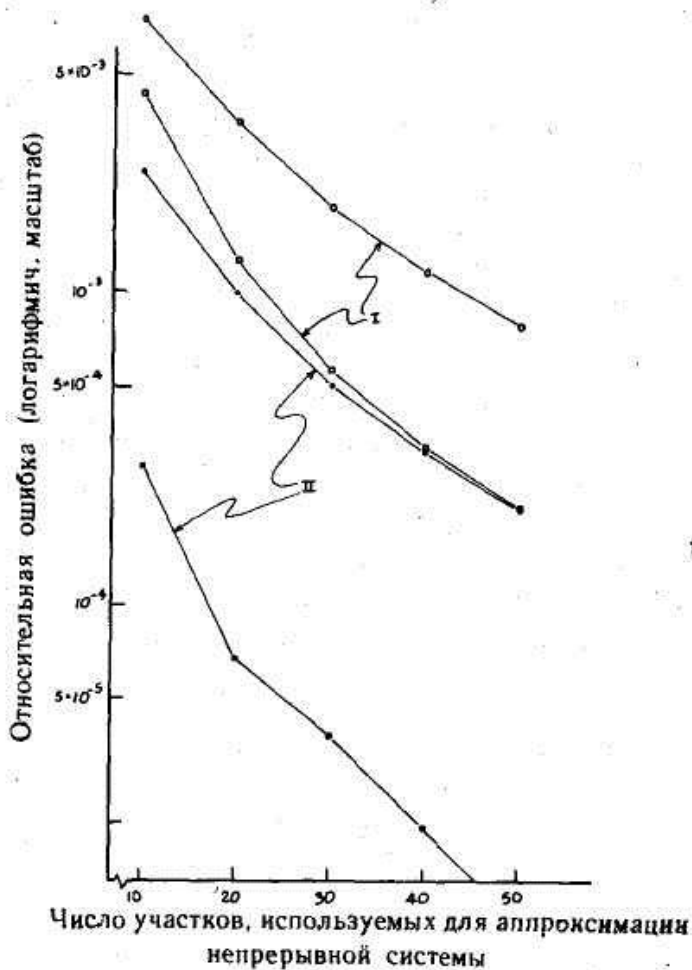


Рис.11. Относительная ошибка дискретизации.

Случаи I и II те же, что и на рис.10; ○ ошибки на участке № 1;
□ ошибки в средней точке.

Относительная ошибка максимальна на первом участке, где концентрация быстро изменяется от граничного значения и значительно меньше в средней точке графа связей, где градиент концентрации меньше.

Кононюк А.Е. Графы

Облегченный перенос кислорода. В качестве более реалистического теста программы для сопряженных реакции и диффузии было выбрано моделирование переноса кислорода через слой раствора гемоглобина. Эта система изучалась неоднократно и экспериментально, и теоретически. Наличие гемоглобина, имеющегося в системе, оказывает такое же действие, как и молекула носителя: было показано, что он существенно увеличивает установившийся диффузионный поток кислорода. Как видно из показанной на рис. 7 модели облегченного переноса с двумя участками, соединяясь с кислородом, гемоглобин создает транспортный контур, параллельный чисто диффузионному переносу кислорода. Хотя коэффициент диффузии кислорода уменьшается в присутствии гемоглобина, параллельная диффузия оксигемоглобина приводит в итоге к увеличению скорости переноса кислорода через слой. Физиологическое значение этого эффекта в красных кровяных клетках в настоящее время активно исследуется. Вычисления Молла наводят на мысль, что диффузия оксигемоглобина увеличивает скорость поглощения и выделения кислорода красными клетками. Облегченный перенос имеет также большое значение при доставке кислорода к клеткам мускулов .

Реакция $\text{Hb} + \text{O}_2 \rightleftharpoons \text{HbO}_2$ и одновременная диффузия гемоглобина Hb, оксигемоглобина HbO₂ и кислорода O₂ изучались при помощи программы для сопряженных реакции и диффузии для слоев различной толщины. Предполагалось, что Hb и HbO₂ не проходят границы системы и в то же время вдоль раствора приложен постоянный градиент давления O₂. Поскольку Hb и HbO₂ задерживаются внутри слоя, решение стационарной задачи сопряженных реакций и диффузии зависит от общего количества гемоглобина в системе. Однако поскольку итеративная схема, описанная в разделе «Распределение потенциала и концентрации», не отслеживает траекторию системы от начальных до конечных состояний, необходимо в качестве дополнительного ограничения использовать условие сохранения гемоглобина. После решения уравнения (34) концентрации и химические потенциалы Hb и HbO₂ на каждом участке были нормированы так, чтобы общее количество гемоглобина в системе равнялось некоторой наперед заданной величине.

На рис.12а и рис.12б показано распределение установившейся концентрации O₂ и HbO₂ для растворов гемоглобина переменной толщины.

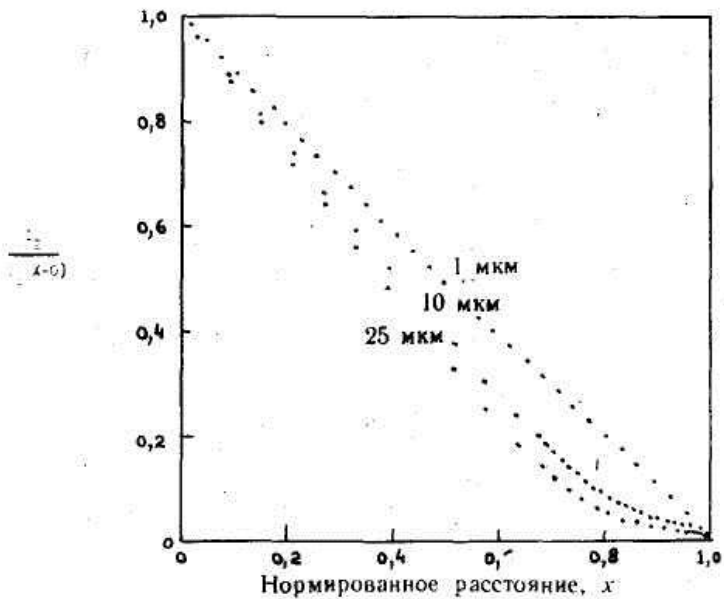


Рис.12а. Распределение концентрации O_2 . (Каждая точка соответствует значению, вычисленному в центре участка.)

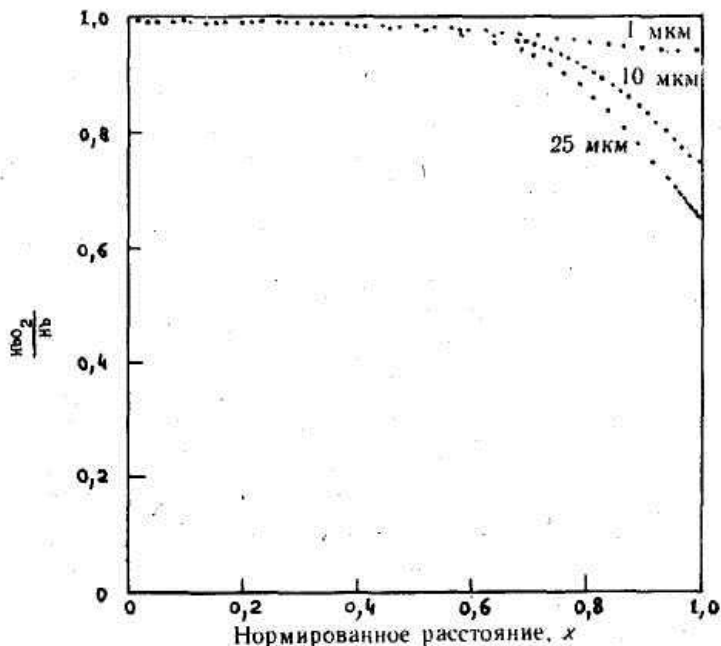


Рис.12б. Распределение концентрации HbO_2 . (Каждая точка соответствует значению, вычисленному в центре участка.)

Поскольку экспериментальные данные о распределении концентрации отсутствуют, мы сопоставили свои результаты с расчетными данными приведенными в ряде работ, в которых использовался метод квазилинеаризации. Хотя мы не столкнулись с проблемой сходимости, наш метод требовал использования очень тонких участков в правой части системы. В случае растворов гемоглобина толщиной 10^{-2} см и больше 33 участков было недостаточно для получения распределения. Размеры красных клеток лежат в диапазоне 10^{-4} — 10^{-3} см.

О влиянии гемоглобина на перенос кислорода можно судить по таблице, где приведены данные о диффузионном потоке кислорода в отсутствие гемоглобина с суммарным потоком кислорода в присутствии гемоглобина.

Кононюк А.Е. Графы

Увеличение потока кислорода в присутствии гемоглобина

Толщина слоя раствора гемоглобина	Суммарный поток кислорода	
	чисто диффузионный поток кислорода	
1	1,05	
10	1,24	
25	1,34	

Во всех случаях гемоглобин увеличивает перенос кислорода, причем большее увеличение соответствует более толстым слоям раствора.

Заключение

Показано, что методом графов связей очень удобно пользоваться при исследовании неравновесных термодинамических систем. Схемы химических реакций произвольной сложности можно описать непосредственно; они могут быть сопряжены с другими термодинамическими процессами, например с процессом диффузионного переноса. Представление с помощью графа связей позволяет естественным образом обобщить классический формализм термостатики на нелинейные динамические процессы, причем это обобщение математически строго и интуитивно оправданно. Более того, модель графа связей может учитывать все существенные феноменологические аспекты химических реакций, а также позволяет включать в модель различные предположения о структуре.

Достоверность модели графов связей проверялась на хорошо исследованной системе облегченной диффузии кислорода в присутствии гемоглобина. Результаты расчетов показывают, что такая модель позволяет достоверно описывать физические системы.

ПРИЛОЖЕНИЕ

Определяющее соотношение „действующих масс“ для химических реакций

а) Пусть скорость реакции

$$I_r = (1/\nu_i) (dn_i/dt) \quad (A1)$$

и сродство

$$A = - \sum_1^N \nu_i \mu_i \quad (A2)$$

Конюк А.Е. Графы

тогда

$$AJ_r = - \sum_1^N \mu_i \frac{dn_i}{dt} = - \frac{dG}{dt}. \quad (A3)$$

Пусть, далее,

$$v_F = k_f \prod_{i=1}^N a_i^{\nu_i^f}, \quad a_i \text{ — активность} = e^{(\mu_i - \mu_i^0)/RT},$$

$$v_R = k_r \prod_{i=1}^N a_i^{\nu_i^r}, \quad N \text{ — общее число компонентов.} \quad (A4)$$

$$\text{Отсюда при } J^R = v_F - v_R \text{ имеем} \quad (A5)$$

$$= v_F \left(1 - \frac{v_R}{v_F} \right) = v_F \left(1 - \frac{k_r \prod a_i^{\nu_i^r}}{k_f \prod a_i^{\nu_i^f}} \right) \quad (A6)$$

$$= v_F \left(1 - \frac{k_r}{k_f} \prod a_i^{\nu_i} \right), \quad \text{где } \nu_i = \nu_i^r - \nu_i^f$$

$$= v_F \left(1 - \frac{k_r}{k_f} \prod (e^{(\mu_i - \mu_i^0)/RT})^{\nu_i} \right)$$

$$= v_F \left(1 - \frac{k_r}{k_f} \prod e^{-\nu_i \mu_i^0 / RT} \prod e^{\nu_i \mu_i / RT} \right)$$

$$= v_F \left(1 - \frac{k_r}{k_f} e^{-\sum \nu_i \mu_i^0 / RT} e^{\sum \nu_i \mu_i / RT} \right)$$

$$= v_F \left(1 - \frac{k_r}{k_f} e^{-\Delta G^0 / RT} \cdot e^{-\Delta I / RT} \right). \quad (A7)$$

Определим константу равновесия

$$K_{eq} = e^{-\Delta G^0 / RT} \quad (A8)$$

$$K_{eq} = (k_f / k_r)_{equil}. \quad (A.9)$$

$$J^R = v_F (1 - e^{-\Delta I / RT}) = \phi(A, a_i). \quad (A.10)$$

б) Определяющее соотношение для 2-связного элемента имеет вид

$$v_F = k_f \prod e^{\nu_i^f (\mu_i - \mu_i^0) / RT} = k_f e^{-\sum \nu_i^f \mu_i^0 / RT} e^{\sum \nu_i^f \mu_i / RT}. \quad (A.11)$$

Пусть

$$A_f \approx \sum \nu_i^f \mu \quad (A.12)$$

$$A_r \approx \sum \nu_i^r \mu_i. \quad (A.13)$$

Кононюк А.Е. Графы

Тогда, подставляя уравнение (A.12) в уравнение (A.11) и уравнение (A.11) в уравнение (A.10), получим

$$J^R = k_f e^{-\sum_i \nu_i q_i / RT} (e^{A_f / RT} - e^{A_r / RT}) \approx \kappa (e^{A_f / RT} - e^{A_r / RT}) \quad (\text{A.14})$$

$$J^R = \phi(A_f, A_r), \quad (\text{A.15})$$

1.7. Мемристор - новый элемент графов связей

Введенный первоначально Чжуа элемент электрических цепей «мемристор» рассматривается как новый, равноправный с R, L и C, элемент графов связей, обладающий рядом уникальных свойств для моделирования нелинейных систем.

Недостающее соотношение

В своих трудах, посвященных методу графов связей, Пэйнтер изобразил «тетраэдр состояний» (рис.1), характеризующий связи между переменными состояниями e, f, p, q .

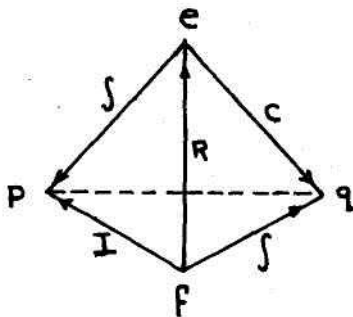


Рис.1. Связи переменных состояний и характеристических соотношений («тетраэдр состояний» Пэйнтера).

Из шести возможных соотношений между любыми двумя переменными первые два являются определениями: перемещение

$$q(t) = q(0) + \int_0^t f(t) dt$$

и величина

Кононюк А.Е. Графы

$$p(t) = p(0) + \int_0^t e(t) dt,$$

тракуемая как количество движения, магнитный поток или «момент —давление» . Следующие три соотношения описывают элементарные свойства диссипативных и накапливающих энергию элементов:

$$F_C(e, q) = 0; \quad (1a)$$

$$F_I(p, f) = 0; \quad (1b)$$

$$F_R(e, f) = 0. \quad (1c)$$

Возникает вопрос, что же представляет собой недостающее шестое соотношение, которое Пэйнтер изобразил штриховой линией на рис.1. Видимо можно предположить, что это соотношение не менее «фундаментально», чем три выше приведенных.

Чжуа предложил назвать шестое соотношение

$$F_m(q, p) = 0 \quad (2)$$

«мемристором», так как подобный элемент «запоминает» полный поток (q) и общее приложенное усилие (p).

Чем же мемристор отличается от остальных основных элементов? Каковы его свойства и что дает его модель? Чжуа указал несколько примеров использования его в теории электрических цепей, однако нетрудно обнаружить ряд систем вне этой области, модель которых достаточно просто представляется мемристором.

Свойства

Основной характеристикой однополюсного мемристора служит кривая на p — q -плоскости (рис.13.65).

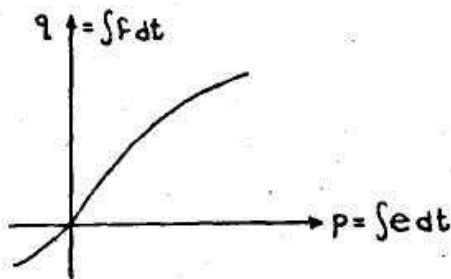


Рис.2. Характеристика мемристора

Кононюк А.Е. Графы

В дальнейшем величину

$$p(t) = p(0) + \int_0^t e(t) dt$$

будем интерпретировать просто как полное усилие («импульс»).

В зависимости от того, как управляется мемристор — по заряду или импульсу, — его характеристику можно представить в виде

$$q = F(p) \text{ (управление по импульсу),} \quad (3a)$$

либо

$$p = G(q) \text{ (управление по заряду),} \quad (3b)$$

Дифференция (3a) и (3b), получим

$$\dot{q} = F'(p)\dot{p} \text{ или } f = W(p)e; \quad (4a)$$

$$\dot{p} = G'(q)\dot{q} \text{ или } e = M(q)f, \quad (4b)$$

где $M(q)$ — элементарное «сопротивление» мемристора, а $W(p)$ — его элементарная «проводимость». Таким образом, с динамической точки зрения **мемристор является сопротивлением, управляемым по импульсу или заряду**. В частном случае, когда характеристика линейна: $M = \text{const}$ и $W = \text{const}$, моделью мемристора служит обычное сопротивление. **Мемристоры поэтому имеют смысл только для нелинейных систем** — этим можно объяснить, в частности, почему они не были введены раньше. Кроме того, из рассмотрения «тетраэдра состояний» (рис.1) можно сделать вывод, что, поскольку при трактовании мемристора как «сопротивления» допускается и интегральное, и дифференциальное представление (на e — f -плоскости), мемристор, как и обычное сопротивление, с точки зрения причинности является нейтральным элементом. **Входным сигналом его может быть как усилие, так и поток**. Однако при моделировании систем могут возникнуть некоторые ограничения, на которых мы остановимся ниже.

Так как мемристор — односвязное устройство, он является взаимным для p — q -параметров. Очевидно также, что все определения можно обобщить и на случай многосвязных невзаимных сопротивлений.

Чем же тогда отличается мемристор от сопротивления? Рассмотрим в качестве примера конический амортизатор, показанный на рис.3.

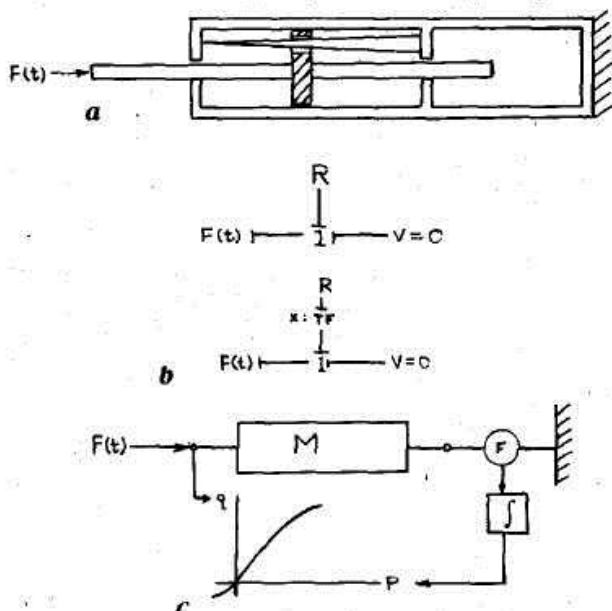


Рис.3.

a — конический амортизатор; b — граф связей для управляемого смещением амортизатора (вверху) и граф связи для обычного амортизатора (внизу); c — структурная схема для определения характеристики мемристора.

При попытке изобразить на e — f -плоскости свойства этого устройства (ошибочно считая его простым сопротивлением) вместо однозначной характеристики $F(e, f)=0$ мы получили бы кривую с некоторым гистерезисом, так как элементарное сопротивление зависит от мгновенного положения поршня. Учтеть это, казалось бы, можно, моделируя амортизатор переменным сопротивлением R , зависящим от параметра x (рис. 3) (здесь мы используем аналоги: усилие — скорость и поток — сила).

Но x не является переменной состояния какого-либо элемента системы. На самом деле, нужно перемещение самого амортизатора. (Примером подобного управляемого сопротивления может служить электролитический раствор, число носителей зарядов в котором зависит от концентрации электролита. Такая модуляция концентрацией всегда неявным образом присутствует в электрических системах, так как в определении переменной потока входит концентрация, от которой зависит сопротивление: $f=qv_d$, где v_d — скорость дрейфа

Кононюк А.Е. Графы

электрона или иона). Выбор мемристора в качестве модели дает возможность исключить громоздкую модуляцию и получить однозначную характеристику в x — p -плоскости. При этом следует учитывать важное ограничение, вытекающее из формы характеристической кривой, а именно: мемристор можно использовать для моделирования только линейных управляемых сопротивлений. Характеристика же реального амортизатора, например, может иметь вид $e=F(q)/|f|$. На рис.3, *c* приведена структурная схема определения характеристической кривой реального конического амортизатора.

Примеры

В качестве примеров использования мемристоров были промоделированы механическая и электрическая системы. Моделируемая механическая система с коническим амортизатором может служить грубой моделью автомобильной подвески с зависящими от смещения характеристиками амортизатора. Моделируемая электрохимическая система представляет собой простейшую электрическую цепь с мембранным выпрямителем : электролитический элемент, электролитическое сопротивление которого зависит от концентрации электролита между заряженными мембранами.

На рис.4, *a* дана схема механической системы, где масса — это масса автомобиля, пружина и амортизатор — его подвеска, а источник скорости— воздействие неровностей дороги.

Кононюк А.Е. Графы

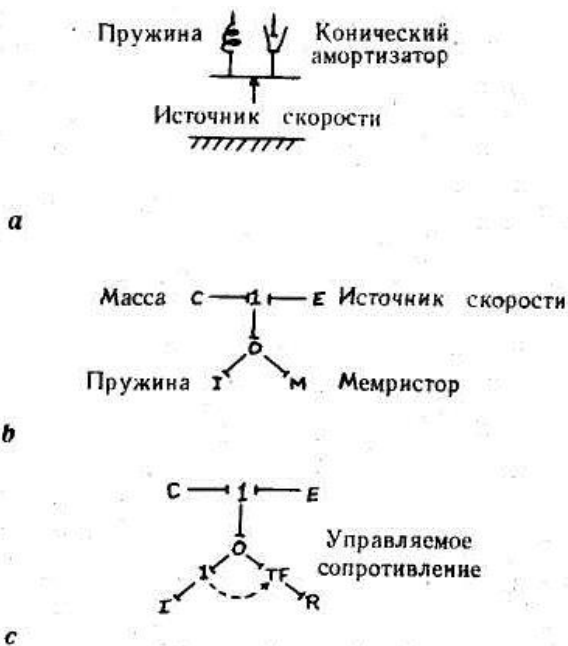


Рис.4. Схематическое изображение и граф связей для механической системы с управляемым смещением амортизатора.

a — схематическое изображение системы; *b* — граф связей с мемристором; *c* — граф связей с управляемым сопротивлением. 1 — пружина; 2 — конический амортизатор; 3 — источник скорости; 4 — масса; 5 — мемристор; *b* — управляемое сопротивление.

Граф связей системы с мемристором в качестве модели амортизатора представлен на рис.4, *b*, где через *e* и *f* обозначены соответственно скорость и сила. В данной системе конический амортизатор можно представить также и в виде управляемого сопротивления, поскольку смещение амортизатора совпадает со смещением пружины и, следовательно, оказывается пропорциональным напряжению в пружине. Граф связей для этого приведен на рис.4, *c*. Характерно, что даже для такой системы, которая допускает представление с помощью стандартных графов связей, использование мемристоров позволяет сократить количество элементов и избежать введения связи принципиально иного типа — прерывистого графа модулятора.

Кононюк А.Е. Графы

На рис.5 изображены энергетические зависимости, характеризующие проводимость мемристора, для трех промоделированных случаев: $n > 1$ — амортизатор с монотонно возрастающей, $n < 1$ — убывающей и $n = 1$ — постоянной проводимостью.

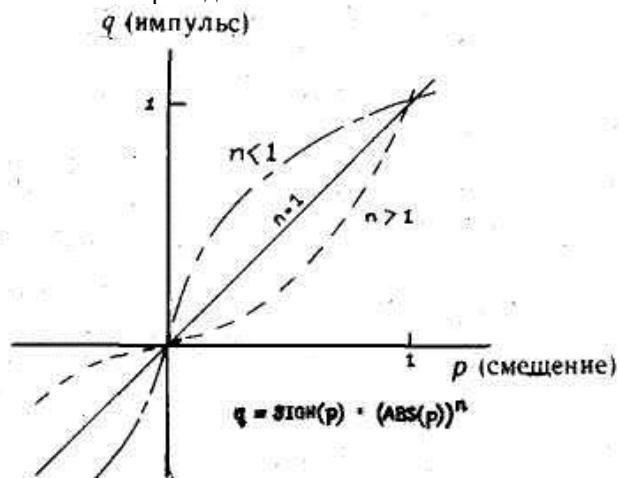


Рис.5. Кривые проводимостей мемристора

Реакция системы на синусоидальное входное воздействие приведена на рис.6.

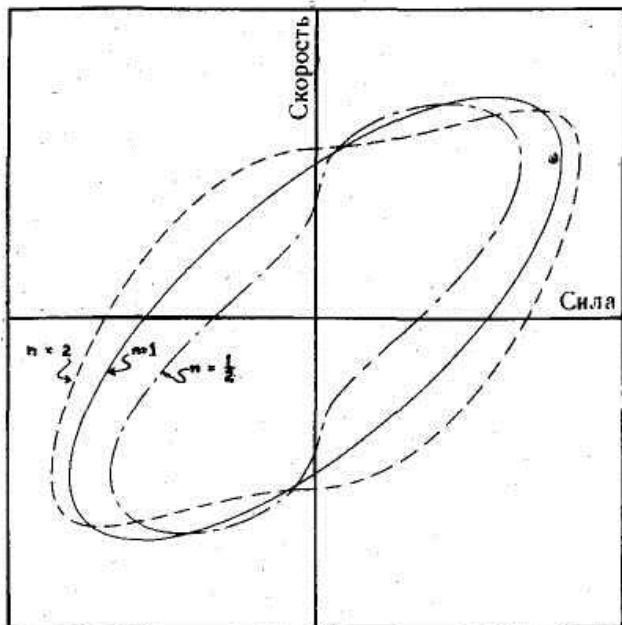


Рис.6. Фазовый портрет системы при синусоидальном возбуждении

Обозначения кривых на фазовой плоскости соответствуют обозначениям рис.5: сплошная линия— $n=1$, штриховая — $n=2$, штрихпунктирная — $n=0,5$. Полученные кривые полностью совпадают с результатами расчетов движения системы с управляемым сопротивлением вместо мемристора. Случай $n=1$ соответствует обычному линейному амортизатору с фазовой траекторией, как и следовало ожидать, в виде эллипса. Две другие траектории — неэллиптического характера, вызванного нелинейной зависимостью между p и q и, следовательно, высшими гармониками, что вообще типично для нелинейных систем. Наклон траектории вблизи оси скорости (т. е. при малом вынуждающем воздействии) характеризует тип амортизатора: при $n=2$ амортизатор очень мягкий относительно своего центрального положения, траектория при этом почти горизонтальна. Наоборот, при $n=0,5$ амортизатор оказывается очень жестким, а траектория — крутой. (Теоретически крутизна траектории в этой точке бесконечна, но при конечноразностной аппроксимации эта величина хотя и велика, но ограничена, что, конечно, более соответствует действительности.)

Кононюк А.Е. Графы

Электрохимическая система, изображенная на рис.7, *a*, представляет собой электролитическую ванну с двумя электродами и двумя противоположно заряженными мембранами.

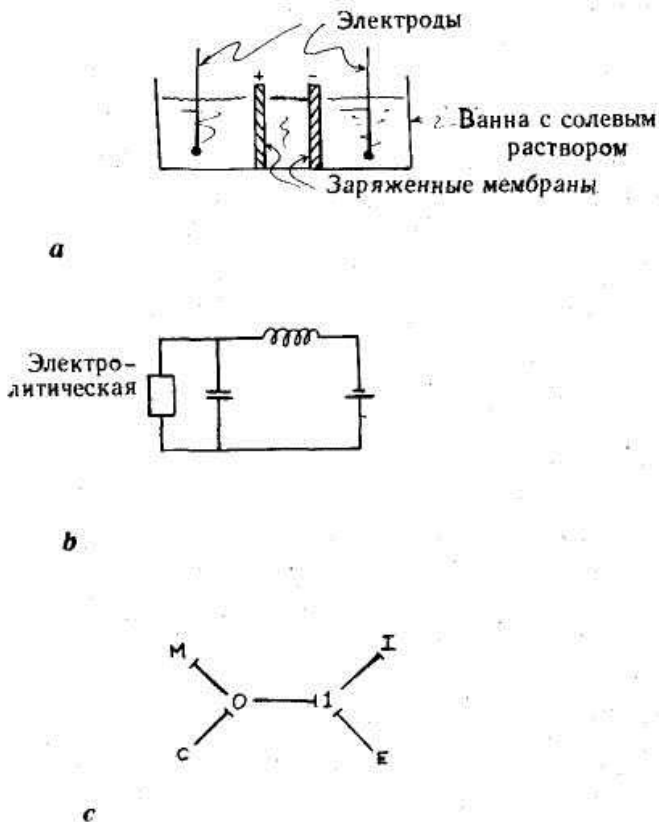


Рис.7. Электрохимическая система с мемристором.

a — электролитическая ванна с противоположно заряженными мембранами; *b* — эквивалентная электрическая цепь; *c* — граф связей для моделируемой системы.

Поскольку мембраны при прохождении электрического тока препятствуют движению ионов с зарядом того же знака, концентрация электролита в межмембранной области будет возрастать или убывать в зависимости от направления тока. Так как электрическая проводимость

Кононюк А.Е. Графы

падает с уменьшением концентрации ионов, внутреннее сопротивление окажется зависящим от общего количества прошедшего через систему тока. Концентрация (а следовательно, и сопротивление) будет непрерывно изменяться, пока протекает ток. (При моделировании предполагалось, что мембраны обладают полной селективностью. В действительности же система придет в установившееся состояние при очень высокой (или очень низкой) концентрации из-за стабилизирующего межмембранную концентрацию возросшего градиента диффузии. При этом возникнет некоторый поток одноименно заряженных с мембраной ионов).

Была промоделирована также система, изображенная на рис.7, *b* (ее граф связей дан на рис.7, *c*). Здесь уже нет переменных, которые можно было бы использовать для модуляции сопротивления, так что мемристор оказался единственным возможным элементом для точного моделирования ионных потоков в электролитической ванне. Отметим, что, хотя мемристор и является диссипативным элементом, он — чисто динамическое устройство, требующее задания начального условия $q(0)$. Размерность пространства состояний рассматриваемой системы равна трем, а не двум, как для RLC-модели.

Поскольку концентрация — величина неотрицательная, следует ожидать, что характеристическая кривая в плоскости p — q будет асимметричной. Если бы предположить, что электрическое сопротивление обратно пропорционально концентрации в межмембранной области, то получили бы экспоненциальную характеристику мемристора (штриховая линия на рис.8).

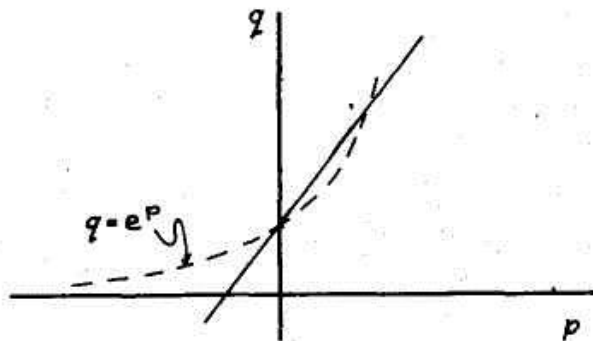


Рис.8. Кривые сопротивления мемристора.

Для сравнения на рис.8 приведена характеристика (сплошная линия) линейной системы, т. е. случай постоянного сопротивления.

Результаты исследования на фазовой плоскости приведены на рис.9 для случая синусоидального возбуждения.



Рис.9. Фазовый портрет системы при синусоидальном возбуждении.

Как и следовало ожидать, линейная система даст эллиптическую траекторию, указывающую на отсутствие гармоник. Асимметричная траектория системы указывает на присутствие гармоник, отличных от гармоник возбуждающего сигнала, вызванных нелинейностью характеристики мемристора.

Отметим два очевидных ограничения при использовании мемристора в качестве элемента моделей.

1) В соответствии с уравнениями (4а)-(4б) мемристор, рассматриваемый на $e-f$ -плоскости, моделирует линейно управляемое сопротивление. Нелинейность проявляется вследствие нелинейной

Кононюк А.Е. Графы

зависимости между p и q , но поверхность $f=f(e, q)$, отражающая характеристики сопротивления, может быть только линейчатой поверхностью, т. е. с образующими — прямыми линиями.

2) Мемристоры с горизонтальными ($q=\text{const}$) или вертикальными ($p=\text{const}$) характеристиками, как правило, непригодны в качестве моделей, так как в этих случаях устройство будет накапливать усилие даже при статическом смещении. При смене полярности поэтому возникнет гистерезис в p — q -и e — f -плоскостях.

1.8. Применение правила циклов к графам связей

Правило циклов Шэннона — Мэсона позволяет непосредственно осуществить превращение линейного графа потоков сигналов с постоянными параметрами приведения к передаточной функции. Графы потока сигналов могут быть построены из графов связей или из систем уравнений. Ниже приводится методика, предложенная Ф. Брауном, применения правила передачи непосредственно к графам связей с учетом определенных правил, которая позволяет получить передаточную функцию быстрее и с меньшей вероятностью ошибки. Выделены также четыре независимых класса сетей графов связей с применением в физических аналогиях и при вычислениях.

Введение

Топологическое правило циклов в своей классической форме позволяет, минуя промежуточные этапы, получить аналитическое выражение передаточной функции между любым входом и выходом графа потока сигналов (или блок-схемы) линейной системы с постоянными параметрами. Линейность понимается здесь в самом узком смысле допустимости наложений сигналов и операций; лишь в операции суммирования допускается два и более входных сигнала. Несмотря на это строгое ограничение и широкое распространение ЭВМ, правило циклов остается чрезвычайно полезным, особенно при итеративном предварительном анализе, позволяющем сэкономить время и средства, что в особенности важно для проектировщика систем. Правило циклов было найдено Шэнноном в 1941 г., но не приобрело широкой известности, пока не было вновь открыто Мэсоном в 1955 г. Это правило в стройной форме изложено во многих работах, где даны как индуктивный вариант доказательства, так и вариант, основанный на правиле Крамера, а также множество приложений. Само правило воспроизводится в ряде элементарных руководств по системному анализу и управлению. Основные

Кононюк А.Е. Графы

положения, предполагающие знакомство с графами потока сигналов, вкратце изложены в данном разделе.

Здесь правило циклов приводится к виду, удобному для использования в графах связей, в результате чего метод выигрывает в скорости, оставаясь тем не менее пригодным для аналитического определения передаточных функций в больших пассивных линейных системах.

Причина потенциальной простоты применения правила циклов к графам связей лежит отчасти в относительной простоте самих графов. Пожалуй, более существенная причина кроется в четкой выраженности энергетически пассивных и активных элементов в графах связей по сравнению с большой неясностью отражения этого факта в графах потока сигналов. Вследствие этого непосредственное использование графов связей дает возможность почти без ошибок определять знаки передаточной функции, тогда как при использовании графов потока сигналов это составляет основную проблему, являясь источником большинства возможных ошибок. Далее приводится ряд теорем, в которых устанавливаются определенные фундаментальные различия между классами сетей графов связей.

Правило циклов

Правило циклов утверждает, что соответствие между заданным входным сигналом и заданным выходным сигналом в графе потока сигналов определяется следующим выражением:

$$L = \frac{\sum_n L_n \Delta_n}{\Delta}, \quad (1)$$

где Δ — *определитель графа* (одинаковый для всех пар входов и выходов данной системы), L_n — передача n -го *пути* между заданными входом и выходом, а Δ_n — определитель сокращенного графа, образующегося в результате исключения пути с передачей L_n и вершин, через которые он проходит, из исходного графа. Суммирование производится по всем имеющимся путям; путь по определению не содержит замкнутых циклов.

Определитель графа потока сигналов может быть записан следующим образом:

$$\Delta = 1 - \sum_r p_r^{(1)} + \sum_r p_r^{(2)} - \sum_r p_r^{(3)} + \dots, \quad (2)$$

где $p_r^{(m)}$ — r -е произведение коэффициентов передачи циклов или коэффициентов усиления для m циклов графа, взятых из множества

Кононюк А.Е. Графы

независимых циклов. Сумма берется по всевозможным таким комбинациям. Так,

$\sum_r \rho_r^{(1)}$ — сумма всех отдельных коэффициентов передачи отдельных циклов,

$\sum_r \rho_r^{(2)}$ — сумма всех произведений пар коэффициентов передачи циклов, взятых из множества независимых циклов,

$\sum_r \rho_r^{(3)}$ характеризуют независимые циклы, взятые по три, и т. д.

Два цикла, проходящие через различную последовательность вершин, различны, а если оба они проходят хотя бы через одну или несколько общих вершин, то они касаются. (Цикл, проходящий только через одну вершину, называется *петлей* и касается всех без исключения других циклов, проходящих через эту вершину). Поиск элементов рассматриваемых выше сумм обычно обрывается на ранних этапах, а сами элементы можно легко найти из непосредственного рассмотрения графа.

Рассмотрим пример на рис.1, где имеется три входа (τ_1, τ_2, τ_d) и два различных выхода (ω_1, ω_2 ; переменная ω_d является зависимой: $\omega_d = (\omega_1 + \omega_2)$).

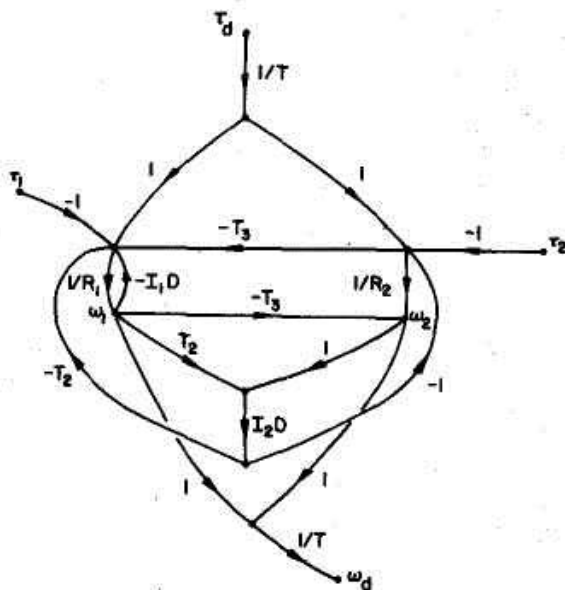


Рис.1. Пример графа потока сигналов.

Кононюк А.Е. Графы

Остальные символы определяют постоянные коэффициенты, за исключением D , являющегося оператором дифференцирования d/dt .

Коэффициенты усиления шести различных циклов графа равны: — I_1D/R_1 — I_1D/R_1 (только два независимых цикла); + $T_2T_3I_2D/R_1$ (два одинаковых цикла); — $T^2_3I_2D/R_1$; — $T^2_2I_2D/R_1$.

Независимых циклов, взятых по три, нет, следовательно,

$$\sum_r \rho_r^{(3)} = 0,$$

и определитель графа равен

$$\begin{aligned} \Delta = 1 + \frac{I_1D}{R_1} + \frac{I_2D}{R_2} + \frac{T^2_2I_2D}{R_1} + \frac{T^2_3I_2D}{R_1} + \frac{2T_2T_3I_2D}{R_1} + \\ + \frac{I_1I_2D^2}{R_1R_2} = 1 + \left[\frac{I_1}{R_1} + \frac{I_2}{R_2} + \frac{I_2}{R_1} (T_2 - T_3)^2 \right] D + \\ + \left[\frac{I_1I_2}{R_1R_2} \right] D^2. \end{aligned} \quad (3)$$

Для входа τ_1 и выхода ω_1 , имеется только один коэффициент передачи пути $L_1 = -I/R_1$. Граф, в котором удалены две вершины, принадлежащие этому пути, имеет один цикл с коэффициентом передачи, равной I_2D/R_2 . Следовательно,

$$\omega_1 = -\frac{1}{\Delta R_1} \left(1 + \frac{I_2D}{R_2} \right) \tau_1. \quad (4)$$

Используя аналогичную процедуру, можно проверить пять других передаточных функций приведенных ниже:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \frac{1}{R_1 \Delta} \begin{bmatrix} - \left(1 + \frac{I_3 D}{R_2} \right) \\ T_3 + \frac{T_2 I_2 D}{R_2} \end{bmatrix}$$

$$\begin{bmatrix} - \left(\frac{R_1}{R_2} + T_3 + \frac{I_1 D}{R_2} + \frac{T_2 I_2 D}{R_2} \right) \\ \frac{1}{T} \left[1 - T_3 + (1 - T_3) \frac{I_3 D}{R_2} \right] \\ \frac{1}{T} \left[\frac{R_1}{R_2} + T_3 - T_3 + \frac{I_1 D}{R_2} + (T_2 - 1) \frac{T_2 I_2 D}{R_2} \right] \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_d \end{bmatrix}$$

(5)

Рассмотрим эту проблему в дальнейшем, используя графы связей.

Принятые обозначения в графах связей.

Здесь приводятся нестандартные обозначения графов связей, которые подробно описаны в табл. 1, большей частью используя для представления элемента графа связей обозначение, употребляемое для параметрического задания элемента.

Кононюк А.Е. Графы

Таблица 1

Элементы графа связей и определения инвариантных модулей

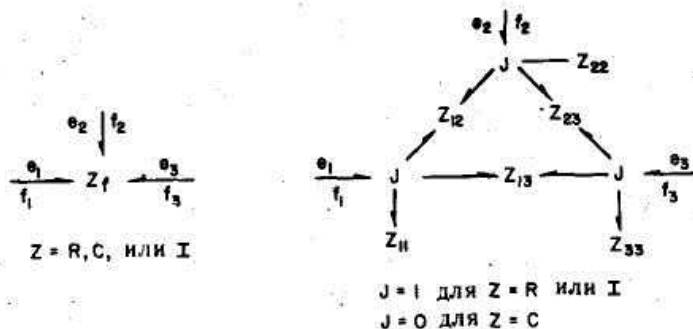
Тип элемента	Название элемента	Символика, употребляемая автором	Общепринятая символика	Отношения для линейных элементов (инвариантных модулей)	Накопленная энергия ($\epsilon = \int e/dt$) или рассеиваемая мощность ($P_d = eI$)	
Одно-связные элементы	Источники	1. Источник усилия	$E \frac{e}{f}$ или $S_e \frac{e}{f}$	Та же	$e = E(t) = S_e(t)$	
		2. Источник потока	$F \frac{e}{f}$ или $S_f \frac{e}{f}$		$f = F(t) = S_f(t)$	
	Потери	3. Сопротивление	$\frac{e}{f} \rightarrow R$	Не применяется	$e - Rf = 0$	$P_d = Rf^2 = \frac{1}{R} e^2$
Аккумуляторы энергии	4. Упругая деформация 5. Инерционность	4. Упругая деформация	$\frac{e}{f} \rightarrow R$	Не применяется	$f - C \frac{de}{dt} = 0$	$\epsilon = \frac{1}{2} Ce^2 = \frac{1}{2C} q^2$
		5. Инерционность	$\frac{e}{f} \rightarrow I$		$e - I \frac{df}{dt} = 0$	$\epsilon = \frac{1}{2} If^2 = \frac{1}{2I} p^2$
		6. Трансформаторы 7. Гираторы	$\frac{e_1}{f_1} T \frac{e_2}{f_2}$ $\frac{e_1}{f_1} G \frac{e_2}{f_2}$		$\frac{TF}{\text{или } MTF}$ $\frac{GY}{\text{или } MGY}$	$f_2 - Tf_1 = 0; e_1 - Te_2 = 0$ $e_1 - Gf_2 = 0; e_2 - Gf_1 = 0$
Двусвязные элементы	Связывающие элементы	8. Взаимное сопротивление	$\frac{e_1}{f_1} R \frac{e_2}{f_2}$	Не применяется	$e_1 - Rf_2 = 0; e_2 - Rf_1 = 0$	$P_d = 2Rf_1f_2 = \frac{1}{R} e_1e_2$
		9. Взаимная упругая деформация 10. Взаимная инерционность	$\frac{e_1}{f_1} C \frac{e_2}{f_2}$ $\frac{e_1}{f_1} I \frac{e_2}{f_2}$	Не применяется	$f_1 - C \frac{de_2}{dt} = 0;$ $f_2 - C \frac{de_1}{dt} = 0;$ $e_1 - I \frac{df_2}{dt} = 0;$ $f_2 - I \frac{df_1}{dt} = 0$	$\epsilon = Ce_1e_2 = \frac{1}{C} q_1q_2$ $\epsilon = If_1f_2 = \frac{1}{I} p_1p_2$
	Потери	13. Поле сопротивления	$\frac{e}{f} \rightarrow R$	$\frac{e}{f} \rightarrow R$	$e - Rf = 0; R = R_f$	$P_d = IfRf = e^2 R^{-1} e$
Трех-связные (многосвязные) элементы	Узлы	11. Нулевой узел	$\frac{e_2}{f_2} \downarrow \frac{e_3}{f_3}$ $\frac{e_1}{f_1} \downarrow 0$	Та же	$e_1 = e_2 = e_3;$ $f_1 + f_2 + f_3 = 0$	$P_d = 0; \epsilon = 0$
		12. Единичный узел	$\frac{e_2}{f_2} \downarrow \frac{e_3}{f_3}$ $\frac{e_1}{f_1} \downarrow 1$		$f_1 = f_2 = f_3;$ $e_1 + e_2 + e_3 = 0$	$P_d = 0; \epsilon = 0$
	Аккумуляторы энергии	14. Поле упругой деформации 15. Инерционное поле	$\frac{e}{f} \rightarrow C$ $\frac{e}{f} \rightarrow I$	$\frac{e}{f} \rightarrow C$ $\frac{e}{f} \rightarrow I$	$f - C \frac{de}{dt} = 0; C = Ct$ $e - I \frac{df}{dt} = 0; I = It$	$\epsilon = \frac{1}{2} e^2 C = \frac{1}{2} q^2 C^{-1} q$ $\epsilon = \frac{1}{2} If^2 = \frac{1}{2} p^2 I^{-1} p$

Например, емкость --- C_3 равна C_3 , трансформатор $T = (f_2/f_1) = (e_1/e_2)$, гиратор $G = (e_1/f_2) = (e_2/f_1)$.

Такие обозначения объясняют сокращения символов TF и GY до T и G . Обобщенные многосвязные поля обозначаются теми же буквами, что и

Кононюк А.Е. Графы

односвязные, но Браун Ф. предпочитает придавать более специфическое значение символам $\text{---}R\text{---}$, $\text{---}C\text{---}$ и $\text{---}I\text{---}$, называя их соответственно *взаимное* сопротивление, *взаимная* податливость и *взаимная* инерционность, снова допуская двойственное использование символов. При анализе физических систем очень часто могут быть выявлены взаимные энергии и потери, и в этом случае можно использовать простые топологические эквивалентные преобразования графов связей (некоторые из которых показаны на рис.2) для удаления из графа связей всех многосвязных полей прежде, чем обращаться к отдельным алгебраическим процедурам.



a



b

Рис.2. Сокращение дву- и трехсвязных полей.

c — сокращение обобщенного линейного взаимного трехсвязного поля; *b* — дальнейшее сокращение составляющих двусвязных полей.

Иногда (хотя и редко) встречаются обобщенные многосвязные поля; они сопровождаются символом *f* (сокращение слова field — поле). При этом предполагается, что ориентации мощностей, показанные стрелками на табл. 1, выбраны там, за исключением источников и узлов, где ориентация произвольна. В частности, у всех односвязных

Кононюк А.Е. Графы

импедансов ($\text{---}R$, $\text{---}C$ и $\text{---}I$) мощность ориентирована *внутрь*, у всех идеальных 2-связных элементов ($\text{---}G$ и $\text{---}T$) обе мощности ориентированы *в одном направлении*, а у всех взаимных элементов ($\text{---}R$ --- , $\text{---}C$ --- , $\text{---}I$ ---) обе мощности ориентированы *симметрично внутрь*. Следовательно, необходимым (и достаточным в 1-связном случае) условием пассивности являются положительные значения R , C и I , хотя в общем случае пассивность не обязательна. Далее предполагается, что циклические коэффициенты передачи, о которых шла речь, должны быть определены в терминах символов R , C , I , T , G (с соответствующими индексами), а не в терминах значений самих параметров, которые могут быть отрицательны. Активность связей, при которой одним из двух сигналов, относящихся к связи, можно пренебречь, отмечается обычно стрелкой, расположенной в центре связи.

Применение правила передачи к графам связей

Прежде чем применять правило передачи, необходимо удалить из графа связей все обычно неопределенные поля потерь и аккумуляторные поля R_f , C_f и I_f . Последние можно заменить специальными схемами односвязных и взаимных элементов тех же типов, как показано на рис.2, *a*. Однако из соображений, приведенных ниже, лучше продолжить сокращение, исключив также и взаимные элементы, с использованием эквивалентов, представленных на рис. 2, *b*.

Затем, исходя из требований непротиворечивости, в графе связей производится распределение причинных связей. В результате этого граф связей (или граф энергий, мощностей) переходит в граф потока сигналов. Если необходимо, чтобы передаточная функция была выражена полиномом относительно оператора D (или s), а не $1/D$ (или $1/s$), то вместо интегральных причинных отношений (т. е. $\text{---}C$, $\text{---}I$, $\text{---}C$ --- , $\text{---}I$ ---) необходимо пользоваться дифференциальными причинными отношениями (т. е. $\text{---}C$, $\text{---}I$, $\text{---}C$ --- , $\text{---}I$ ---). Это отлично от ситуации, где желательна формулировка в виде пространства состояний .

Граф потока сигналов для соответствующего графа связей образуется путем замены каждой простой связи парой противоположно направленных потоков сигналов (дуг) с применением эквивалентных соотношений для элементов, приведенных в табл. 2, и соединением связей с соответствующими вершинами. Активные связи заменяются элементарным потоком сигнала.

Интерпретация обозначений элементов графов потока сигналов для элементов причинного графа связей

Элемент причинного графа связей	Интерпретация в обозначениях графа потока сигналов	Элемент причинного графа связей	Интерпрет в обознач графа потока сигн.

Чтобы найти определитель графа, мы должны вычислить коэффициенты передачи всех циклов в графе потока сигналов и определить множество независимых циклов. В целях разъяснения разобьем проблему на пять этапов:

- 1) определение всех циклов;
- 2) определение всех независимых циклов;
- 3) определение коэффициентов передач для каждого цикла;
- 4) определение знаков коэффициентов передач для каждого цикла;
- 5) расчет действия имеющихся активных связей.

На практике эти этапы могут быть выполнены почти одновременно.

Определение циклов. Будем различать две значительные категории циклов: *плоские* и *открытые*. Плоский цикл включает *оба* потока сигналов для каждой рассматриваемой связи. В итоге эти связи образуют простой каскад односвязных элементов (не источников) с любого конца, как изображено на рис.3.

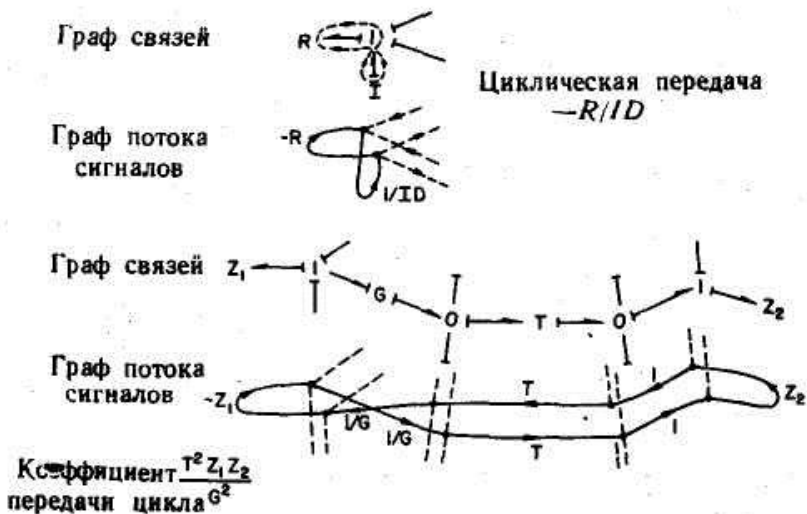


Рис.3. Примеры плоских циклов.

При определении открытого цикла учитывается только один из двух потоков сигналов, соответствующих некоторым или всем рассматриваемым связям. Это определение требует учета замкнутых каскадов, или *сетей* (цикл графов связей) *.

* Заметим, что циклы на рис.3 отсутствуют, так как не существует причинных связей, исходящих из центрального узла.

Большинство циклов в пассивных системах плоские, и фактически *древообразный граф связей (без сетей) не имеет других циклов, кроме плоских.*

Для обоих типов циклов *причинные связи должны быть направлены в обе стороны каждого из 0- или 1-узлов*, что можно увидеть на соответствующем графе потока сигналов. Перемена направления причинных связей на обратное имеет место только в случае гираторов и взаимных элементов (если таковые имеются).

Это свойство является ключевым для быстрого определения различных циклов без фактического вычерчивания графа потока сигналов. Для плоских циклов каждый односвязный элемент может быть выбран лишь один раз в качестве возможного указателя окончания каскада связей, и от этого элемента ко всем другим анализируются причинные пути. Каждый плоский цикл находится с помощью этой процедуры только дважды (что полезно для проверки). Для открытых циклов каждая сеть выбирается один раз. Если причинность всех связей в сети

Кононюк А.Е. Графы

направлена в одну сторону (против или по часовой стрелке), то одна такая сеть позволяет определить пару противоположно ориентированных открытых циклов.

Однако чаще в одном или нескольких узлах причинная связь для сети меняется на обратную, и это может быть причиной возникновения ответвления (рис.4).



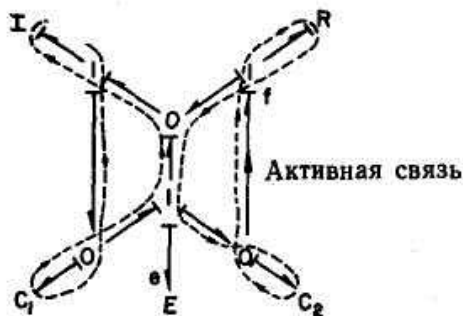
Рис.4. Пример открытого цикла, содержащего ответвление сети.

Ответвления сети обычно включаются в односвязный импеданс, а последний входит затем в открытые циклы. Возможно, что ответвление окажется длинным и даже соединяющимся со вторым ответвлением. К счастью, такие осложнения встречаются редко, хотя и их можно решить удовлетворительно.

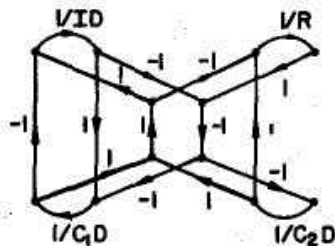
Определение касающихся циклов. Циклы, не имеющие общих 0- или 1-узлов, не касаются один другого. Плоский цикл касается всех циклов, плоских и открытых, которые включают хотя бы один из его узлов. Два открытых цикла, связанных с сетью, касаются тогда и только тогда, когда они имеют хотя бы одно общее ответвление.

Два цикла из различных сетей касаются, если они проходят хотя бы через один общий узел, который является узлом ответвления хотя бы для одной из сетей.

Для выполнения вышеприведенных указаний достаточно беглой проверки графа связей. Только в случаях циклов из различных сетей, проходящих через один или несколько неотвечающих узлов, необходима более тщательная проверка общих узлов. Каждый из этих циклов проходит только через тот поток общего узла, который связан с усилием в точке прохождения; такие циклы касаются один другого тогда и только тогда, когда хотя бы в одном из общих узлов они проходят в одном направлении. На рис.5 приведен пример двух открытых циклов, которые не касаются, несмотря на два общих узла.



a



b

Рис.5. Пример двух независимых открытых циклов.

a — граф связей, иллюстрирующий независимые открытые циклы (штриховые линии); *b* — граф потока сигналов.

Достаточным, но не необходимым условием зависимости двух циклов является наличие у них общего односвязного импеданса. Это правило особенно полезно на практике, так как оно охватывает большинство случаев зависимых циклов и при его применении достаточно проверки списка коэффициентов передачи циклов вместо непосредственного обращения к графу связей.

Определение величин коэффициентов передач циклов.

Коэффициент передачи цикла содержит модули элементов содержащихся связей (некоторые в числителе и некоторые в знаменателе в зависимости от причинности). Импеданс односвязного элемента $1/Z$ (это может быть R , ID или $1/CD$) появляется в

Кононюк А.Е. Графы

числителя (или адмиттанс в знаменателе), а для элемента $\text{---}|Z\text{---}$ наоборот.

Забывчивые практики это правило перед решением проблемы видоизменяют, различая переменные, являющиеся причинными входами, и переменные, являющиеся причинными выходами; после видоизменения это правило можно применять многократно, не обращаясь к самим потокам сигналов.

В случае плоских циклов модули трансформаторов, гираторов и взаимных элементов в циклических передачах всегда возникают дважды (т. е. возводятся в квадрат); для $\text{---}|T\text{---}$ имеем T^2 , для $\text{---}|G\text{---}$ имеем G^2 , для $\text{---}|R\text{---}$ имеем R^2 , для $\text{---}|I\text{---}$ имеем I^2D^2 и для $\text{---}|C\text{---}$ имеем C^2D^2 , что можно наблюдать из причинных связей сигналов. Для причинных связей, направленных противоположно, берутся соответственно обратные значения. Если эти элементы встречаются в сетевой части (в противоположность ответвлениям) открытого цикла, то их модули используются лишь один раз. Ввиду нашего соглашения об ориентации мощностей знаки элементов в обоих случаях берутся положительными.

Безразмерность всех коэффициентов передачи циклов особенно полезна при выявлении ошибок.

Определение знаков коэффициентов передачи циклов, топология сетей графов связей. Знаки любых сигнальных путей, включая циклы, можно определить, многократно используя одно или два правила. Для циклов менее сложной и менее чувствительной к ошибкам является альтернативная система правил, которая включает семь теорем. Эти правила и теоремы изложены ниже; схемы доказательств приведены в приложении.

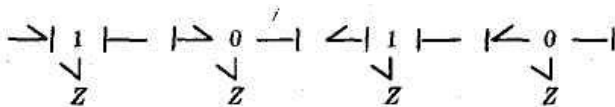
Изменения знаков могут возникнуть только у 0- или 1-узлов. Если стрелки, обозначающие мощности двух связей на каждой стороне узла, направлены в одну сторону, как в случаях

$$\text{---}|J\text{---}| \text{ или } \text{---}|J\text{---}| \text{---} \quad (J=0 \text{ или } 1),$$

то знаки сигнальных путей, направленных как влево, так и вправо, не меняются на противоположные. Однако если они направлены встречно, т. е. $\text{---}|J\text{---}| \text{ или } \text{---}|J\text{---}|$, то в одном из сигнальных путей знак меняется. В случае 0-узла знак меняется у переменной, обозначающей поток, в случае 1-узла — у переменной, обозначающей усилие.

Применяя это правило к ответвлению сети или к другому пути, определяемому ситуациями

Кононюк А.Е. Графы



(эти ситуации требуют исключения из рассмотрения причинных связей, в которых ответвление не входит в сигнальный путь), получим производное правило: *знак сигнала в применении к ответвлению и его узлу не меняется тогда и только тогда, когда направление сигнала в узле совпадает с направлением мощности в том же узле.* Следовательно, знаки левоориентированных сигнальных передач для ответвлений, указанных выше, зависят только от ориентации мощности в связях справа от приведенных выше (не указаны).

Этих правил достаточно для определения всех необходимых знаков. Однако проверка правильности знаков в графах потока сигналов может оказаться затруднительной и отнимающей много времени. Во избежание этих трудностей при работе с графами связей и в целях определения важных топологических особенностей четырех инвариантных классов графов связей (полезных также при проведении определенных аналогий и при создании программ моделирования на ЭВМ) сформулированы семь теорем. Предполагается, что переменные мощности заданы стандартным образом (подробно описаны ранее и в табл. 1).

Теорема 1. *Плоские циклы, не содержащие взаимных элементов, имеют отрицательный коэффициент передачи.*

Замечания. Из этой важной теоремы вытекает следствие, что *все составляющие определителя графа в древовидном графе связей без взаимных элементов положительны.* (Как утверждалось ранее, это имеет место в терминах параметрических символов, но не обязательно в терминах значения параметров). Один этот простой факт позволяет, по-видимому, избежать большинства ошибок в определении знаков, которые возможны при построении графов потока сигналов и применении к ним правила циклов.

Теорема 2. *Каждый взаимный элемент плоского цикла меняет знак коэффициента передачи цикла на обратный.* Остальные теоремы связаны с открытыми циклами. Сеть называется *простой*, если в самой сети отсутствуют трансформаторы, гираторы или взаимные элементы, хотя такие элементы допускаются в ответвлениях сети. Также предполагается, что исключены связи, непосредственно соединяющие два 1-узла или два 0-узла и 2-узлы (последние в любом случае исключаются неявным образом, так что этот момент имеет значение только для облегчения понимания на языке графов связей). Так как каждый узел сети соединен с обеих сторон с узлами противоположного

Кононюк А.Е. Графы

типа, то должно иметься четное число сетевых узлов (половина каждого типа) и четное число связей (т. е. образующих квадраты, шестиугольники, восьмиугольники и т. п.). Простая сеть называется *четной или нечетной* в зависимости от того, четное или нечетное число сетевых связей имеют одинаковую ориентацию мощностей, т. е. против или по часовой стрелке. (Само собой разумеется, нечетное количество ориентации по часовой стрелке требует также нечетного количества ориентации против часовой стрелки, а четное количество ориентации по часовой стрелке требует четного количества ориентации против часовой стрелки.)

Теорема 3. *Четность и нечетность являются свойствами простых сетей (т. е. сохраняются при любых допустимых изменениях в ориентации мощностей).*

Замечание. Четности и нечетности соответствуют совершенно различные физические системы. Так случилось, что язык графов связей был опубликован без учета ориентации мощностей. Для древовидных графов связей это зачастую ведет к двусмысленностям в знаках переменных. Однако для сетевидных графов связей сущность структуры системы сама по себе двусмысленна. В результате возникали большие вычислительные трудности. Из графа связей можно удалить простые четные сети, что способствует получению древовидных структур с помощью эквивалентов, показанных на рис.6. Простые нечетные сети таким же образом заменить нельзя, что также видно из рис.6. Следовательно, открытые циклы являются неизбежным результатом для нечетных сетей.

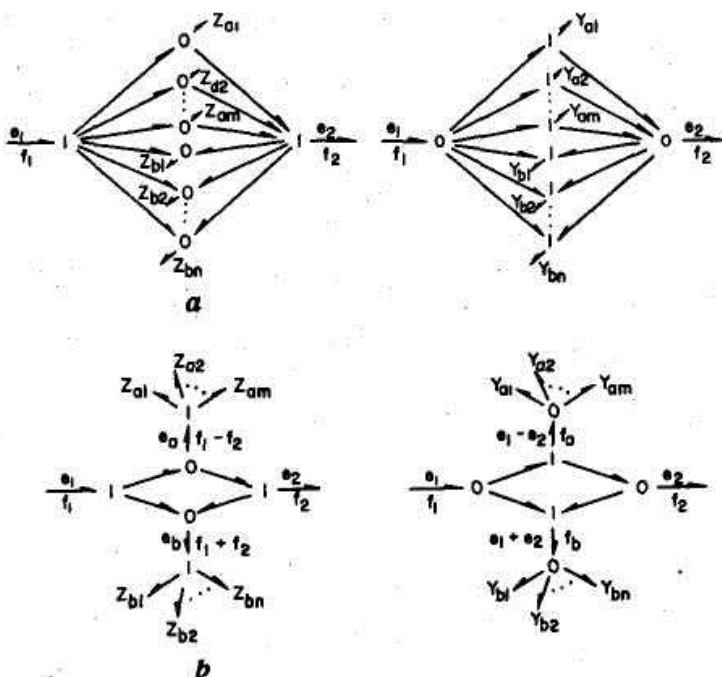


Рис.6. Сокращение простых сетей графов связей.
 а — графы связей двойственных типов; б — соответственно сокращенные графы связей.

Теорема 4. Открытые циклы, связанные с пассивными простыми сетями, представляют собой различные пары, положительные для четных сетей и отрицательные для нечетных.

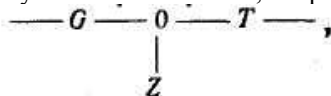
Замечание. Из положительных коэффициентов передач открытых циклов в четных сетях непосредственно получаются отрицательные

составляющие в выражении $\sum_r \rho_r^{(1)}$ определителя графа. Мы знаем, что четные сети можно свести к древовидным структурам, имеющим

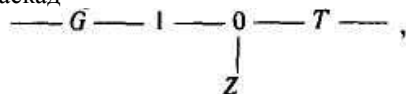
только положительные составляющие в $\sum_r \rho_r^{(1)}$. Определитель же графа в целом не меняется: все отрицательные знаки действительно компенсируются другими составляющими последовательности, что должно иметь место в случае энергетической пассивности системы.

Кононюк А.Е. Графы

Замкнутый путь в графе связей, содержащий произвольную последовательность двусвязных элементов и узлов, называется *общей сетью*. Путь между 0-узлом и первым 1-узлом (в любом направлении) называется *ветвью сети*. Будем требовать, чтобы сами ветви не содержали узлов, но это не будет являться ограничением, так как можно ввести *фиктивные* узлы, чтобы разорвать недопустимую ветвь на каскад меньших допустимых ветвей. Так, например, ветвь



связывающая 0-узел слева с 1-узлом справа (не показано), превращается в каскад



содержащий три отдельные ветви: гиратор, простую связь и трансформатор, называемые: первый — *гироскопической* ветвью, а два последних — *трансформаторной* ветвью. Введем реальное ограничение, не допуская существования взаимных элементов в ветвях сети, хотя в ответвлениях сети они будут допускаться. На практике желательно заменять последние с помощью различных эквивалентных соотношений, подобных приведенным на рис.2, *b*. Допущение взаимных элементов в ветвях сети значительно усложнило бы рассмотрение следующих теорем. Два открытых цикла общих сетей всегда равны по величине (так как в них входят одни и те же элементы с одинаковыми причинными связями), но иногда имеют противоположные знаки; в этом случае сеть называется *нейтральной*; можно также определить, какой из двух циклов — по часовой стрелке или против часовой стрелки — положителен. Определения четности и нечетности простых сетей обобщаются таким образом, что сеть с обоими положительными коэффициентами передачи открытого цикла называется *четной*, а с обоими отрицательными коэффициентами передачи — *нечетной*.

Теорема 5. *Четность, нечетность и нейтральность являются свойствами общих сетей, не содержащих взаимных элементов.*

Теорема 6. *Включение гиратора в четную или нечетную сеть превращает последнюю в нейтральную сеть, а включение гиратора в нейтральную сеть - превращает ее или в четную, или в нечетную сеть.*

Кононюк А.Е. Графы

Замечание. Два открытых цикла с противоположными знаками в нейтральной сети взаимно сокращаются в выражении

$$\sum_r p_r^{(1)}$$

определителя графа.

Во многих (или даже в большинстве) случаях нет необходимости знать, какой из циклов имеет положительную передачу, а какой — отрицательную. Для остальных случаев предлагается следующая теорема. Она основана на прямом следствии из теоремы 6, которая требует, чтобы нейтральная сеть имела нечетные числа гироскопических и трансформаторных ветвей.

Теорема 7. *Открытый цикл нейтральной сети с положительным коэффициентом передачи ориентирован в том же направлении (против или по часовой стрелке), что и мощность ветвей (число которых нечетно) в трансформаторной сети.*

Определение передаточных функций; активность связей

Определение числителя в уравнении (1) требует определения коэффициентов передач L_n от входа к выходу для различных путей, а также определения циклов, касающихся этих путей и, следовательно, удаляемых из Δ для получения Δ_n . Сами пути и соответствующие значения их коэффициентов передач можно легко определить путем перебора ациклических последовательностей причинных связей, направленных от входа к выходу с учетом возможных разветвлений. Знаки этих передач определяются с помощью правил, приведенных перед теоремами. При определении циклов, касающихся путей, необходимо последние рассматривать в точности так же, как открытые циклы, разветвления и пр. При учете активности связи отмечается, какие циклы и пути уничтожаются активной связью, и опускают соответствующие составляющие. Однако если посмотреть на проблему в целом, то необходимо заметить, что граф связей, в котором все связи активные, по сути дела переходит в граф потока сигналов, и не остается никаких преимуществ в подходе и терминологии графов связей. Тем не менее если граф связей в значительной мере или полностью пассивен, то он гораздо проще и понятнее, чем соответствующий граф потока сигналов.

Методика вычислений и примеры

Передаточные функции для простых графов связей, особенно без сетей, обычно можно записать, пользуясь только наглядным изображением графа. В более сложных ситуациях предлагается следующая процедура:

Кононюк А.Е. Графы

1. Определить все циклы и записать их передачи, отмечая четыре категории (плоские, открытые четные, открытые нечетные и открытые нейтральные) и все исключения циклов из-за активных связей. Этот этап составляет две трети всей работы.
 2. Записать все пары независимых циклов, все независимые циклы, взятые по три и т. д.
 3. Найти определитель графа Δ , используя уравнение (2) и полученную выше информацию.
 4. Определить все пути от входа к выходу и найти их передачи.
 5. Определить циклы, касающиеся этих путей; эти циклы не участвуют в формировании Δ_n из Δ . Получить выражение для передаточной функции, заданной уравнением (1).
- Граф потока сигналов на рис.1, обсуждавшийся ранее, соответствует зубчатому дифференциалу на рис.7, где показаны все инерции вращающихся частей и предполагается, что все силы торможения пропорциональны относительным скоростям.

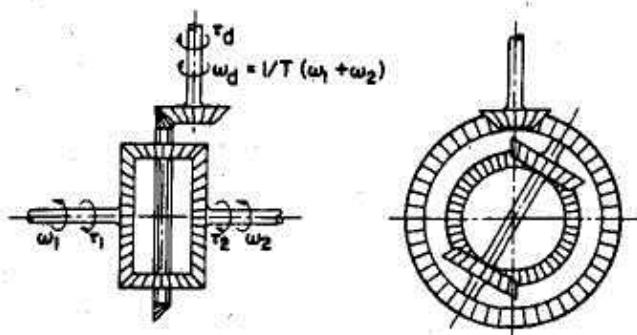


Рис.7. Схема дифференциации с коническими шестернями.

Наиболее удобным является энергетический подход (и можно показать, что он в общем случае устраняет проблемы причинных связей в методе прямого моделирования):

ограничительное соотношение:

$$\omega_1 + \omega_2 = T_1 \omega_d (T_1)$$

определяется передаточным числом);

потенциальная энергия:

$$\begin{aligned} \epsilon &= 1/2 I_1' \omega_1^2 + 1/2 I_p (\omega_1 - \omega_2)^2 + 1/2 I_c (\omega_1 + \omega_2)^2 + 1/2 I_2' \omega_2^2 = \\ &= 1/2 I_1'' \omega_1^2 + I_{12} \omega_1 \omega_2 + 1/2 I_2 \omega_2^2, \end{aligned}$$

$$I_1'' = I_1' + I_p + I_c; \quad I_{12} = I_c - I_p; \quad I_2 = I_p + I_c + I_2';$$

Конюк А.Е. Графы

входная мощность:

$$P_i = \tau_d \omega_d - \tau_1 \omega_1 - \tau_2 \omega_2;$$

потери:

$$\begin{aligned} P_d &= R_1' \omega_1^2 + R_2' \omega_2^2 + R_d \omega_d^2 + R_p (\omega_1 - \omega_2)^2 = \\ &= R_1'' \omega_1^2 + 2R_{12} \omega_1 \omega_2 + R_2 \omega_2^2, \end{aligned}$$

$$R_1'' = R_1' + \frac{R_d}{T_1^2} + R_p; \quad R_{12} = \frac{R_d}{T_1^2} - R_p; \quad R_2 = R_2' + \frac{R_d}{T_1^2} + R_p.$$

Эти четыре вида условий могут быть представлены непосредственно графом связей (рис.8).

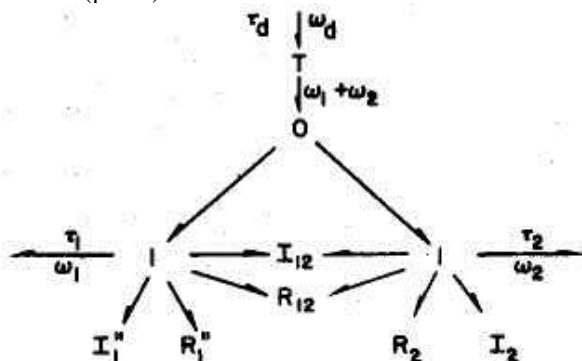
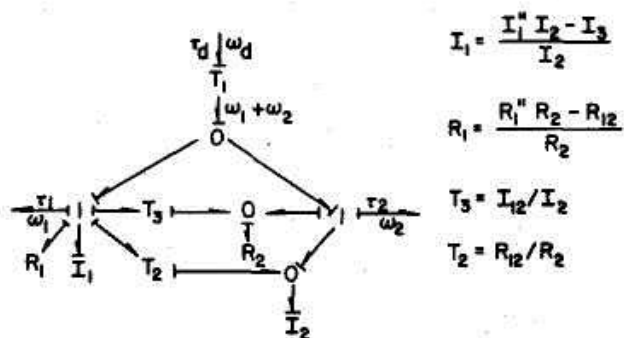
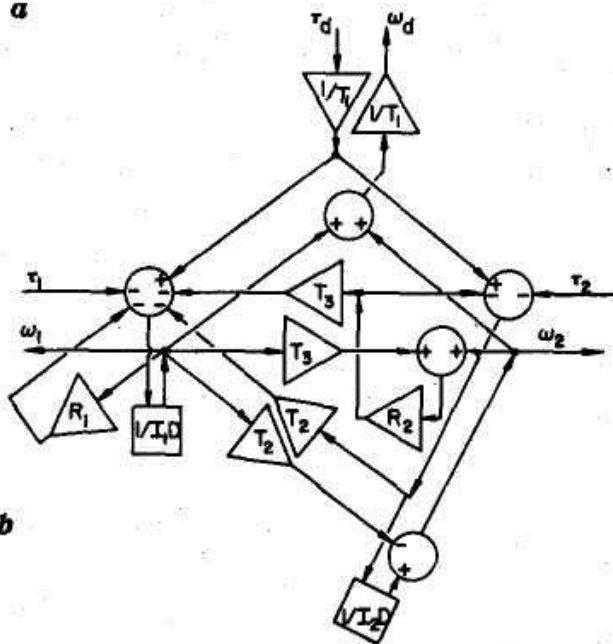


Рис.8. Граф связей дифференциала (рис.7).

Последовательное использование эквивалентов рис.2, *b* сокращает взаимные элементы, как показано на рис.9.



a



b

Рис.9. Приведенный граф связей дифференциала (рис.7) с интегральными причинными связями и соответствующая блок-схема.

a — причинный граф связей; b — блок-схема.

Теперь можно применить различные причинные связи с входом τ_1 или выходом ω_1 , что желательно для сравнения с нашими предыдущими результатами. На рис.9 показаны интегральные причинные связи с

Конюк А.Е. Графы

соответствующей блок-схемой, пригодной для моделирования на АВМ. Однако чтобы сразу получить полиномиальные отношения, используются *дифференциальные причинные связи* (рис.10), которым соответствует граф потока сигналов, изображенный на рис.1. Отметим, что граф связей по всей структуре проще, чем граф потока сигналов, несмотря на то что он содержит дополнительную информацию относительно энергии и мощности.

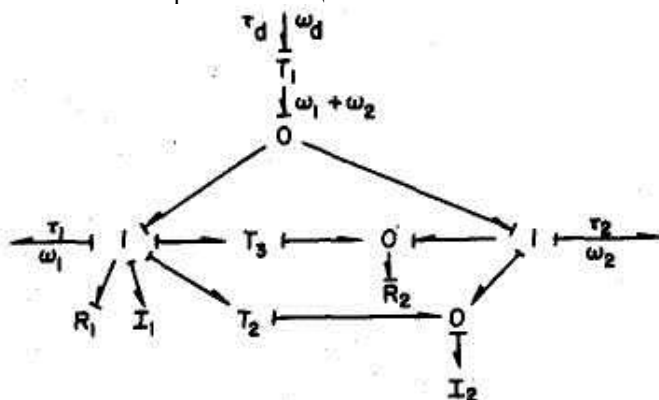


Рис.10. Дифференциальные причинные связи для графа связей на рис.9.

Пять этапов нахождения передаточной функции между τ_1 и ω_1 выглядят следующим образом:

1. Плоские циклы:

$$-\frac{I_2 D}{R_2}; \quad -\frac{I_1 D}{R_1}; \quad -\frac{T_2^2 I_2 D}{R_1}; \quad -\frac{T_3^2 I_2 D}{R_1}.$$

Четные открытые циклы: +2 цикла, передача каждого равна

$$+ \frac{T_2 T_3 I_2 D}{R_1}$$

(только одна из трех сетей порождает циклы).

2. Первые два плоских цикла, полученных из единственной пары независимых циклов.

Кононюк А.Е. Графы

$$\begin{aligned}
 3. \Delta = 1 + \frac{I_2 D}{R_2} + \frac{I_1 D}{R_1} + \frac{T_2^2 I_2 D}{R_1} + \frac{T_3^2 I_2 D}{R_1} - 2 \frac{T_2 T_3 I_2 D}{R_1} + \\
 + \frac{I_1 I_2 D^2}{R_1 R_2} = 1 + \left[\frac{I_1 + (T_2 - T_3)^2 I_2}{R_1} + \frac{I_2}{R_2} \right] D + \\
 + \left[\frac{I_1 I_2}{R_1 R_2} \right] D^2.
 \end{aligned}$$

4. Только один путь: $L_1 = I/R_1$.

5. Циклы, не касающиеся этого пути: только один (первый) из плоских циклов, полученных выше. Таким образом,

$$\omega_1 = - \frac{1}{R_1 \Delta} \left(1 + \frac{I_2 D}{R_2} \right) \tau_1.$$

Другие интересные передаточные функции можно найти путем соответствующей модификации 4-го и 5-го этапов.

Пример на рис.5 интересен тем, что граф связей здесь представляет собой общую сеть, в которой одна из ветвей — гиратор. Эта сеть имеет два открытых цикла противоположных знаков, и, согласно теореме 6, она нейтральна. Так как эти циклы касаются один другого и всех путей, то они не появятся ни в какой интересующей нас передаточной функции. Поэтому то, что передачи этих циклов равны $\pm RT/G$, причем цикл против часовой стрелки положителен (из теоремы 7), представляет для нас только академический интерес. *Пять* эталон выглядят так:

1. Плоские циклы:

$$\begin{aligned}
 - \frac{I_1 D}{G^2 C D} ; - \frac{R I_1 D}{G^2} \quad - \frac{R}{I_2 D} ; - \frac{I^2 R}{G^2 C D} ; \\
 - \frac{1}{C I_2 D^2}
 \end{aligned}$$

Открытые циклы нейтральны.

2. Все циклы зависимы (отчасти из-за нестандартности причинных связей).

$$3. \Delta = 1 + \frac{I_1}{G^2 C} + \frac{R I_1 D}{G^2} + \frac{R}{I_2 D} + \frac{T^2 R}{G^2 C D} + \frac{1}{C I_2 D^2}.$$

4. Два пути из f_1 в f_2 :

Кононюк А.Е. Графы

$$L_1 = \frac{RT}{G^2 CD}; L_2 = \frac{R}{G}.$$

5. Все циклы касаются этих путей. Таким образом,

$$f_2 = \frac{L_1 + L_2}{\Delta}.$$

Пример на рис.5 дает две простые сети: левую — четную (и, таким образом, сократимую с использованием эквивалентов рис.6) и правую — нечетную (и тем самым несокращаемую). Далее, правая сеть содержит активную связь, разрывающую один ее открытых циклов и один — из плоских. Активная связь сокращает также число пар независим циклов с девяти до трех. Используются интегральные причинные связи. Приводим пять этапов:

1. Плоские циклы:

$$-\frac{1}{IC_1 D^2} (2 \text{ цикла}); -\frac{1}{RC_2 D};$$

$$\frac{1}{RC_1 D}; -\frac{1}{IC_2 D^2}.$$

Четные открытые циклы $+\frac{1}{IC_1 D^2}$ (пара).

Нечетный открытый цикл: $-\frac{1}{RC_2 D}$ (только один).

2. Первый плоский цикл (но не его двойник) не касается плоского

цикла $-\frac{1}{RC^2 D}$ и открытого четного цикла (имеющего ту же самую циклическую передачу). Открытый нечетный цикл также не касается одного из открытых четных циклов.

3. Отметим значительное сокращение составляющих в Δ , приводящее к

$$\Delta = 1 + \frac{1}{RC_1 D} + \frac{1}{RC_1 D} + \frac{1}{IC_2 D^2} + \frac{1}{RIC_1 C_2 D^3}.$$

4. Имеем следующие два пути из e в f :

$$L_1 = -\frac{1}{R} \quad \text{и} \quad L_2 = \frac{T_1}{IC_2 R D^2}.$$

5. Путь L_1 не соприкасается с первым плоским циклом и с одним из четных открытых циклов. Путь L_2 касается всех циклов. Таким образом,

Кононюк А.Е. Графы

$$f = \frac{1}{\Delta} \left[-\frac{1}{R} \left(1 + \frac{1}{IC_1D^2} - \frac{1}{IC_1D^2} \right) + \frac{1}{IC_1RD^2} \right] e =$$

$$= \frac{1}{\Delta R} \left[-1 + \frac{1}{IC_1D^2} \right] e.$$

Большинство графов связей сложнее, чем вышеприведенные примеры, содержащие пропорционально меньше сетей и открытых циклов и больше односвязных импедансов и плоских циклов. Можно найти несколько удобных вычислительных схем для кодирования циклов и запоминания, если они независимы, и т. д. Типовой пример (рис.11) применен к модели системы, включающей электродвигатели постоянного тока, шестеренчатый насос, аккумулятор, нагрузку в виде поршня-цилиндра с нагружением пружиной-амортизатором и трубопроводы с несжимаемой жидкостью.

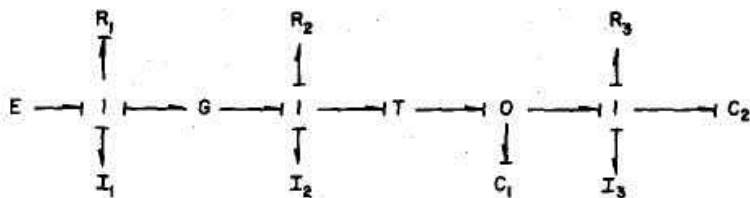


Рис.11. Типичный древовидный граф связей.

Предлагается самому проверить составляющие определителя графа

Кононюк А.Е. Графы

$$\begin{aligned}
 \sum_r p_r^{(1)} &= \frac{I_1 D}{R_1} + \frac{T_2 C_1 G^2 D}{R_1} + \frac{T_2 C_2 G^2 D}{R_1} + T^3 I_1 C_1 D^3 + \\
 &+ T^3 I_2 C_2 D^3 + T^3 R_2 C D + T^3 R_2 C_1 D + C_2 I_1 D^2 + R_2 C_2 D; \\
 \sum_r p_r^{(2)} &= \frac{I_1 D}{R_1} (T^3 I_1 C_1 D^3 + T^3 I_2 C_2 D^3 + T^3 R_2 C_1 D + T^3 R_2 C_2 D) + \\
 &+ \left(\frac{I_1 D}{R_1} + \frac{T^2 C_1 G^2 D}{R_1} + T^3 I_1 C_1 D^3 + T^3 R_2 C_1 D \right) \times \\
 &\times (C_2 I_1 D^2 + R_2 C_2 D); \\
 \sum_r p_r^{(3)} &= \frac{I_1 D}{R_1} (T^3 I_1 C_1 D^3 + T^3 R_2 C_1 D) (C_2 I_1 D^2 + R_2 C_2 D); \\
 \sum_r p_r^{(4)} &= 0;
 \end{aligned}$$

Для больших пассивных физических систем быстрее и гораздо надежнее применять правило циклов непосредственно к графу связей. Необходимо также запоминать, какие из четырех классов сетей имеют место в конкретном случае; строго аналогичные физические системы должны описываться подобными классами.

ПРИЛОЖЕНИЕ

Схемы доказательства теорем

Теорема 1. Так как две односвязные мощности направлены противоположно, в каскаде связей появляется нечетное число инверсий ориентации мощности последовательных связей. В случае отсутствия взаимных элементов эти инверсии возникают исключительно у 0- и 1-узлов. Каждая такая инверсия направления производит изменение знака точно в одном из двух потоков сигналов, проходящих через узел, как отмечалось ранее. Следовательно, нечетное число инверсий направлений всегда влечет за собой наличие отрицательного коэффициента передачи цикла.

Теорема 2. По определению взаимные элементы имеют противоположно направленные ориентации мощностей, и, следовательно, каждый такой элемент вызывает увеличение или уменьшение инверсий в ориентации мощностей узлов ровно на единицу, изменяя знак коэффициента передачи цикла.

Теорема 3. Если ориентация мощности сетевидной связи меняется, знак соответствующего усилия или потока также нужно изменить, с

тем чтобы физический смысл графа остался неизменным. Если меняется усилие, то оно должно распространиться через соседние 0-узлы, вызывая компенсирующее изменение ориентации мощности в сетевидной связи с другой стороны (или всех других связей, исходящих из 0-узла). Если же меняется поток, то изменение распространяется через 1-узел, соседний с другой стороны, изменяя ориентации мощности у его других сетевидных связей (плюс у всех других связей этого узла). Однако изменения усилий не распространяются вокруг 1-узлов, а изменения потока не распространяются вокруг 0-узлов. Следовательно, ориентации мощностей сетевидных связей в простых сетях могут меняться только попарно, не влияя в конечном итоге на четность или нечетность сети.

Теорема 4. Открытые циклы простых четных сетей, как не имеющие ответвлений, так и те, у которых все мощности ориентированы по часовой стрелке (или все против часовой стрелки), имеют коэффициент передачи $+1$. Изменение ориентации одной из связей (а тем самым изменение смысла системы) изменяет оба коэффициента передачи на -1 . Точнее, одно изменение возникает вокруг 0-узла, другое — вокруг 1-узла. Каждое следующее изменение ориентации мощности изменяет знаки обеих передач аналогично; следовательно, четные циклы этого класса имеют коэффициенты передачи $+1$, а нечетные циклы — коэффициенты передачи -1 . Если теперь причинная связь у одной из сетевидных связей меняется из-за воздействия 0- или 1-узлов, то у двух ответвлений появляются два открытых цикла, по одному с каждого конца измененной связи. Оба ответвления изменяют знак передачи одного и того же цикла, так что суммарный коэффициент передачи не меняется и знаки в пути другого цикла также остаются неизменными в предположении, что оба ответвления имеют положительные передачи путей.

Дальнейшие изменения в причинных связях сетевидных связей сохраняют равенство числа ответвлений, исходящих из 0-узлов, и равенство числа ответвлений, исходящих из 1-узлов, причем в конечном итоге знаки всех циклических передач остаются неизменными в предположении, что все ответвления имеют положительные передачи путей. Используя тот же аргумент, что и при доказательстве теоремы 1, увидим, что все ответвления будут иметь положительные передачи путей, если односвязные элементы у их концов определены как положительные (и мощность направлена внутрь элементов, чего требует стандартная форма записи).

Теорема 5. Достаточно показать, что любые допустимые изменения в ориентации мощностей или в причинных связях не влияют на коэффициенты передачи цикла. Как и в простых сетях, допустимые

Кононюк А.Е. Графы

изменения в ориентации мощностей трансформаторных ветвей появляются попарно, не влияя в целом на знаки коэффициентов передач цикла, независимо от причинных связей и от числа рассматриваемых сетевых ответвлений. Однако симметричные причинные связи гираторных ветвей допускают изменения ориентации мощностей в одной такой ветви при условии изоляции от остальной части сети. Например, $0 \rightarrow G \rightarrow 1$ изменяется на $0 \leftarrow G \leftarrow 1$ только при инвертировании знака потока в левосторонней связи и усилия — в правосторонней связи. (Противоположный выбор вызывает изменение ориентации мощностей ровно в двух других трансформаторных ветвях.) В результате такого изменения ориентации мощностей знаки в двух узлах одного из циклических путей меняются на обратные; ни в каком из других путей перемены знаков не происходит, и, следовательно, в итоге коэффициенты передачи цикла остаются прежними.

Следствия, вызываемые изменением ориентации трансформаторной ветви, аналогичны случаю простой сетевой ветви, рассмотренному при доказательстве теоремы 4, а именно изменение в подключении ровно двух ответвлений (одного к 0-узлу и другого к 1-узлу), не вызывает в конечном итоге изменений знака ни одного из коэффициентов передач (в предположении положительных значений передач в ответвлениях) хотя абсолютные величины коэффициентов передач и даже то, касаются ли два цикла, могут изменяться. Следствием изменения ориентации гираторной ветви на обратную (при сохранении, разумеется, симметрии ориентации) является аналогично одно изменение в подключении ответвления с каждой стороны ветви (непосредственно соседней или; если другие ориентации ветвей в свою очередь изменяются, более отдаленной). Оба ответвления в этих случаях исходят из ветвей того же типа при условии отсутствия других гираторных ветвей между этими ответвлениями. А это как раз то условие, благодаря которому дважды происходит перемена знака в циклических передачах, входящих в один и тот же цикл, взаимное уничтожение этих изменений и, следовательно, сохранение знака в каждом коэффициенте передач цикла. (Взаимные элементы исключаются точно таким же образом, так как они отличаются от гираторов лишь в одном отношении, а именно два изменения знака происходят в противоположных циклах благодаря одновременной симметрии ориентации мощностей и причинных связей. Четность, нечетность, а также нейтральность не играют при этом никакой роли).

Теорема 6. Включение гиратора в трансформаторную сетевидную ветвь вносит в конечном счете изменение в одно из сетевидных ответвлений, входящих в коэффициенты передачи открытых циклов,

Кононюк А.Е. Графы

приводя в свою очередь к изменению знака ровно одного из этих коэффициентов передачи. Следовательно, если оба знака были одинаковы перед включением гиратора (четный или нечетный вид связи), то теперь они становятся различными (нейтральный вид связи), и наоборот.

Теорема 7. Если ни одна из трансформаторных связей не направлена против часовой стрелки, так что число трансформаторных связей, направленных по часовой стрелке, нечетно, то коэффициент передачи по часовой стрелке положителен. Если одна из связей направлена против часовой стрелки, так что число трансформаторных связей, направленных по часовой стрелке, четно, то оба коэффициента передачи имеют противоположные знаки. Все другие комбинации ориентации мощностей могут быть получены из этих двух, без изменения как четности или нечетности числа трансформаторных связей, направленных по часовой стрелке, так и знаков каждого из коэффициентов передачи; необходимо только подходящим образом изменить ориентации мощностей в парах трансформаторных связей и в отдельных гираторных связях (см. доказательство теоремы 5).

1.9. Лагранжевы графы связей

Для систем, описываемых в терминах энергии и мощности, Е.Брауном найдены два типа графов связей, которые включают уравнения Лагранжа и обобщенные координаты. Основная форма содержит более простые уравнения и большее число первообразных переменных. Графы позволяют понять как энергетическую структуру системы, так и ее функциональное поведение, включая нелинейности, и обеспечивают вывод уравнений в пространстве состояний.

Обозначения

C — податливость или коэффициент податливости;

e, \mathbf{e} — обобщенная сила и ее вектор;

f, \mathbf{f} — обобщенная скорость и ее вектор;

G — гиратор или гираторный коэффициент;

I, I^* — инерция или инерционный коэффициент, коинерция;

L — функция Лагранжа;

P — мощность;

$$p = \int e dt \quad \text{— обобщенный импульс;}$$

$$q, \mathbf{q} = \int f dt \quad \text{— обобщенное перемещение и его вектор;}$$

R — сопротивление или коэффициент диссипации;

Кононюк А.Е. Графы

T — преобразователь или коэффициент преобразования;

t — время;

X, \mathbf{X} — первообразное квазиперемещение и его вектор;

x, \mathbf{x} — первообразная квазискорость и ее вектор;

V — обобщенная или квазипотенциальная энергия;

Z, \mathbf{Z} — первообразная квазисила и ее вектор;

z — обобщенная сила;

δ — символ вариационного исчисления;

ϵ — запасенная энергия;

τ, τ^* — обобщенная или квазикинетическая энергия, обобщенная кинетическая коэнергия;

Введение.

Уравнения Лагранжа являются мощным средством моделирования физических систем, поскольку они основаны на выражениях для энергии и мощности, содержащих только обобщенные перемещения и скорости. Альтернативное использование обобщенных принципов количества движения часто сложнее в силу того, что эти принципы включают обобщенные ускорения, такие, как ускорение Кориолиса.

Графы связей также служат мощным средством моделирования физических систем, так как благодаря графическому представлению запасов энергии и потоков мощности они позволяют охватить энергетическую структуру и функциональное поведение таких систем. Влияние неучета отдельных энергий или новых зависимостей можно обнаружить просто путем отсечения соответствующих частей графа или добавления соответствующих связей. Кроме того, граф связей перебрасывает мост к эффективным схемам аналогового или цифрового моделирования.

Эти оба метода объединены здесь в лагранжев граф связей, являющийся стандартным графом связей, который составлен и расположен по специальному способу, позволяющему сформировать его при помощи выражений для энергии и мощности. В случае необходимости из графа можно вывести уравнения Лагранжа и факт модуляции параметров. Для строго линейных систем модуляция отсутствует и параметры сохраняют постоянные значения. Итак, *структура* графа содержит всю информацию о модели (кроме значений параметров), если система линейна, или почти всю информацию, если система умеренно нелинейна. Для редких систем с сильной нелинейностью структура графа содержит относительно мало информации и граф связей малополезен. В любом случае структура графа правильно выявляет энергетическую пассивность, позволяя предотвратить ошибки, возможные при прямом обращении с

Кононюк А.Е. Графы

уравнениями Лагранжа. Граф может также составлять основу соответствующих схем моделирования.

Кэрноп рассмотрел множество ограничений и тонкостей, связанных с сохраняющими энергию преобразованиями, и дал соответствующую структуру графа связей, которая, будучи дополнена конкретной совокупностью гираторов, оказывается подобной основному лагранжеву графу связей. Использованный Брауном Е. здесь подход основан на новом выводе уравнений Лагранжа.

Вывод уравнений Лагранжа и основной лагранжев граф связей

Уравнения Лагранжа наиболее широко используются в механике, и большинство выводов начинается с сохранения импульса. Однако графы связей не привязаны к каким-либо конкретным формам энергии, и условие сохранение импульса и его аналогии предпочтительнее выводить из условия сохранения энергии.

Соотношения между различными механическими, электромеханическими и термодинамическими видами энергии, а также функциями состояния представлены во многих работах.

Первообразные координаты или обобщенные перемещения X_j , $j=1, 2, \dots, N$, выбраны таким образом, что запасенную энергию системы можно записать в виде

$$\epsilon = \sum_{j=1}^N [\tau_j(x_j) + V_j(X_j)] \quad (1a)$$

$$x_j = \frac{dX_j}{dt} \quad (1b)$$

где $\sum \tau_j$ и $\sum V_j$ называются *квазикинетической* и *квазипотенциальной* энергиями соответственно, хотя на самом деле они могут содержать электрическую, магнитную, тепловую или другие виды энергии, и в этом случае X_j фактически не является геометрическим перемещением.

Энергию τ_j может изменить действие консервативных *квасисил* Z_{cj} (не обязательно реальных «сил»), неконсервативных *квасисил* Z_{aj} (вынуждающих «сил», а также диссипативных «сил») и квазиреакций внутренней (например, геометрической) связи Z_{bj} ; поэтому условие сохранения энергии требует, чтобы соблюдалось следующее соотношение:

Кононюк А.Е. Графы

$$\frac{d\tau_j}{dt} = (Z_{aj} + Z_{bj} + Z_{cj})x_j. \quad (2)$$

Квазиреакции связи не совершают работы по всей системе на ее виртуальном перемещении $\delta X = \delta X_1, \delta X_2, \dots, \delta X_N$, удовлетворяющем связям. Итак,

$$\sum_{j=1}^N Z_{bj} \delta X_j = 0, \quad (3)$$

и из уравнения (2) можно вывести фундаментальное уравнение

$$\sum_{j=1}^N \left(\frac{1}{x_j} \frac{d\tau_j}{dt} - Z_{cj} - Z_{aj} \right) \delta X_j = 0, \quad (4)$$

которое оказывается полезным, так как не содержит квазиреакций связи.

Теперь введем ключевой интеграл

$$\int_{t_0}^{t_1} \left[\delta\tau^* + \sum_{j=1}^N (Z_{cj} + Z_{aj}) \delta X_j \right] dt,$$

в котором

$$\tau^* = \sum_{j=1}^N \tau_j^*(x_j)$$

называется квазикинетической коэнергией. Кроме размерности, она не имеет ничего общего с реальной энергией, но определена так, что интеграл обращается в нуль, когда вариации $\delta\tau^*$ и δX_j равны нулю на пределах $t=t_0, t_1$. Интегрирование по частям дает

$$\int_{t_0}^{t_1} \sum_{j=1}^N \left[\frac{d}{dt} \left(\frac{d\tau_j^*}{dx_j} \right) - Z_{cj} - Z_{aj} \right] \delta X_j dt = 0. \quad (5)$$

Сравнение уравнений (4) и (5) приводит к достаточному определяющему соотношению

$$\frac{1}{x_j} \frac{d\tau_j}{dt} = \frac{d^2\tau_j^*}{dx_j^2} \quad (6a)$$

или к выражению τ_j^* через τ_j

Кононюк А.Е. Графы

$$\tau_j^* = x_j \int_0^{x_j} \frac{\tau_j(y_j)}{y_j^2} dy_j, \quad (6b)$$

где y_i — переменная интегрирования. (Часто, как в классической динамике, $\tau_j \sim y_j^2$, поэтому $\tau_j^* = \tau_j$.)

Член $\sum_{j=1}^N Z_{cj} \delta X_j$ в уравнении (5) является вариацией работы консервативных квазисил, т. е. — δV .

Итак, после введения определения функции Лагранжа

$$L = \tau^* - V = \sum_{j=1}^N \tau_j^*(x_j) - \sum_{j=1}^N V_j(X_j) \quad (7)$$

уравнение (5) превращается в следующее:

$$\int_{t_0}^{t_1} \left(\delta L + \sum_{j=1}^N Z_{aj} \delta X_j \right) dt = 0. \quad (8)$$

Введем теперь лагранжевы *обобщенные* перемещения или координаты $\mathbf{q} = q_1, q_2, \dots, q_m$ и связанные с ними обобщенные скорости $\mathbf{f} = d\mathbf{q}/dt$.

(Определение *обобщенный* используется в классическом смысле минимально возможной совокупности переменных состояния и включает также содержание понятия *квазиординат*, использованного в первообразной совокупности переменных состояния). Отчасти это необходимо потому, что практически неудобно иметь дело с большим числом первообразных координат и соответствующих им запасов энергии, поэтому уравнение (1) мало полезно. Вывод из приведенных ниже уравнений (11), (12) требует, чтобы число лагранжевых координат m было по возможности минимально и равно числу степеней свободы системы. Даже в тех случаях, когда первообразные координаты полезны, обобщенные координаты допускают удобное представление внутренних связей.

Сосредоточим внимание в дальнейшем на *голономных* системах (неголономные системы, несвойственные инженерной практике, включают одну или несколько неинтегрируемых дифференциальных связей между x и q) для которых все обобщенные геометрические внутренние связи представлены в виде

$$X_j = X_j(\mathbf{q}) \quad j = 1, 2, \dots, N. \quad (9)$$

Итак, q и f образуют вместе достаточный вектор состояния. Из уравнений (9) имеем

Кононюк А.Е. Графы

$$\dot{x}_j \equiv \frac{dX_j}{dt} = \sum_i T_{ji}(\mathbf{q})f_i; \quad T_{ji}(\mathbf{q}) = \frac{\partial X_j}{\partial q_i}. \quad (10)$$

Второй член уравнения (8), представляющий вариацию работы внешних или неконсервативных сил, можно выразить через обобщенные силы $\mathbf{e} = e_1, e_2, \dots, e_m$ и вариации $\delta \mathbf{q}$, так что уравнение преобразуется в

$$\int_{t_0}^{t_1} \left(\delta L + \sum_{i=1}^m e_i \delta q_i \right) dt = 0; \quad (11a)$$

$$L = \tau^*(\mathbf{q}, \mathbf{f}) - V(\mathbf{q}). \quad (11b)$$

Оно выражает принцип Гамильтона, из которого традиционным способом можно вывести уравнения Лагранжа

$$\frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \mathbf{f})}{\partial \dot{f}_i} \right] - \frac{\partial L(\mathbf{q}, \mathbf{f})}{\partial q_i} = e_i. \quad (12)$$

Уравнения (1), (10), и (12) образуют скелет лагранжева графа связей, но требуют дальнейших преобразований.

Полезно ввести инерции I_j^* , коинерции I_j и податливости C_j при помощи соотношений

$$\tau_j = 1/2 I_j(x_j) x_j^2; \quad (13a)$$

$$\tau_j^* = 1/2 I_j^*(x_j) x_j^2; \quad (13b)$$

$$V_j = \frac{1}{2C_j(X_j)} X_j^2. \quad (13c)$$

Соотношение между T_j^* и I_j следует из уравнения (6b)

$$I_j^* = \frac{1}{x_j} \int_0^{x_j} I_j(y_j) dy_j. \quad (14)$$

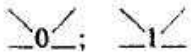
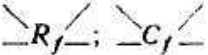

Частный случай $I_j = \text{const}$, дающий $I_j^* = I_j$, включает нерелятивистскую механику. Во втором частном случае $T_{ji} = \text{const}$, поэтому τ_j и I_j не являются функциями \mathbf{q} . Третий частный случай представляет комбинацию первых двух совместно с условием $C_j = \text{const}$ и является случаем линейной системы, для которой граф отражает полную структуру, за исключением постоянных значений параметров.

Кононюк А.Е. Графы

Использованная символика символика графа связей представлена в табл. 1 раздела 1.8. Линейные соотношения этой таблицы заменены в настоящей работе нелинейными (табл. 1 и 2).

Таблица 1.

Изменения, допускаемые в элементах лагранжевых графов связей

Символ графа связей	Основной лагранжевый граф связей	Альтернативный лагранжевый граф связей
$E-$; $F-$ 	Те же, что и в линейном случае	
$\rightarrow R_i$	$R_i = R_i(f_i)$	$R_{ci} = R_{ci}(f_i, q)$
$\rightarrow C_i$	$C_i = C_i(q_i)$	$C_{qi} = C_{qi}(q)$
$\rightarrow I_i$	$I_i = I_i(f_i)$	$I_{fi} = I_{fi}(f_i, q)$
$\xrightarrow{f_i} T_{ij} \xrightarrow{f_j}$	$T_{ij} = T_{ij}(q)$	$T_{ij} = T_{ij}(q, f)$
$\rightarrow G_{ij} \leftarrow$	$G_{ij} = G_{ij}(q, f)$	$G_{ij} = G_{ij}(q, f)$
$\rightarrow R_{ij} \leftarrow$	Не используются	$R_{cij} = R_{cij}(q)$
$\rightarrow C_{ij} \leftarrow$		$C_{qij} = C_{qij}(q')$ (см. примечание)
$\rightarrow I_{ij} \leftarrow$		$I_{fij} = I_{fij}(q)$
 	Обобщенные многосвязные поля не используются	

Примечание. q' есть q без q_i и q_j .

Различные определения параметров сопротивления, податливости и инерции

Элемент	Аппроксимация по касательной	Аппроксимация по хорде	Мощность или энергия	
			по Лагранжу	дуальная
Сопротивление	$R_1 = de/df$	$R_c = e/f$	$R = \frac{P}{f^2} = R_c$	$R = \frac{e^2}{p} = R_c$
Податливость	$C_1 = dq/de$	$C_e = q/e$	$C_q = \frac{q^2}{2\epsilon}$	$C_p = \frac{2\epsilon}{e^2}$
Инерция	$I_1 = dp/df$	$I_c = p/f$	$I_f = \frac{2\epsilon}{f^2}$	$I_p = \frac{p^2}{2\epsilon}$

Структура графа, показанная на рис.1, содержит преобразователи с коэффициентами, определяемыми уравнением (9b), и гираторы, рассматриваемые ниже.

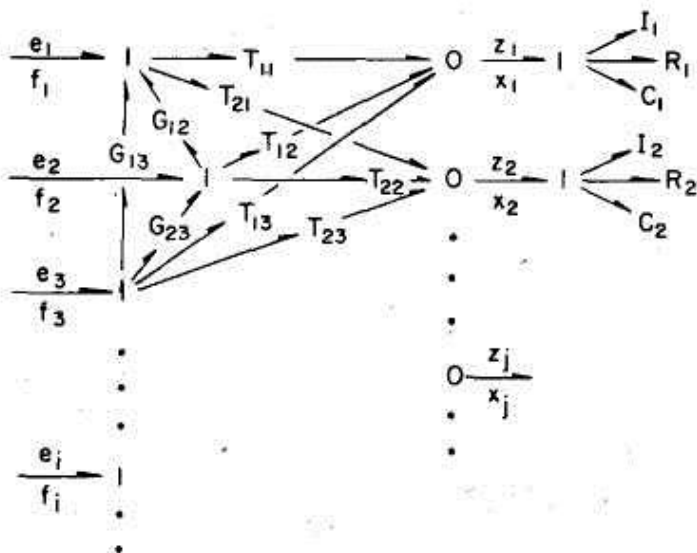


Рис.1. Основной лагранжев граф связей.

Кононюк А.Е. Графы

Структура является гибкой, так как для заданной системы n -го порядка полное число C_j и I_j -элементов может превышать n (но только за счет усложнений вычислительных причинно-следственных отношений) и быть ниже n (но только ценой необязательной нелинейности коэффициентов преобразователей и гираторов в остальном линейных систем).

Суммирование обобщенных сил (усилий) вокруг каждого из левых 1-узлов дает уравнения Лагранжа. Итак, уравнение (12) используется для определения коэффициентов гираторов. Сумма усилий на гирационных связях 1-узла для f_i после подстановки уравнений (10) и (13) - (14) будет равна

$$e_{Gi} = \sum_k \sum_j x_j \left(I_j^* - \frac{x_j}{2} \frac{dI_j^*}{dx_j} \right) \left(\frac{\partial T_{jk}}{\partial q_i} - \frac{\partial T_{ji}}{\partial q_k} \right) f_k +$$

$$+ \sum_j T_{ji} \left[I_j - I_j^* + x_j \left(\frac{1}{2} \frac{dI_j}{dx_j} - 2 \frac{dI_j^*}{dx_j} \right) - \right.$$

$$\left. - \frac{x_j^2}{2} \frac{d^2 I_j^*}{dx_j^2} \right] \frac{dx_j}{dt}. \quad (15)$$

С помощью уравнения (14) можно показать, что член в квадратных скобках тождественно равен нулю. Остаток правой части уравнения (15) также обращается в нуль, когда $k=i$, что позволяет представить это уравнение в форме

$$e_{Gi} = \sum_{k \neq i} G_{ik} f_k \quad (16)$$

$$G_{ik} = \sum_j \left(I_j^* - \frac{x_j}{2} \frac{dI_j^*}{dx_j} \right) x_j \left(\frac{\partial T_{jk}}{\partial q_i} - \frac{\partial T_{ji}}{\partial q_k} \right). \quad (17)$$

Это значение G_{ik} в точности является коэффициентом гиратора, связывающего 1-узлы f_k и f_i (с условием, что поток мощности направлен в сторону узла f_i). Заметим, что

$$G_{ik} = -G_{ki}, \quad (18)$$

Данное условие необходимо для того, чтобы гиратор сохранял, но не запасал энергию. Заметим, далее, что все гираторы исчезают в вышеупомянутом втором частном случае, включающем линейные системы, и что, как правило, C_i не влияют или не дают вклада в G_{ik} .

Кононюк А.Е. Графы

Диссипацию энергии в этой схеме можно учесть выражением

$$P_d = \sum_j P_{d_j}(x_j) = \sum_j R_j(x_j)x_j^2, \quad (19)$$

где R_j — *сопротивление*, аппроксимируемое хордой, постоянное в линейных системах. Эти сопротивления введены в схему на рис.1. Скорости x_j , входящие в диссипативную функцию, могут совпадать или не совпадать со скоростями сохраняющих энергию частей системы. Никакие из нелинейностей P_{d_j} не дают вклада в коэффициенты гиратора, что сходно с характером поведения V_j , но отлично от поведения τ_j . Вклад, связанный с τ_j , следует из дифференцирования по времени, необходимого для вычисления z_j из $\tau_j(x_j)$ и x_j . Вместо дифференцирования в P_{d_j} входит статическая связь, и в V_j — интегрирование по времени.

Пример 1

Пример представлен на рис.2.

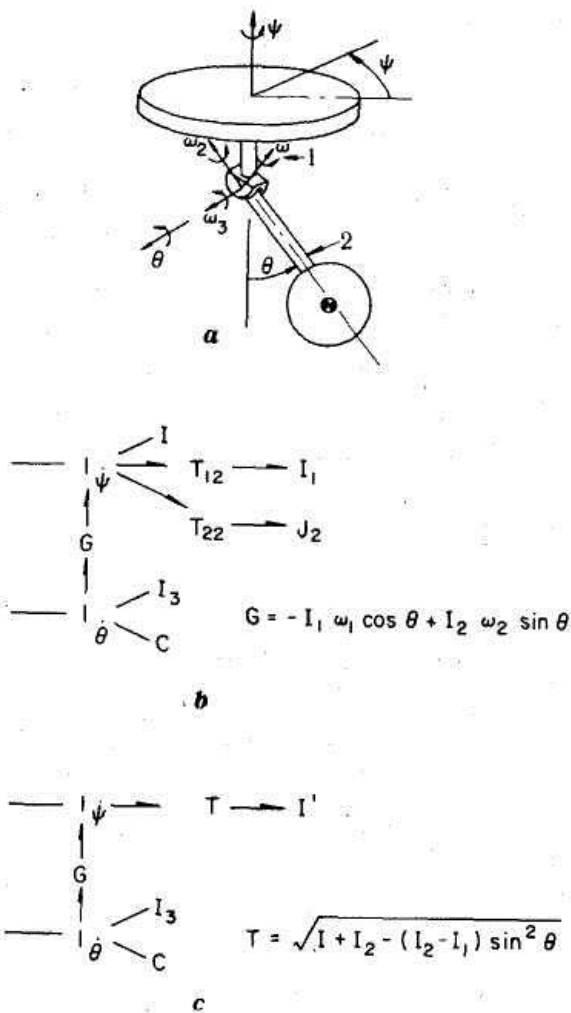


Рис.2. Пример 1.

a—схема; *b*—основной лагранжев граф связей; *c*—упрощенный граф.
 1— подшипники вала не показаны; 2 — I_1, I_2, I_3 — главные моменты инерции относительно шарнира.

Энергии системы таковы:

Кононюк А.Е. Графы

$$\tau = \frac{1}{2}I_1\omega_1^2 + \frac{1}{2}I_2\omega_2^2 + \frac{1}{2}I_3\omega_3^2 + \frac{1}{2}I\dot{\psi}^2$$

$$V = -mgl \cos \theta$$

и геометрические связи имеют форму

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \\ T_{31} & T_{32} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & \sin \theta \\ 0 & \cos \theta \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Соответствующий граф связей, изображенный на рис.2, *b*, строится непосредственно; коэффициент гириатора определяется из уравнения (17). Общий метод Кэрнопа дает такую же структуру преобразователя, но не выявляет возможности существования гириаторов. В этом примере точный, но сравнительно сложный граф связей выводится относительно трудоемким применением принципа сохранения импульса. Значительно более простой граф на рис.2, *c* вытекает из прямого суммирования энергий инерционных звеньев I_1 , I_2 и I_3 в энергию $\frac{1}{2}I^1(T\dot{\psi})^2$ с произвольным выбором $I^1 = 1$. Этот граф прост для понимания, отражает особенности системы и приводит к соответствующим схемам моделирования.

Альтернативные лагранжевы графы связей

Иногда преобразование $X(q)$, позволяющее представить энергию в форме уравнения (8), найти нелегко. В этих случаях можно выйти из затруднения путем использования элементов графа связей с повышенной гибкостью, перечисленных в табл. 1. Общая структура получающегося альтернативного лагранжева графа связей показана на 3, *a*. Переменные x основного графа не используются, хотя это обычно достигается ценой повышения сложности моделирования, что будет показано ниже.

Первым шагом в общей методике моделирования является определение выражений для запасенной энергии и рассеиваемой мощности с последующим их представлением при помощи соответствующих элементов графа связей. Итак, нелинейные запоминающие элементы обычно идентифицируются их соответствующими *энергетическими* определениями вместо их аппроксимаций по хорде или по касательной и диссипативные элементы — рассеиваемой мощностью или аппроксимацией по хорде вместо аппроксимации по касательной. Переход к другим определениям может потребоваться в процессе проведения моделирования.

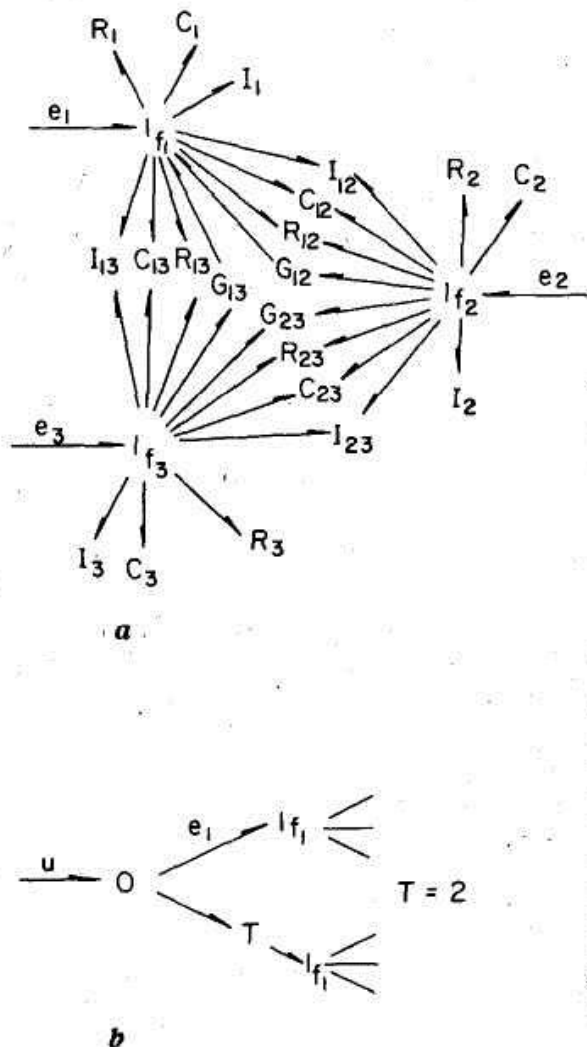


Рис.3. Альтернативный лагранжев граф связей для системы с тремя степенями свободы.

a — общая форма без полной структуры входа; *b* — пример структуры входа.

Четыре различных определения податливости и инерции, а также два определения сопротивления приведены в табл.2. Аппроксимации по

Кононюк А.Е. Графы

касательной и хорде (за исключением R_c) обычно неудобны для моделирования нелинейных элементов на базе выражений энергии и мощности, поскольку они часто не представимы в неоператорной форме. Однако они существуют в простой форме для элементов основного лагранжева графа связей, допускающего в этом случае подстановки аппроксимаций по хорде и касательной как удобных для численного моделирования, что будет показано ниже.

Альтернативные определения различных элементов дают следующие результаты для обобщенных усилий соответствующих связей: сопротивление

$$e_i = R_{ci}(f_i, \mathbf{q}) f_i, \quad (20)$$

податливость

$$e_i = \left[\frac{1}{C_{qi}} + \frac{q_i}{2} \frac{\partial (1/C_{qi})}{\partial q_i} + \sum_{j \neq i} \frac{q_j}{2} \frac{\partial (1/C_{qi})}{\partial q_j} \right] q_i, \quad (21)$$

инерция

$$e_i = I_{fi} \frac{df_i}{dt} + \frac{f_i}{2} \left(\frac{\partial I_{fi}}{\partial f_i} \frac{df_i}{dt} + \sum_j \frac{\partial I_{fij}}{\partial q_j} \right). \quad (22)$$

преобразователь

$$f_j = T_{ji}(\mathbf{q}, \mathbf{f}) f_i; \quad e_i = T_{ji} e_j, \quad (23)$$

гиратор

$$e_i = G_{ji}(\mathbf{q}, \mathbf{f}) f_j; \quad e_i = -G_{ji} f_j, \quad (24)$$

взаимное сопротивление

$$e_i = R_{aij} f_j. \quad (25)$$

взаимная податливость

$$e_i = \frac{1}{C_{qij}} q_j + \frac{q_i q_j}{2} \sum_k \frac{\partial (1/C_{qij})}{\partial q_k} \frac{f_k}{f_i}, \quad (26)$$

взаимная инерция

$$e_i = I_{fij} \frac{df_j}{dt} + \sum_k \frac{\partial I_{fij}}{\partial q_k} \frac{f_i f_k}{2}. \quad (27)$$

Кононюк А.Е. Графы

Для частных случаев линейных систем с постоянными коэффициентами взаимные элементы $-R$, $-C$ и I — введенные ранее. Их использование не обязательно, но может быть весьма полезным.

Предполагается, что мощность, поступающая в i -ю связь, равна $e_i f_i$ и что вся мощность подводится через полусвязи (т. е. модуляции являются единственным регулятором потока мощности через реальные энергетические связи). Взаимные элементы симметричны (т. е. $P_i = P_j$), и мощность, рассеиваемая на $-R_{ii}$, равна $2R'_{cij} f_i f_j$.

Число степеней свободы системы должно быть равно числу обобщенных перемещений, в противном случае внутренние связи не полностью представлены в энергетических выражениях. Возможны многие различные идентификации \mathbf{f} и \mathbf{q} , в частности, если система существенно нелинейна. Наилучший выбор обычно соответствует несущественной нелинейности (модуляции параметров).

Второй шаг в процессе моделирования состоит в определении выражения для входной или граничной мощности:

$$P = \sum_i e_i f_i. \quad (28)$$

Определенный таким образом вектор \mathbf{e} может быть функцией одной или нескольких переменных; взаимосвязь представлена структурой узлов и связей. Например, если входная мощность

$$P = (f_1 + 1/2 f_2) u,$$

где u — независимая переменная, то

$$\mathbf{e}_1 = \mathbf{u}; \mathbf{e}_2 = \mathbf{u}/2,$$

и имеет место структура, приведенная на рис.3, *b*. Конечный шаг заключается в определении коэффициентов гираторов. Как и прежде, используем уравнения Лагранжа для установления этих коэффициентов и достаточности построенного графа, поэтому необходимо знать соотношение между τ^* и τ , чтобы найти функцию Лагранжа L . Заметим, что

$$\begin{aligned} \sum_i \frac{\partial L(\mathbf{q}, \mathbf{f})}{\partial f_i} f_i &= \sum_i \sum_j \frac{d\tau_j^*}{dx_j} \frac{\partial x_j}{\partial f_i} f_i = \sum_j \frac{\tau_j^* + \tau_j}{x_j} \sum_i \frac{\partial X_i}{\partial q_i} \frac{dq_i}{dt} = \\ &= \sum_j [\tau_j^*(x_j) + \tau_j(x_j)] = \sum_j [\tau_j^*(\mathbf{q}, \mathbf{f}) + \tau_j(\mathbf{q}, \mathbf{f})]. \quad (29) \end{aligned} \quad (29)$$

Конюк А.Е. Графы

(Уравнение (30) можно использовать для вывода хорошо известного преобразования Лежандра).

В частности, для предполагаемой функции Лагранжа

$$L = \sum_i [1/2 I_i^*(f_i, \mathbf{q}) f_i^2 - V_i(\mathbf{q})] + \sum_{i, j \neq i} [I_{ij}^*(\mathbf{q}) f_i f_j - V_{ij}(q_{ij}, \mathbf{q}')] \quad (30)$$

уравнение (29) дает соотношение

$$\sum_i \left[\left(I_i^* + \frac{\partial I_i^*}{\partial f_i} \frac{f_i}{2} \right) f_i^2 + 2 I_{ij}^* f_i f_j \right] = \sum_i [1/2 I_i^* f_i^2 + 1/2 I_j f_i^2 + I_{ij}^* f_i f_j + I_{ij} f_i f_j], \quad (31)$$

которое удовлетворяется при условии

$$\frac{\partial I_i^*}{\partial f_i} f_i - I_i^* = I_i \quad (32)$$

и

$$I_{ij}^* = I_{ij}; \quad \tau_{ij}^* = \tau_{ij}. \quad (33)$$

Равенства (33) отражают основное свойство линейности взаимных инерций относительно потока мощности. Равенство (32) дает следующее соотношение:

$$I_i^* = \frac{1}{f_i} \int_0^{f_i} [I_i(f_i', \mathbf{q})]_0 df_i', \quad (34)$$

з котором f_i' — переменная интегрирования.

Сумма всех усилий гираторов G_{ij} , действующих на i -й 1-узел, равна

$$\sum_j e_{G_{ij}} = e_{G_i} + e_{I_i} + \sum_{j \neq i} e_{G_{ij}} + \sum_{j \neq i} e_{I_{ij}} - e_i. \quad (35)$$

После многократных подстановок предыдущих результатов и ряда сложных сокращений подобных членов уравнение (35) сводится к следующему:

Кононюк А.Е. Графы

$$\sum_{j \neq i} e_{G_{ij}} = \sum_{j \neq i} G_{ij} f_j \quad (36)$$

$$G_{ij} = 1/2 \left[\left(\frac{\partial I_{fj}^*}{\partial q_i} f_j - \frac{\partial I_{fi}^*}{\partial q_j} f_i \right) + \right. \\ \left. + \left(\sum_{k \neq j} \frac{\partial I_{fjk}}{\partial q_i} f_k - \sum_{k \neq i} \frac{\partial I_{fik}}{\partial q_j} f_k \right) + \right. \\ \left. + \left(\frac{\partial(1/C_{qi})}{\partial q_j} \frac{q_i^2}{f_i} - \frac{\partial(1/C_{qj})}{\partial q_i} \frac{q_j^2}{f_j} \right) + \right. \\ \left. + \sum_{k \neq i, j} \left(\frac{\partial(1/C_{qik})}{\partial q_j} \frac{q_i q_k}{f_i} - \frac{\partial(1/C_{qjk})}{\partial q_i} \frac{q_j q_k}{f_j} \right) \right]. \quad (37)$$

Заметим, что, как и прежде,

$$\forall G_{ij} = -G_{ji}. \quad (38)$$

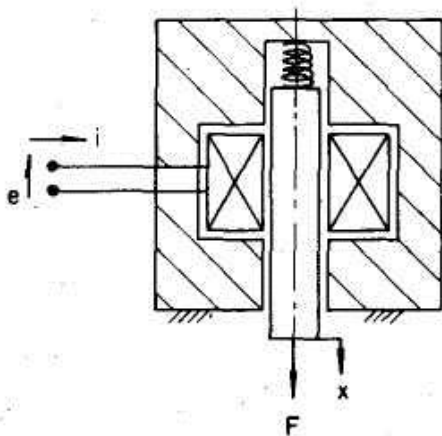
Диссипативные элементы никогда не требуют гираторов. Гираторы перераспределяют *запасенную* энергию, отсутствующую в диссипативных элементах, чтобы надлежащим образом компенсировать модуляцию параметров.

Пример 2

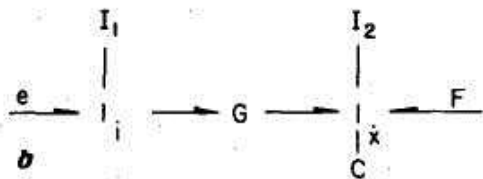
Индуктивность с переменной геометрией (соленоид), показанная на рис.4, обладает энергией, определяемой следующим образом:

$$\epsilon = 1/2 I(i, x) i^2 + 1/2 m \dot{x}^2 + 1/2 k (x - x_0)^2$$

где учтено магнитное насыщение.



a



$$I_1 = I; I_1^* = \frac{1}{i} \int_0^i [I(i', x)] di'$$

$$I_2 = I_2^* = m$$

$$C = \frac{1}{k}$$

$$G = \frac{1}{2} \left[\frac{\partial I_1^*}{\partial x} \right] = \frac{1}{2i} \int_0^i \left[\frac{\partial I(i', x)}{\partial x} \right] di'$$

Рис.4. Пример 2, индуктивность с переменной геометрией (соленоид). *a* — схема; *b* — альтернативный лагранжев граф связей.

Но это выражение не имеет формы уравнения (1), необходимой для построения основного лагранжева графа связей, и в общем не может

Кононюк А.Е. Графы

быть преобразовано к указанной форме, за исключением случая, когда функция I разложима в произведение $I_1(i)I_2(x)$. Однако альтернативный граф связей, показанный на рис.4, b , изображается без особого труда. Входная мощность имеет форму

$$P = ie + F\dot{x}$$

и диссипация не учитывается.

Формулировка уравнений в переменных состояниях

Переменными состояниями, необходимыми для основного лагранжева графа связей, являются различные X_i для податливостей и x_i для инерций. При построении альтернативного лагранжева графа связей преобразования переменных не применяются, поэтому приходится использовать \mathbf{q} и \mathbf{f} в качестве переменных состояний. В этом последнем случае уравнениями состояния будут непосредственно уравнения Лагранжа, которые сложнее, чем уравнения в преобразованных переменных.

Рекомендуется использовать в качестве переменных состояний вектор \mathbf{q} для податливостей и вектор $\mathbf{p} = \int e dt$ для инерции, предполагая, что статические соотношения имеют форму

$$\text{инерции } f_i = \psi_i(\mathbf{q}, \mathbf{p}), \quad (39)$$

$$\text{податливости } e_j = \psi_j(\mathbf{q}, \mathbf{p}). \quad (40)$$

Уравнения состояния выводятся просто и систематически, что позволяет рекомендовать эту методику для применения. К сожалению, уравнения Лагранжа содержат принципиально отличные переменные состояния, что не позволяет использовать эту методику для построения альтернативных лагранжевых графов связей. Однако методика применима к основному лагранжеву графу связей с небольшим видоизменением, заключающимся в обращении соотношений для инерции вида

$$p_i = \psi_i(\mathbf{q}, f_i). \quad (41)$$

На практике уравнения состояния могут быть записаны путем анализа графа связей, в котором отражена причинная последовательность, как показано в следующем примере. Используются интегральные причинно-следственные отношения.

Для альтернативного лагранжева графа связей уравнения состояния выводятся из следующих уравнений:

$$\mathbf{E} \frac{d\mathbf{f}}{dt} = -\mathbf{F}\mathbf{f} - \mathbf{H}\mathbf{q} + \mathbf{A}\mathbf{u} \quad (42a)$$

Кононюк А.Е. Графы

$$\frac{d\mathbf{q}}{dt} = \mathbf{f}, \quad (42b)$$

где

$$E_{ii} = I_i + \frac{\partial I_i}{\partial f_i} \frac{f_i}{2} \quad (43a)$$

$$E_{ij} = I_{ij} \quad (43b)$$

$$F_{ii} = \frac{\partial I_i}{\partial q_i} \frac{f_i}{2} + R_i \quad (43c)$$

$$F_{ij} = 1/2 \left[\left(\frac{\partial I_i^*}{\partial q_j} + \frac{\partial I_i}{\partial q_j} \right) f_i - \frac{\partial I_j^*}{\partial q_i} f_j + \right. \\ \left. + \sum_{k \neq i} \frac{\partial I_{ik}}{\partial q_j} f_k - \sum_{k \neq j} \frac{\partial I_{jk}}{\partial q_i} f_k + \sum_k \frac{\partial I_{ij}}{\partial q_k} f_k \right] + R_{ij} \quad (43d)$$

$$H_{ii} = \frac{1}{C_i} + \frac{\partial(1/C_i)}{\partial q_i} \frac{q_i}{2} \quad (43e)$$

$$H_{ij} = \frac{1}{C_{ij}} + 1/2 \left[\frac{\partial(1/C_j)}{\partial q_i} q_j + \sum_{k \neq i, j} \frac{\partial(1/C_{jk})}{\partial q_i} q_k \right]. \quad (43f)$$

Преобразование уравнений (42) к форме, выраженной через переменные состояния, выполнить несложно. Если матрица \mathbf{E} обратима, вектор состояния будет $\begin{bmatrix} \mathbf{f} \\ \mathbf{q} \end{bmatrix}$.

Однако обычно одна или несколько I_i и соответствующие I_{ij} обращаются в нуль, поэтому соответствующие f_i нужно выбросить из вектора состояния, и соответствующее уравнение внутри уравнения (42a) становится вспомогательным для определения f_i .

Пример 3

Устройство, изображенное на рис.5, a, b , может быть применено в качестве скоростного гироскопа (оно эффективно служит многим насекомым).

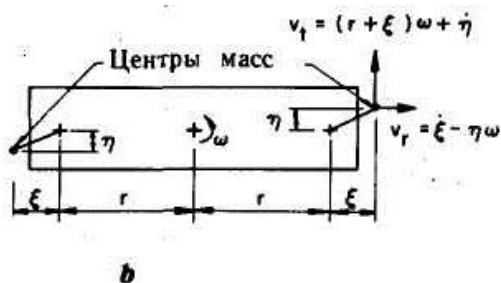
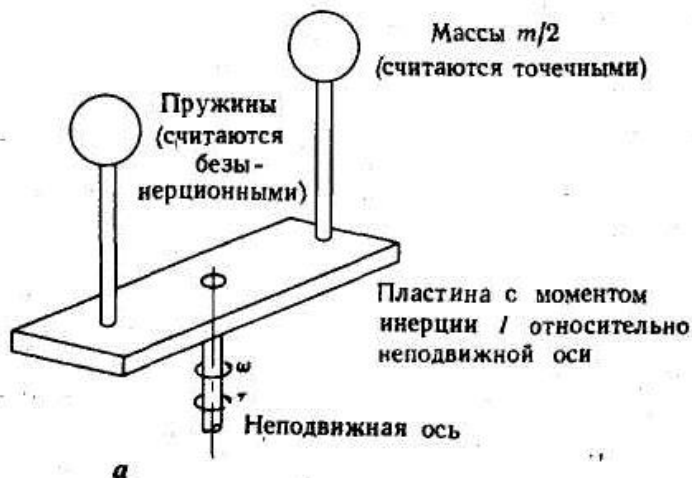


Рис.5. Пример 3.

a — схема; *b* — вид сверху, показывающий перемещения и скорости.

Предполагается, что геометрия и начальные условия симметричны, поэтому описаны только два симметричных перемещения (степени свободы). Считая массы точечными, а пружины линейными и безынерционными, можно записать потенциальную и кинетическую энергии

Конюк А.Е. Графы

$$V = 2[1/2 k_r \xi^2 + 1/2 k_t \eta^2];$$

$$\tau = \tau^* = 1/2 I \omega^2 + 2 \left[1/2 \left(\frac{m}{2} \right) \right] v^2;$$

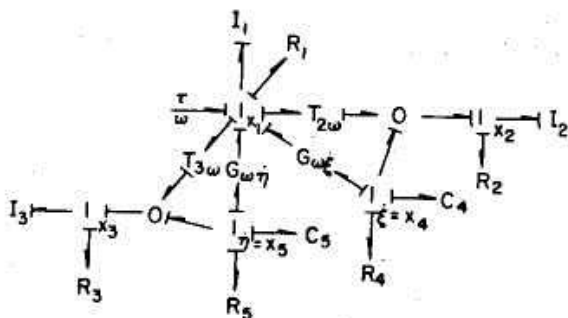
$$v^2 = v_r^2 + v_t^2 = [(\tau + \xi)\omega + \dot{\eta}]^2 + (\dot{\xi} - \eta\omega)^2,$$

и представить их в следующей форме:

$$V = \frac{1}{2C_r} \xi^2 + \frac{1}{2C_t} \eta^2;$$

$$\tau = \tau^* = 1/2 I_1 \omega^2 + 1/2 I_2 v_r^2 + 1/2 I_3 v_t^2.$$

Совместно с наиболее общей возможной линейной диссипативной функцией эти уравнения отражены при помощи основного лагранжева графа связей на рис.6, а. Коэффициенты гираторов следуют из уравнения (17).



$$I_1 = I; I_2 = I_3 = m$$

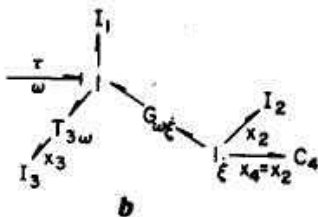
$$\frac{1}{C_4} = 2k_r; \frac{1}{C_5} = 2k_t$$

$$T_{2\omega} = -\eta; T_{3\omega} = r + \xi$$

$$G_{\omega\xi} = -m[(r + \xi)\omega + \eta] = -m x_3$$

$$G_{\omega\eta} = m(\xi - \omega\eta) = m x_2$$

a



b

Рис.6. Пример 3, продолжение.

a — основной лагранжев граф связей; b — упрощенный граф для случая нулевой диссипации и $k_r \rightarrow \infty, \eta \rightarrow 0$.

Для применения устройства в качестве скоростного гироскопа податливость C_η делают настолько малой (путем использования широкой полосы для пружин), что перемещение η и энергия $(1/2C_\eta)\eta^2$ пренебрежимо малы. Итак, C_t и $T_{\eta\omega}$ можно исключить из структуры

Кононюк А.Е. Графы

графа, коэффициенты $G_{\xi\omega}$ и $G_{\omega\eta}$ упрощаются и приводят к графу связей, показанному на рис.6, *b*. Теперь движение ξ возбуждается на собственной высокой частоте $\sqrt{T_2 C_\xi}$ неизообразимым внешним приводом. При постоянном вращении с угловой скоростью ω момент τ фактически пропорционален известной синусоидальной функции времени ξ и величине ω , подлежащей определению.

Итак, измерение τ (датчиком деформации) дает информацию об ω , превращая устройство в скоростной гироскоп. Сущность его поведения выражает гирационная связь $G_{\xi\omega}$.

На рис.6, *a* показаны причинные последовательности, дающие представление об интегральных причинно-следственных отношениях. В случае когда отдельные x_j не имеют соответствующих инерций (как, например, x_4 и x_5), эти x_j не будут переменными состояниями и должны быть связаны с другими x . Следуя причинным последовательностям, будем иметь

$$x_4 = x_2 - T_{2\omega} x_1;$$

$$x_5 = x_3 - T_{3\omega} x_1.$$

Остальные x_j имеют z_j , которые могут быть выражены при помощи причинно-следственных отношений через x и e :

$$\begin{bmatrix} 1 & 0 & 0 & -T_{2\omega} & -T_{3\omega} \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix} = \begin{bmatrix} 0 & G_{\omega\xi} & G_{\omega\eta} & 0 & 0 \\ -G_{\omega\xi} & 0 & 0 & 0 & 0 \\ -G_{\omega\eta} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} \tau \\ 0 \\ 0 \end{bmatrix}$$

Вектор z далее связан с x и x посредством аппроксимаций инерции по касательной и податливостей по хорде. В настоящей задаче инерции и податливости будут постоянны

Конюк А.Е. Графы

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix} = \begin{bmatrix} I_1 & 0 & 0 & 0 & 0 \\ 0 & I_2 & 0 & 0 & 0 \\ 0 & 0 & I_3 & 0 & 0 \\ 0 & 0 & 0 & 1/C_4 & 0 \\ 0 & 0 & 0 & 0 & 1/C_5 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ X_4 \\ X_5 \end{bmatrix}$$

В результате вышеприведенные уравнения дают

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ X_4 \\ X_5 \end{bmatrix} = \begin{bmatrix} 0 & \frac{G_{\omega \dot{x}}}{I_1} & \frac{G_{\omega \ddot{x}}}{I_1} & -\frac{T_{2\omega}}{C_4 I_1} & -\frac{T_{3\omega}}{C_5 I_1} \\ -\frac{G_{\omega \dot{x}}}{I_2} & 0 & 0 & -\frac{1}{C_4 I_2} & 0 \\ -\frac{G_{\omega \dot{x}}}{I_2} & 0 & 0 & 0 & -\frac{1}{C_5 I_3} \\ -T_{2\omega} & 1 & 0 & 0 & 0 \\ -T_{3\omega} & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ X_4 \\ X_5 \end{bmatrix} + \begin{bmatrix} 1/I_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tau,$$

После вывода выражений модулированных параметров через X и x этот результат можно преобразовать к более простой форме

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ X_4 \\ X_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \left[k_t r + \frac{X_4}{I_1} (k_t - k_r) \right] \\ x_3 & 0 & 0 & -\frac{2k_r}{m} & 0 \\ -x_2 & 0 & 0 & 0 & -\frac{2k_t}{m} \\ 0 & 1 & 0 & 0 & x_1 \\ -(\tau + X_4) & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ X_4 \\ X_5 \end{bmatrix} + \begin{bmatrix} 1/I_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tau,$$

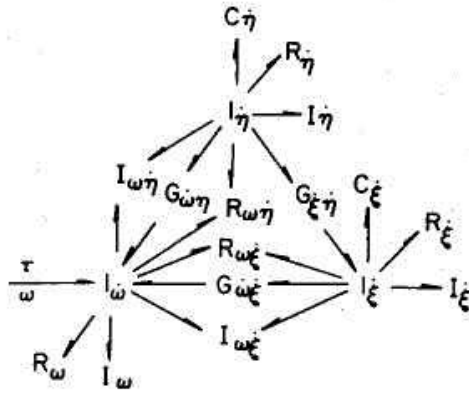
Кононюк А.Е. Графы

которая требует только пяти умножений переменных.

Эквивалентный альтернативный лагранжев граф связей, показанный на рис.7, а, строится при помощи того же выражения для V , но с привлечением более детального выражения для τ через

f -переменные ω , $\dot{\eta}$ и $\dot{\xi}$:

$$\tau = \tau^* = 1/2 I_{\omega} \omega^2 + 1/2 I_{\dot{\eta}} \dot{\eta}^2 + 1/2 I_{\dot{\xi}} \dot{\xi}^2 + I_{\omega \dot{\eta}} \omega \dot{\eta} + I_{\omega \dot{\xi}} \omega \dot{\xi}.$$



$$I_{\omega} = I + m[(r+\xi)^2 + \eta^2];$$

$$I_{\eta} = I_{\xi} = m;$$

$$I_{\omega\eta} = m(r+\xi); \quad C_{\xi} = \frac{1}{2k_r};$$

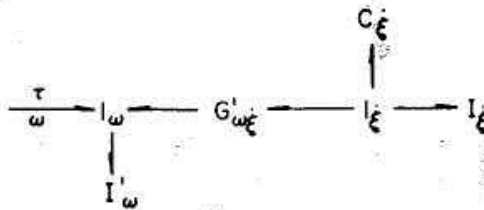
$$I_{\omega\xi} = -m\eta; \quad C_{\eta} = \frac{1}{2k_t};$$

$$G_{\omega\eta} = -m\omega\eta + \frac{1}{2}m\xi;$$

$$G_{\omega\xi} = -m(r+\xi) - \frac{1}{2}mn;$$

$$G_{\xi\eta} = m\omega.$$

a



b

Рис.7. Пример 3, продолжение.

a — альтернативный лагранжев граф связей; b — упрощенный граф для случая нулевой диссипации и $G_{\eta} = 0$.

Кононюк А.Е. Графы

Гираторные коэффициенты определялись по уравнению (37), и соотношения между старыми сопротивлениями R_1, \dots, R_5 и новыми могут быть получены из выражения для рассеиваемой мощности:

$$P_d = [R_1 + \eta^2 R_2 + (r + \xi)^2 R_3] \omega^2 + [R_2 + R_4] \dot{\xi}^2 + [R_3 + R_5] \dot{\eta}^2 + 2[-(\eta R_2)] \dot{\xi} \omega + 2[(r + \xi) R_3] \dot{\eta} \omega = \\ = R_\omega \omega^2 + R_\xi \dot{\xi}^2 + R_\eta \dot{\eta}^2 + 2R_{\xi\omega} \dot{\xi} \omega + 2R_{\eta\omega} \dot{\eta} \omega.$$

Податливости остаются теми же самыми, что и прежде.

Практический гироскоп, вероятно, будет спроектирован так, что k_i намного больше k_r , поэтому, как отмечалось ранее, $\eta \simeq 0$ по сравнению с ξ . Большинство запаасающих энергию элементов можно отбросить, и если демпфированием можно также пренебречь, то в результате получим граф связей, изображенный на рис.7, *b*. Он фактически идентичен показанному на рис.6, *b*, хотя I_ω содержит энергии, принадлежащие I_1 и I_2 .

Для вывода уравнения состояния заметим, что

$$\mathbf{E} = \begin{bmatrix} I_\omega & I_{\omega\xi} & I_{\omega\eta} \\ I_{\omega\xi} & m & 0 \\ I_{\omega\eta} & 0 & m \end{bmatrix}; \\ \mathbf{F} = m \begin{bmatrix} 0 & 2(r + \xi)\omega & 2\eta\omega \\ -n - (r + \xi)\omega & 0 & -\omega \\ \xi - \eta\omega & \omega & 0 \end{bmatrix}; \\ \mathbf{H} = \begin{bmatrix} 0 & 0 \\ 2k_r & 0 \\ 0 & 2k_i \end{bmatrix}; \quad \mathbf{q} = \begin{bmatrix} \xi \\ \eta \end{bmatrix}; \quad \mathbf{A}u = \begin{bmatrix} \tau \\ 0 \\ 0 \end{bmatrix};$$

После преобразования уравнения состояния к простейшей форме для большинства применений, в которых ω редко приближается к нулю, но

ξ и η часто проходят через нуль, имеем

$$\frac{d}{dt} \begin{bmatrix} \omega \\ \xi \\ \eta \\ \xi \\ \eta \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -\frac{2k_r}{I} \eta & \frac{2k_i}{I} (r + \xi) \\ r\omega & 0 & 2\omega & \omega^2 - \frac{2k_r}{I} \left(\frac{I}{m} + \eta^2 \right) & \frac{2k_i}{I} (r + \xi) \eta \\ 0 & -2\omega & 0 & \frac{2k_r}{I} (r + \xi) \eta & \omega^2 - \frac{2k_i}{I} \left[\frac{I}{m} + (r + \xi)^2 \right] \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ \xi \\ \eta \\ \xi \\ \eta \end{bmatrix} + \begin{bmatrix} 1/I \\ \eta/I \\ r + \xi \\ 0 \\ 0 \end{bmatrix} \tau.$$

Кононюк А.Е. Графы

Девять элементов вышеприведенной квадратной матрицы непостоянны, причем многие из них зависят от времени сложным образом, поэтому фактически требуется двенадцать умножений переменных вместо пяти в случае основного лагранжева графа связей.

Дуальная лагранжева формулировка

Дуальная лагранжева формулировка использует \mathbf{p} и \mathbf{e} в качестве переменных состояния вместо \mathbf{q} и \mathbf{f} соответственно. Все уравнения и графы в данной работе могут быть дуализированы. Потенциальная коэнергия играет роль, ранее исполнявшуюся кинетической коэнергией, и т. д.

Если полностью не связывать физическое перемещение и «обобщенное перемещение» и сделать это с другими обобщенными переменными, то дуальная формулировка может свестись к простому переименованию переменных. Конечные уравнения, подлежащие решению, вполне возможно, окажутся идентичными уравнениям, полученным при помощи более привычной формулировки. Таким образом, ничто не выигрывается и не теряется.

Если же дуальная формулировка побуждает выразить запасенную энергию через другие физические переменные, то конечные уравнения фактически могут стать иными, возможно более простыми. Последующая редуализация, естественно, вновь возвращает переменные к обычной форме, поэтому дуальная формулировка не всегда приводит к желаемым упрощениям, но может иметь полезное психологическое значение или служить средством общения.

Выводы

Лагранжевы графы связей следуют из моделирования голономных физических систем при помощи выражений энергии и мощности. Хотя уравнения Лагранжа содержатся в основной и альтернативной формах, основная форма дает преобразованные уравнения в более естественных или первообразных переменных, что часто значительно упрощает формулировку и последующее моделирование.

Графы связей позволяют охватить энергетическую структуру и функциональное поведение системы. В результате удастся лучше понять значимость возможных приближений. Графы также непосредственно приводят к формулировке уравнений в пространстве состояний или к аналоговой модели. При наличии математического обеспечения операций с символами составление уравнений в пространстве состояний можно провести на ЭВМ.

1.10. Исследование линейных и нелинейных характеристик подвесок аппаратов на воздушной подушке с использованием моделей, основанных на графах связей

С использованием графов связи Уормали Д., Гаргом Д. и Ричардсом Н. разработана нелинейная аналитическая модель динамики подвесок на воздушной подушке с упругой юбкой в диапазоне низких частот. Модель учитывает нелинейные характеристики источника потока и сопла и модуляцию емкости подушки и гидравлического сопротивления уплотняющего зазора изменением геометрии и давления подушки. Приведены графы связей для нелинейной модели подушки и для линеаризованной модели в случае малых возмущений. Получены и решены на ЦВМ нелинейные и линеаризованные уравнения динамики вертикального движения для реального диапазона условий работы. Сравнение решений нелинейных и линеаризованных уравнений показывает, что линеаризованная модель пригодна для предварительного расчета и оптимизации подвесок аппаратов на воздушной подушке. Однако для проверки предварительного расчета при больших входных воздействиях желательно нелинейное моделирование.

Обозначения

$a_c, a.$ — безразмерный коэффициент зависимости давление — расход;
 A_e — эффективная площадь основания подвески;
 A_i — внутренняя площадь упругой юбки;-
 A_r — отношение площадей;
 A_s — площадь сечения подводящего сопла;
 b_b — демпфирование упругой юбки;
 $b_{c.a.}$ — безразмерный коэффициент зависимости перемещение — расход;
 c_{do} — коэффициент расхода через выходной зазор;
 c_{ds} — коэффициент расхода через подводящее сопло;
 C — емкость подушки;
 C_d — емкость трубопровода;
 C_v — постоянная произвольная емкости зазора, см. (9);
 F — внешняя сила, действующая на подвеску;
 F_e — уравновешивающая сила;
 F_i — сила в камере сжатия;
 F_x — внешняя сила, действующая на подрессоренную массу;

Кононюк А.Е. Графы

- F_2 — сила в подвеске;
 h — высота уплотняющего зазора;
 h_b — разностная высота юбки;
 h_e — равновесная высота зазора;
 I_F — инерционность аппарата при продольном движении;
 k_b — жесткость упругой юбки;
 l — длина окружности уплотняющего выступа подушки;
 m — неподдресоренная масса;
 M — поддресоренная масса;
 \bar{m}_r — отношение масс;
 p_a — атмосферное давление;
 p_c — абсолютное давление в подушке;
 p_{ce} — абсолютное равновесное давление в камере сжатия;
 p_d — абсолютное давление в трубопроводе;
 p_f — абсолютное давление в воздухонагнетателе;
 p_{fe} — абсолютное равновесное давление в воздухонагнетателе;
 Q_e — равновесный расход;
 Q_0 — расход утечки через основание подвески;
 Q_s — расход из воздухонагнетателя;
 R — отношение жесткостей;
 R_f — сопротивление, определяемое по характеристикам воздухонагнетателя;
 R_0 — эквивалентное линейное сопротивление выходного зазора;
 R_s — эквивалентное последовательное сопротивление между трубопроводом и камерой сжатия;
 S_f — характеристики компрессора;
 S_{p0} — эталонный источник атмосферного давления;
 S_{ve} — эталонное значение равновесного объема подушки;
 t — время;
 V_c — суммарный объем юбки и области зазора;
 V_{ce} — равновесный объем юбки;
 V_d — объем трубопровода;
 y_0 — неровность пути (входное возмущение);
 y_1 — перемещение неподдресоренной массы;
 y_2 — перемещение поддресоренной массы;
 Δ — приращение;
 κ — показатель политропы;
 ξ — коэффициент демпфирования юбки;
 ρ — плотность жидкости;
 τ — безразмерное время;
 τ_1 — постоянная времени пребывания;

Кононюк А.Е. Графы

ω_n —собственная частота подрессоренной массы;

$(\dot{})$ — производная по времени;

$(\bar{})$ — безразмерный параметр.

Введение

Подвески, в которых воздушная подушка низкого давления (порядка 0,1 кгс/см²) служит источником опорных и управляющих сил для гусеничных наземных транспортных средств, являются предметом многих исследований и разработок. Модели и прототипы таких средств для скоростных междугородных и низкоскоростных внутригородских перевозок разработаны в США и других странах.

При больших скоростях (150—500 км/ч) воздушные подушки обладают высокой способностью компенсировать неровности пути, что увеличивает удобства пассажиров, безопасность движения и уменьшает эксплуатационные расходы по сравнению с обычными колесными подвесками. Для низкоскоростного (менее 150 км/ч) городского транспорта привлекательными являются простота конструкции и способность к боковому маневру.

Предложено множество подвесок на воздушной подушке, включая подвески камерного и соплового типов с использованием упругой юбки и без нее, аэродинамические подвески или крыло экраноплана, несущая способность которых создается за счет поступательного движения аппарата. В данном разделе рассмотрена подвеска камерного типа, схематически изображенная на рис. 1, типичная для многих практических конфигураций, предложенных для гусеничных аппаратов.

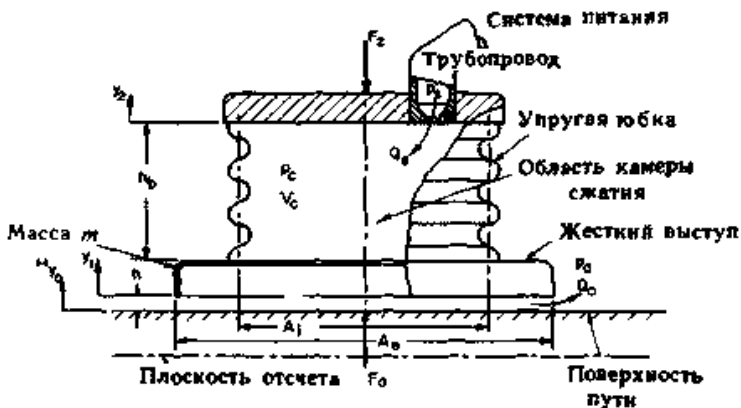


Рис. 1. Подвеска камерного типа с упругой юбкой.

Воздух нагнетается через систему сужений и уплотнений таким образом, что под подвеской создается подушка воздуха, удерживающая нагрузку. Обычно используют от 4 до 6 подушек протяженностью порядка метра и более для поддержания веса аппарата, а для управления поступательным движением по бортам аппарата устанавливают дополнительные подушки. Большая площадь, низкое давление в подушке позволяют распределить вес аппарата по поверхности пути и компенсировать локальные неровности, так что можно использовать такие локально шероховатые поверхности, как асфальт.

Механизм работы подвески камерного типа иллюстрирует рис. 1. Поток воздуха с расходом Q , подается в камеру повышенного давления (камеру сжатия) подушки от источника (воздухонагнетателя), в качестве которого обычно используют компрессор или вентилятор. В установившемся состоянии этот поток проходит через уплотняющий зазор h между поверхностью пути или земли и внешним ободом выступа подвески. Этот зазор, характеристики воздухонагнетателя и другие сужения и утечки определяют p_{ce} , которое, действуя на площадь основания подвески A_e , создает уравновешенную силу F_e . При медленном уменьшении зазора h выходное сопротивление возрастает и возрастает давление p_c , вызывая рост силы F_2 . Таким образом, для установившегося состояния существует определенное соотношение между нагрузкой и зазором h , подобное уравнению пружины (нелинейной). В ряде работ показано, что для реальных воздушных зазоров (от 0,1 до 1,5 см) подвеска будет более жесткой, чем требуется для комфортабельного передвижения (здесь существенно то

Кононюк А.Е. Графы

обстоятельство, что h должно быть достаточно малым, чтобы расход воздуха и мощность аппарата находились в разумных пределах.)

Для уменьшения суммарной жесткости подвески используются упругая юбка или другие деформируемые элементы. При этом упругая часть подвески работает таким образом, что увеличение давления подушки сжимает упругий элемент (h_b на рис. 1). Суммарное перемещение подвески из-за изменения нагрузки ΔF_2 будет равно сумме перемещения юбки и изменения зазора.

Динамическое описание поведения подвески должно включать емкость («сжимаемость») камеры сжатия, массу, упругость, демпфирование упругой юбки, динамические характеристики воздухонагревателя и эффекты модуляции параметров из-за перемещения юбки и изменения воздушного зазора. Ранее были рассмотрены модели с сосредоточенными параметрами динамики подвесок подобного типа. Исследования, опирающиеся на линеаризованные (линейные) модели, послужили основой для проектирования и оптимизации таких подвесок. Однако данная подвеска существенно нелинейна, а приближенная оценка справедливости линейного анализа дана лишь в ряде кратких исследований.

Применение графического представления модели подушки может оказать существенную помощь в понимании нелинейного гидромеханического взаимодействия, характеризующего подушку, а также при совместном рассмотрении моделей подушки с моделями системы питания и динамики аппарата для определения динамических взаимодействий, которые важны для всего гусеничного аппарата на воздушной подушке. Для нелинейной системы с сильным гидромеханическим взаимодействием (таким, как в воздушной подушке) использование графов связей дает компактный мощный метод представления в простой графической форме существующих сложных взаимодействий и четкого определения каналов передачи энергии, модуляции, рассеяния и накопления энергии. Этот метод дает также основы системного синтеза моделей подушки, системы питания и динамики аппарата в полную единую модель системы. Таким образом, данная работа преследует две основные цели: во-первых, разработать граф связей модели подвески на воздушной подушке камерного типа как в полностью нелинейном случае, так и линейном; во-вторых, оценить для реального диапазона изменения параметров системы и входных воздействий обоснованность применения линейных моделей. Последнее осуществлено путем решения уравнений для нелинейной и линейной модели подушки и сравнения полученных результатов. Основные усилия направлены на разработку модели одной лишь подушки и кратко рассмотрен подход, с помощью

Кононюк А.Е. Графы

которого модели подушки могут быть объединены с моделями системы питания и динамики аппарата.

Модель подвески

Физические процессы и основные допущения.

Подвеска, схематически изображенная на рис. 1, по существу есть система с тремя энергетическими входами (полусвязями), которую в общих обозначениях графов связей можно представить в виде



где Π — подвеска, а связи изображают поверхности раздела для пути (F_0, \dot{y}_0) и для аппарата (F_2, \dot{y}_2); третья связь изображает поверхность раздела (p_s, Q_s) между подушкой и системой питания.

В динамической модели подвески приняты следующие допущения:

1. Во всем объеме подушки давление p_c распределено равномерно. Это допущение означает, что внутренняя часть камеры сжатия должна быть спроектирована таким образом, чтобы минимизировать перепады давления, и оно налагает ограничение на частоты в модели, ниже которых эффект распространения волн в камере становится существенным.
2. Давление в камере сжатия и плотность связаны уравнением политропы

$$\frac{p_c}{\rho_c} = \kappa \frac{p_a}{\rho_a} \quad (1)$$

3. Неровности пути достаточно «низкочастотны», так что зазор h принят постоянным по длине подушки.
4. Рассматриваем лишь одномерное вертикальное движение элементов подушки.
5. Принимаем, что упругая юбка обладает жесткостью k_b , демпфированием b_b и массой m , сосредоточенной в жестком выступе или в основании.
6. Давление в подушке и его динамические изменения малы по сравнению с атмосферным (внешним) давлением. Обычно равновесное избыточное давление ($p_{ce} - p_a$) имеет порядок $0,1 \text{ кгс/см}^2$, а

Кононюк А.Е. Графы

изменения меньше равновесного значения (раметим (рис. 7 и 8), что динамические изменения давления обычно меньше равновесных значений и малы по сравнению с абсолютным внешним давлением.. Поэтому для описания сужений потока в подушке и системе питания можно использовать квадратные уравнения для расхода несжимаемой жидкости.

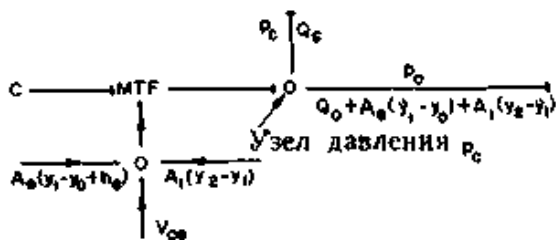
Графы подсистем и основные соотношения.

Для камеры сжатия применимо уравнение неразрывности (суммарный массовый расход в подушку должен равняться скорости аккумуляирования массы в подушке). Используя объемный расход при плотности в камере сжатия и уравнение (1), получим основные уравнения в виде

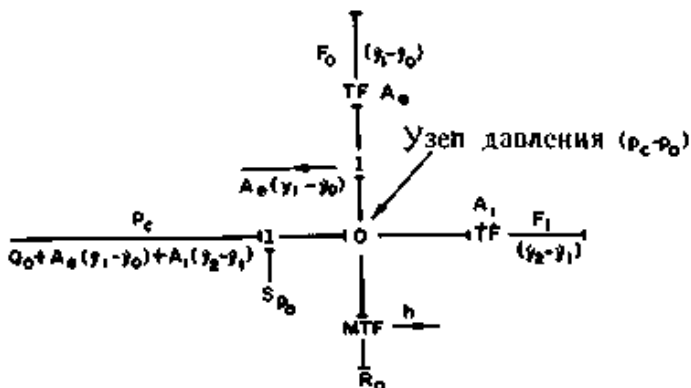
$$Q_c = C\dot{p}_c + Q_0 + A_c(\dot{y}_1 - \dot{y}_0) + A_c(\dot{y}_2 - \dot{y}_1) \quad (2)$$

$$V_c = V_{c0} + A_c h_c + A_c(y_1 - y_0) + A_c(y_2 - y_1) \quad (3)$$

где $C=V_c/\kappa p_c$, а соответствующий граф связей представлен на рис. 2, а.



a. Соотношения для камеры сжатия



b. Гидромеханическое преобразование и утечки

Рис. 2. Графы связей для камеры сжатия.

Члены $A_e(\dot{y}_1 - \dot{y}_0)$ и $A_1(\dot{y}_2 - \dot{y}_0)$ дают объемные расходы в системе за счет механического перекачивания из-за изменения зазора и движения упругой юбки. Заметим, что емкость C существенно нелинейна, во-первых, из-за зависимости от величины абсолютного давления и, во-вторых, из-за геометрической модуляции изменением объема. Соответствующая причинная связь изображена чертой, нормальной к каждой связи, а активные связи изображены направленными связями.

Суммарный расход в узле при давлении p_c , показанный на рис. 2, *a*, можно представить, как показано на рис. 2, *b*. Расход утечки через уплотняющий зазор подушки является нелинейной функцией давления и зависит от зазора h

Кононюк А.Е. Графы

$$Q_0 = kic_{01} \sqrt{\frac{2(p_c - p_a)}{\rho_{ca}}} \quad (4)$$

Гидромеханическое преобразование, показанное на рис. 2, *b* в виде трансформаторов с модулями A_e и A_b , можно представить в виде

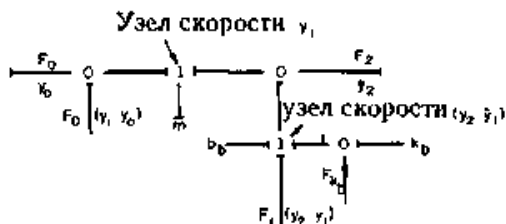
$$F_0 = A_e(p_c - p_a) : Q_{A_e} = A_e(\dot{y}_1 - \dot{y}_a) \quad (5)$$

$$F_1 = A_b(p_c - p_a) : Q_{A_b} = A_b(\dot{y}_2 - \dot{y}_1) \quad (6)$$

Подушку или упругую юбку можно представить, как на рис. 3, *a*, где силы равны

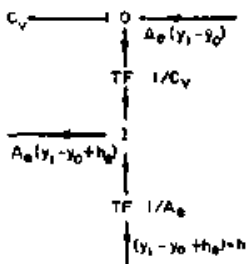
$$F_0 - F_2 = m\ddot{y}_1 \quad (7)$$

$$F_2 = F_1 + k_b(y_1 - y_2) + b_b(\dot{y}_1 - \dot{y}_2) \quad (8)$$



a. Упругая юбка

**Активная
связь**



б Активные связи для определения h

Рис. 3. Графы связей для упругой юбки.

Сила $F_0 - F_2$ действует на массу юбки, вызывая ускорение \ddot{y}_1 .

Сила F_2 складывается из силы F_{1i} , создаваемой давлением в камере сжатия, и суммарной силы в упругой демпфирующей юбке.

Кононюк А.Е. Графы

Необходимы две геометрические величины как модуляторы емкости камеры C и выходного сопротивления R_0 : суммарный объем V_e [уравнение (3)] и зазор h [уравнение (4)]. Член $A_i(y_2 - y_1)$ можно получить умножением упругой силы юбки F_{kb} на модуль трансформатора A_i/k_b . Для определения h и члена $A_e(y_1 - y_0)$ в уравнении (3) необходимо проинтегрировать $A_e(y_1 - y_0)$. Это можно сделать, используя активные связи, как показано на рис. 3, b , где

$$p_a = C_e A_e (y_1 - y_0) \quad (9)$$

И наконец, опишем систему питания. Обычно источник питания состоит из одного или нескольких компрессоров и систем трубопроводов, питающих несколько подушек. Так как целью данной работы является определение характеристик подвески, принята упрощенная система питания, приемлемая для многих практических систем. Основными рассматриваемыми элементами являются нелинейная характеристика компрессора, полная емкость трубопровода, последовательное сопротивление R_s между трубопроводом и камерой сжатия. На рис. 4 приведен граф связей и характеристика компрессора, использованная при исследовании.

Кононюк А.Е. Графы



Рис. 4. Графы связей для системы питания.

Компрессор, сопротивление и емкостные характеристики можно описать в виде

$$S_f: Q_f = f(p_f - p_0) = f(p_d - p_0) \quad (10)$$

$$Q_f - Q_c = C_d \dot{p}_d \quad (11)$$

$$Q_c = c_{d0} A_s \sqrt{\frac{2(p_d - p_c)}{\rho_{c0}}} \quad (12)$$

где $C_d = V_d / k p_d$.

Кононюк А.Е. Графы

Полная модель подвески. Нелинейный вариант. Подсистемы объединены на рис. 5 для того, чтобы получить полный граф связей для нелинейной модели воздушной подушки.

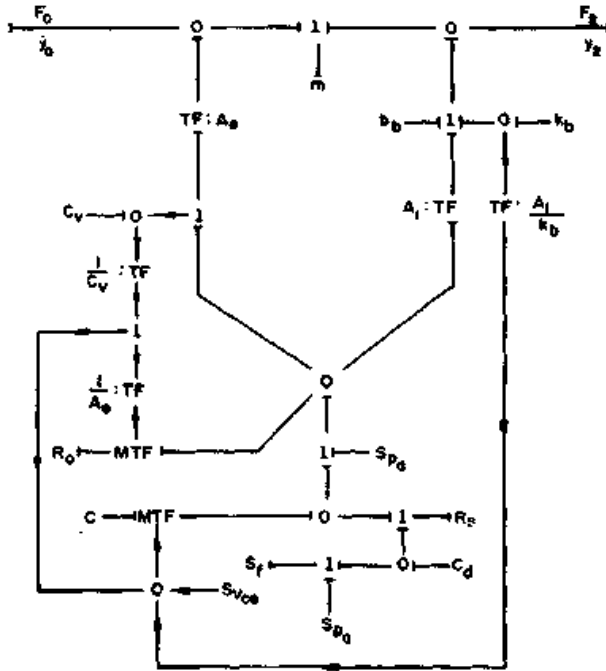


Рис. 5. Граф связей для нелинейной модели подвески.

Отметим, что порядок системы равен 5, числу элементов — аккумуляторов энергии в графе связей (k_b , C_v , C , C_d , m)

Линейный вариант. Приведенные уравнения можно линеаризовать при малых изменениях всех переменных и получить систему уравнений и граф связей для линейной модели воздушной подушки. Уравнение (2) линеаризовано выбором $C = (V_{ce} + A_e h_e) / k p_{ce}$ и заменой Q_s , Q_0 и \dot{p}_c приращениями ΔQ_s , ΔQ_0 и Δp_c . Уравнение (4) в линеаризованной форме имеет вид

$$\frac{\Delta Q_0}{Q_e} = b_{c,a} \frac{\Delta(y_1 - y_0)}{h_e} + a_{c,a} \frac{\Delta p_c}{p_{ce}}, \quad (13)$$

где $b_{c,a}$ и $a_{c,a}$ можно вычислить из (4) и (13). Первый член в правой части уравнения (13) определяет зависимый источник $\Delta Q_i = (b_{c,a} Q_e / h_e) \Delta(y_1 - y_0)$, который является результатом модуляции

Конюк А.Е. Графы

сопротивления воздушного зазора. Второй член определяет линейное постоянное сопротивление $R_0 = (p_{ce} / Q_c a_{ce})$, которое является сопротивлением воздушного зазора при $h = h_c$. Аналогично уравнения системы питания (1G) — (12) приводятся к виду

$$Q_f = Q_0 + \Delta Q_f = Q_0 + \frac{1}{R_f} \Delta p_d \quad (16)$$

$$\frac{1}{R_f} = \left. \frac{\partial Q_f}{\partial p_d} \right|_{p_{de}} \quad (15)$$

$$\Delta Q_f - \Delta Q_0 = \frac{V_d}{k p_{de}} \Delta p_d. \quad (16)$$

$$\frac{\Delta Q_0}{Q_0} = a_{k,c} \frac{\Delta p_d - \Delta p_c}{p_{de} - p_{ce}}. \quad (17)$$

Граф связей для линейной модели приведен на рис. 6.

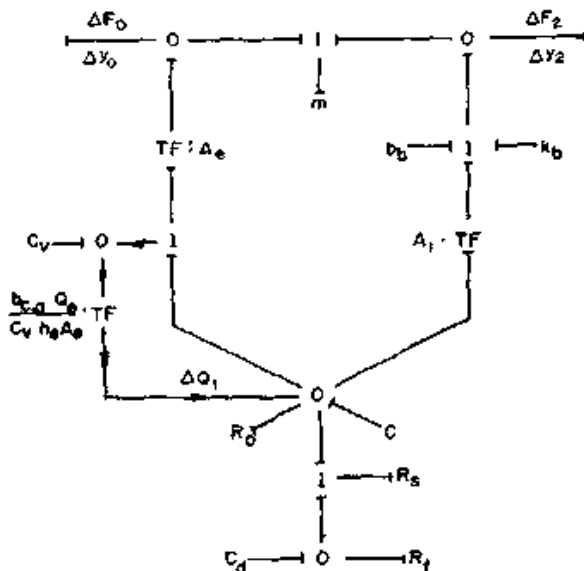


Рис. 6. Граф связей для линейной модели подвески

Заметим, что модуляция емкости камеры полностью исчезает, а нелинейное модулированное сопротивление воздушного зазора заменено постоянным сопротивлением и зависимым источником ΔQ_0 .

Кононюк А.Е. Графы

Однако для интегрирования ($\dot{y}_1 - \dot{y}_0$) по-прежнему нужна емкость C_v . Так как использованы только приращения, внешнее давление больше не появляется и различие между абсолютным и избыточным давлениями исчезает. Таким образом, граф сводится к сравнительно простой форме с пятью элементами — аккумуляторами энергии.

Параметры динамических характеристик. Эффекты нелинейностей подвески на воздушной подушке можно исследовать, рассматривая систему компрессор — трубопровод — подушка, для аппарата с массой M , к которой приложена внешняя сила F_x .

$$\begin{array}{c}
 \frac{F_0}{\dot{y}_0} \quad \Pi \quad \frac{F_2}{\dot{y}_2} \quad \Big| \quad \frac{\Sigma F_x}{1} \quad \text{---} \quad M \quad F_2 = M\ddot{y}_2 + F_x \quad (18) \\
 \rho_s \quad | \quad Q_s \\
 \text{Трубопровод—компрессор}
 \end{array}$$

Для создания основы сравнительного исследования нелинейной и линейной моделей подушки удобно ввести систему безразмерных переменных и параметров, характеризующих конфигурацию подушки и равновесную рабочую точку и полезных с точки зрения проектирования; эта система приведена в табл. 1 и 2.

Таблица 1

Нормирование переменных модели воздушной подушки

Переменная	Безразмерный символ	Определение
Время t	τ	$\omega_n t$
Сила F	\bar{F}	F/F_e
Перемещение y	\bar{y}	y/h_e
Скорость \dot{y}	$\dot{\bar{y}}$	$\dot{y}/h_e \omega_n$
Ускорение \ddot{y}	$\ddot{\bar{y}}$	$\ddot{y}/h_e \omega_n^2$
Давление p	\bar{p}	$p - p_a / p_e - p_a$
Расход Q	\bar{Q}	Q/Q_e
Объем V	\bar{V}	$V/A_e h_e$

Здесь $\omega_n = \sqrt{\frac{2(1-R)F_e}{Mh_e(1+F_e/6F)}}$

Основные параметры комбинации подушки—система питания

Параметр	Описание
$\delta F/F_e = (p_{je} - p_{ce}) / (p_{ce} - p_a)$	Отношение приращения силы к уравновешивающей силе
p_{ce}/p_0	Относительное равновесное давление в подушке
$\bar{m}_r = m/M$	Отношение массы юбки к подressоренной массе
$\xi = b_0 \omega_n / 2Rk_b$	Коэффициент демпфирования юбки
$R = A_r M \omega_n^2 / k_b$	Мера жесткости элемента зазор — юбка ($R = 0$ — жесткая юбка); ($R = 1$ — бесконечно эластичная юбка)
$A_T = A_e - A_i / A_e$	Относительная площадь
$\bar{V}_{ce} = V_{ce} / A_e h_e$	Относительный объем камеры сжатия
$\bar{V}_d = V_d / A_e h_e$	Относительный объем трубопровода
κ	Показатель политропы (константа)
$\bar{\tau}_1 = A_e h_e \omega_n / Q_e$	Постоянная времени, связанная с временем пребывания жидкости в активном объеме

Безразмерные динамические нелинейные и линейные уравнения подвески на воздушной подушке можно получить из графа связей модели (рис. 5) и уравнений (1) — (2); они приведены в приложении для подушки, поддерживающей аппарат с массой M , к которой приложена внешняя сила F_x .

Динамические характеристики подвески. Динамические характеристики моделей подвески можно определить, решая систему уравнений, приведенную в приложении. Исследованы реакции на входные воздействия двух типов: реакция подушки в номинальной равновесной точке на мгновенное ступенчатое изменение внешней силы, равное 25% уравновешивающей силы, т. е. увеличение $\Delta \bar{F}_x$ до 0,25, причем параметры состояния подушки сохраняют равновесные значения в моментах $t=0$; реакция подушки в номинальной равновесной точке на мгновенное ступенчатое изменение рабочего зазора до нуля, причем другие параметры подушки сохраняют

Кононюк А.Е. Графы

равновесные значения в момент $t=0$. Входные воздействия являются верхними границами величин внешней ветровой нагрузки и неровностей пути, ожидаемых при работе подвески. Такие воздействия выбраны для того, чтобы подчеркнуть различие между нелинейной и линейной характеристиками подушки.

Реакции моделей подушки на данные воздействия определены для пяти конфигураций подвески, приведенных в табл. 3.

Таблица 3

Параметры конфигураций подвески						
Конфигурация	f_n , Гц	h_n , см	\bar{v}_1	R	\bar{v}_{0z+1}	\bar{v}_d
1	1,0	0,635	0,0467	0,963	13,0	25,4
2	0,5	1,27	0,0233	0,982	7,0	12,7
3	2,0	1,27	0,0931	0,706	7,0	12,7
4	0,5	0,318	0,0233	0,996	13,0	51,1
5	2,0	0,318	0,0935	0,926	13,0	51,1

Приведенные конфигурации соответствуют гусеничным аппаратам на воздушной подушке, представляющим интерес для городских и междугородних перевозок. Изменения параметров конфигураций можно выразить через изменения номинального рабочего зазора и собственной частоты системы аппарат—подвеска. Рабочий зазор, который при фиксированных площади подушки и весе аппарата прямо пропорционален требуемой для создания подушки мощности, изменяется от 0,318 до 1,27 см, а собственная частота, тесно связанная с комфортом передвижения,— от 0,5 до 2,0 Гц.

Реакции подвесок определены численным интегрированием методом Рунге — Кутты четвертого порядка на ЦВМ и моделированием на АВМ. Основная часть представленных данных получена на АВМ; для отдельных линейных и нелинейных случаев результаты проверены на ЭЦВМ.

Реакции моделей линейной и нелинейной подушек для конфигурации 1 приведены на фиг. 7 и 8.

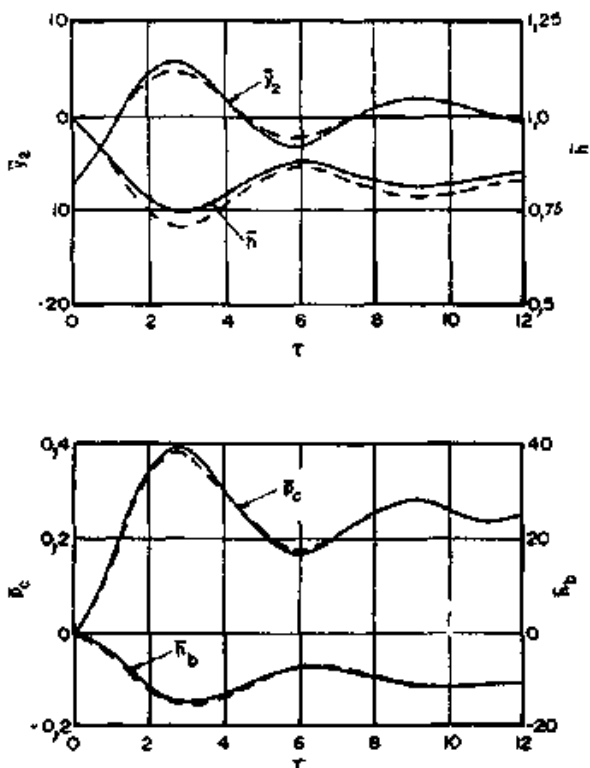


Рис. 7. Реакция конфигурации 1 на ступенчатое изменение силы.
 $f_n=1$ Гц, $h_e=0,635$ см.

————— нелинейная модель;-----линейная модель.

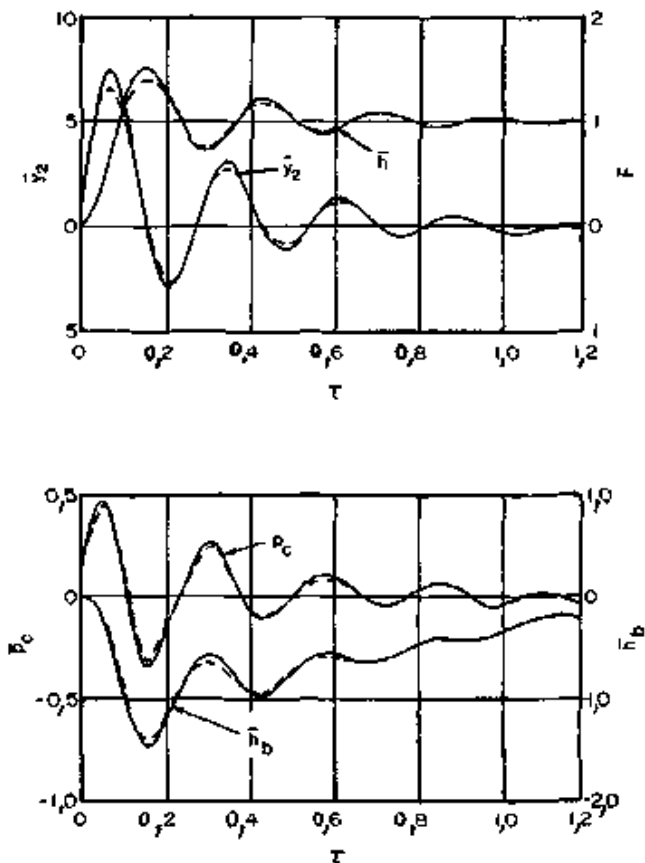


Рис. 8. Реакция конфигурации 1 на ступенчатое изменение неровности пути $f_n=1$ Гц, $h_e=0,635$ см.

— нелинейная модель;-----линейная модель.

Данные, приведенные для силы, иллюстрируют основные свойства вспомогательной подвески с подрессоренной массой упругой юбки, в то время как данные для пути отражают характеристики главной подвески с эффективным зазором, причем время переходного процесса в последнем случае в двадать раз меньше.

Кривые на рис. 7 для $\Delta \bar{F}_x=0,25$ показывают, что ускорение максимально в то время, когда другие параметры имеют равновесные значения, а при $t \rightarrow \infty$ ускорение стремится к нулю, в то время как \bar{h} ,

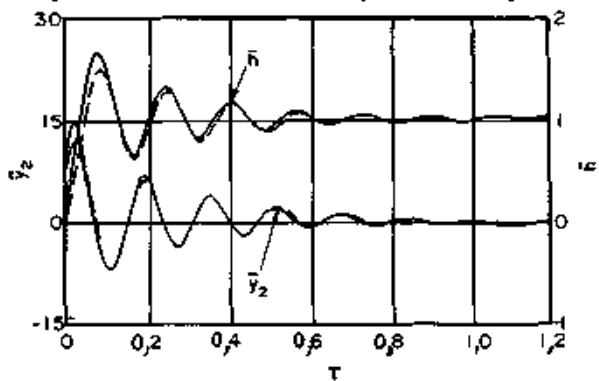
Кононюк А.Е. Графы

\bar{p}_c , \bar{h}_b стремятся к новым равновесным значениям. В переходных характеристиках всех этих параметров преобладает собственная частота подвески, равная 1 Гц.

На рис. 8 приведена реакция параметров состояния подушки на входное возмущение пути, соответствующее уменьшению h до нуля, характеризующая частоты, связанные с активной неподрессоренной массой юбки. Эти частоты почти в двадцать раз выше частот на рис. 7. Зависимости, приведенные на рис. 7 и 8, показывают, что даже при таких тяжелых входных воздействиях, как изменение равновесной нагрузки на 25% и скачок неровности пути на величину, равную рабочему зазору, линейная и нелинейная модели подвески дают очень близкие результаты.

Несмотря на наличие некоторой разницы в реакциях, различие настолько мало, что приближенные значения характерных частот, демпфирования и максимальных амплитуд, полученные при линейном анализе, дают хорошую исходную основу для предварительного проектирования воздушных подушек. На рис. 9—12 приведены реакции линейной и нелинейной моделей для безразмерных ускорений аппарата и рабочего зазора при равновесных зазорах от 0,318 до 1,27 см и собственных частотах от 0,5 до 2,0 Гц.

Ступенчатое изменение неровности пути



Ступенчатое изменение внешней силы

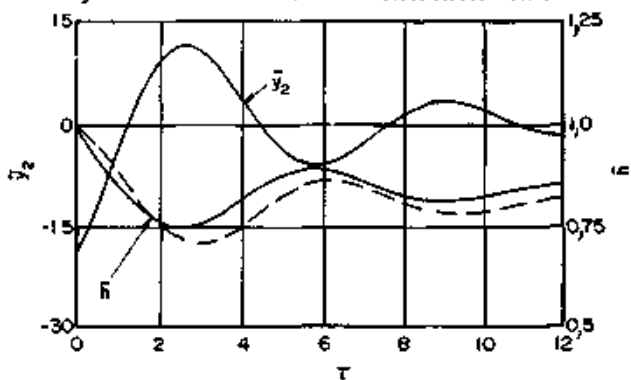


Рис. 9. Переходные характеристики конфигурации 2.
 $f_n=0,5$ Гц, $h_e=1,27$ см.

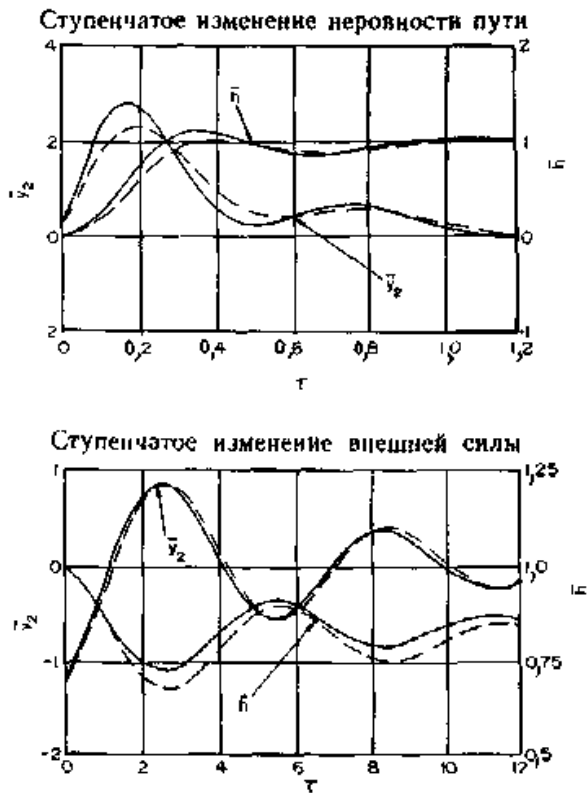


Рис. 10. Переходные характеристики конфигурации 3.
 $f_n=2$ Гц, $h_e=1,27$ см

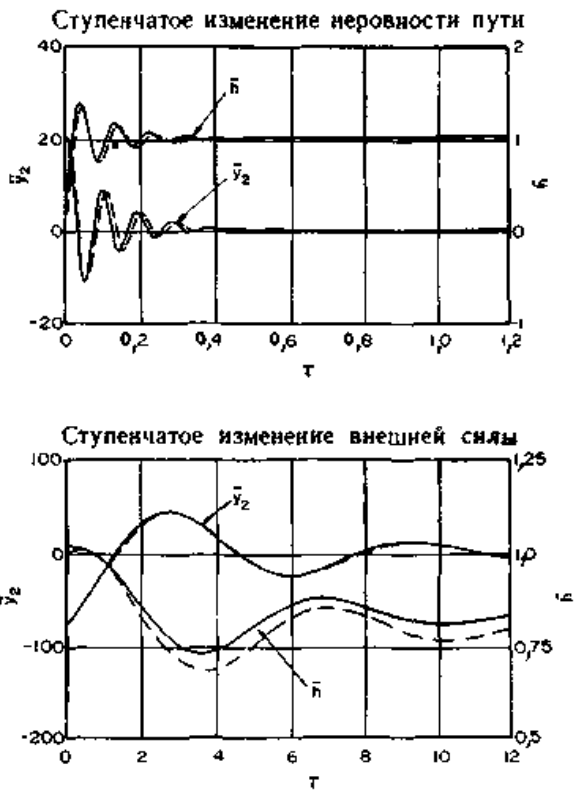
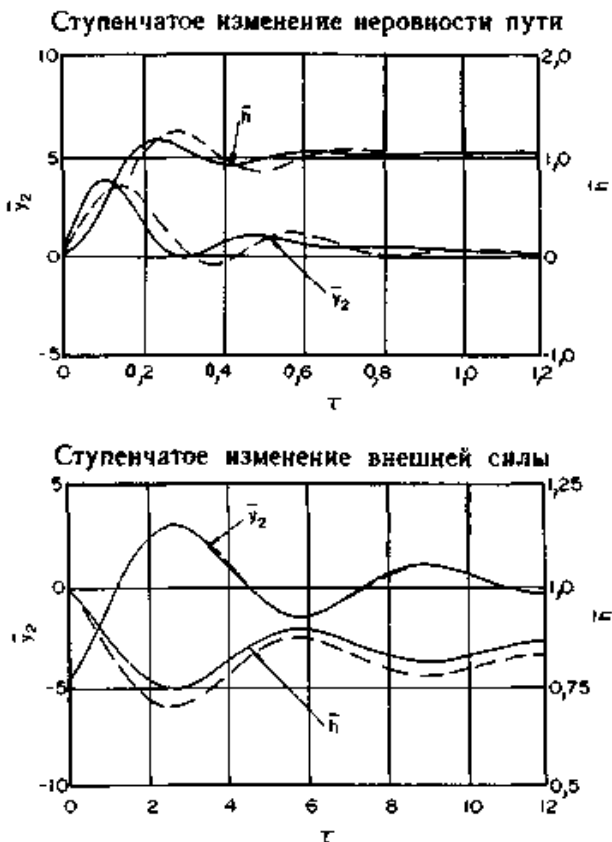


Рис. 11. Переходные характеристики конфигурации 4.
 $f_n=0,5$ Гц, $h_e=0,318$ см



Фиг. 12. Переходные характеристики конфигурации 5.
 $f_n=2$ Гц, $h_e=0,318$ см.

Во всех случаях разница между реакциями линейной и нелинейной моделей мала. Наибольшее различие имеет место для конфигурации 3 ($h_e=1,27$ см; $f_n=2,0$ Гц), но даже здесь максимальная амплитуда ускорения по линейному анализу отличается не более чем на 20% от предсказанной амплитуды по нелинейному анализу. Амплитуда колебания зазора по линейному анализу отличается не более чем на 15% от амплитуды по нелинейному анализу.

Моделирование показывает, что в исследованном диапазоне изменения параметров линейные модели должны давать хорошую исходную основу для проектирования подвески с воздушной подушкой. Использование линейного анализа на первом этапе проектирования,

Кононюк А.Е. Графы

когда проверяются многие варианты, может служить первым шагом для понимания влияния параметров на характеристики подвески и дает метод общего проектирования и оптимизации подвески. Физически хорошее соответствие результатов по линейной и нелинейной моделям даже при наличии характерных эффектов нелинейности (питающее сопло, сужение выходного зазора, емкость камеры сжатия) обусловлено тем, что подвеска должна быть спроектирована таким образом, чтобы обеспечить удобство передвижения — условие, которое требует, чтобы подвеска сводила к минимуму ускорение аппарата. Упругая юбка, которая принята линейной, обеспечивает упругость системы, необходимую для достижения собственных частот, близких к 1 Гц, при диапазоне рабочего зазора подушки от 0,318 до 1,27 см. Для удобства пассажиров необходимо при нормальных условиях работы избегать больших изменений рабочих давлений, зазора и деформации юбки, т. е. в хорошем проекте необходимо ограничить нелинейные эффекты в характеристиках подвески.

Многомерные модели аппарата. Хотя в данной работе подробно исследована модель одной подушки, основной целью построения графа связей модели подушки является использование этой модели для разработки системного метода моделирования аппаратов с несколькими воздушными подушками. Иллюстрация взаимосвязи моделей подушки с моделью трубопровода и моделью корпуса аппарата приведена на рис. 13 для аппарата, пригодного к вертикальному и продольному движению.

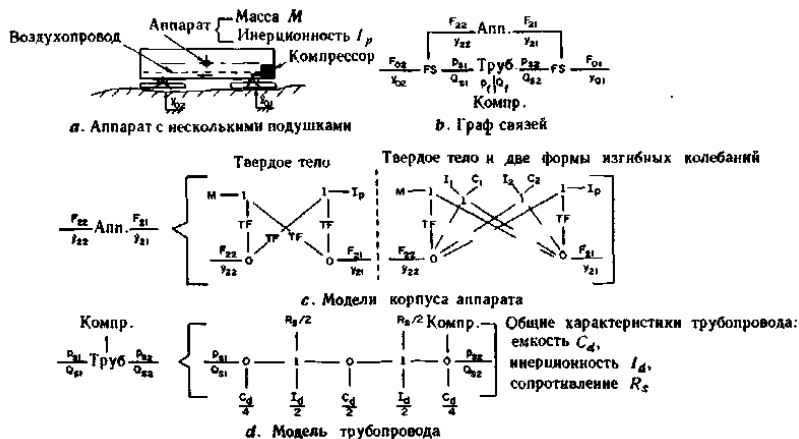


Рис. 13. Модели аппарата с несколькими подушками.

Кононюк А.Е. Графы

Связь между движением аппарата и системой питания подвески очевидна из графа связей. Простейшая модель аппарата, учитывающая как поступательное движение массы, так и момент инерции относительно поперечной оси, приведена на рис. 13, *с*. Если модель учитывает изгибные колебания аппарата, граф связей можно усложнить (рис. 13, *с*). Трубопровод также можно представить моделями различной сложности — от простой модели, в которой учтены лишь эффективная емкость и сопротивление (рис. 4), до модели, в которой учтены две первые акустические моды колебаний (рис. 13, *д*). Таким образом, заложены основы систематической методики построения модели.

Заключение

Разработаны линейная и нелинейная модели графов связей для подвески на воздушной подушке с упругой юбкой. Переходные характеристики показывают, что для исследованного диапазона конфигураций подушки реакции линейной и нелинейной моделей весьма близки. Эти результаты позволяют сделать вывод, что линейная модель пригодна для предварительного расчета и оптимизации подвески на воздушной подушке. Однако для проверки эффектов больших входных воздействий желательное нелинейное моделирование. Наконец, кратко рассмотрен синтез полной динамической модели гусеничного аппарата на воздушной подушке, использующей отдельные модели подушки, системы трубопровод - компрессор и динамики аппарата.

ПРИЛОЖЕНИЕ

Краткий обзор нелинейных и линейных уравнений

Безразмерные уравнения, описывающие подвеску и систему питания, можно выразить через шесть параметров состояния и (для удобства) через три дополнительных параметра потока. Для простоты выразим параметры через отклонения от равновесной рабочей точки.

Уравнения, описывающие нелинейную модель системы, приведены ниже.

Скорость подрессоренной массы

Конюк А.Е. Графы

$$\frac{d\dot{\Delta y}_2}{d\tau} = -2\xi A_r |\dot{\Delta y}_2 - \dot{\Delta y}_1| - \frac{A_r \dot{h}_b}{R} + \frac{(1 - A_r)[1 + F_s/\delta F]}{2(1 - R)} \Delta \bar{p}_c - \frac{1 + F_s/\delta F}{2(1 - R)} \Delta \bar{F}_s. \quad (18)$$

Скорость массы юбки

$$\frac{d\dot{\Delta y}_1}{d\tau} = \frac{1}{m_r} \left[-2\xi A_r |\dot{\Delta y}_1 - \dot{\Delta y}_2| + \frac{A_r \dot{h}_b}{R} + \frac{A_r [1 + F_s/\delta F]}{2|1 - R|} \Delta \bar{p}_c \right], \quad (19)$$

Перемещение юбки

$$\frac{d\dot{h}_b}{d\tau} = |\dot{\Delta y}_2 - \dot{\Delta y}_1|. \quad (20)$$

Объем подвески

$$\frac{d\Delta \bar{V}_c}{d\tau} = (1 - A_r) |\dot{\Delta y}_2 - \dot{\Delta y}_1| + |\dot{\Delta y}_1 - \dot{\Delta y}_2|. \quad (21)$$

Давление в камере сжатия

$$\frac{d\Delta \bar{p}_c}{d\tau} = -\kappa \left(\frac{|\dot{\Delta Q}_c - \dot{\Delta Q}_s|}{\bar{r}_1} - (1 - A_r) |\dot{\Delta y}_2 - \dot{\Delta y}_1| - |\dot{\Delta y}_1 - \dot{\Delta y}_2| \right) \times \left[\Delta \bar{p}_c + \frac{1}{1 - p_a/p_{c0}} \right]. \quad (22)$$

Давление в трубопроводе

$$\frac{d\Delta \bar{p}_d}{d\tau} = \frac{\kappa |\dot{\Delta Q}_f - \dot{\Delta Q}_s|}{V_s \bar{r}_1} \left(\Delta \bar{p}_d + \frac{1}{1 - p_a/p_{f0}} \right). \quad (23)$$

Расходы (22) и (23) можно записать в виде

$$\dot{\Delta Q}_0 = (1 + \Delta \bar{h}) \sqrt{1 + \Delta \bar{p}_c} - 1; \quad (24)$$

$$\Delta \bar{Q}_c = \sqrt{\frac{F_s}{\delta F} \left[\frac{1 - p_a/p_{f0}}{1 - p_a/p_{c0}} \left[1 + \frac{\delta F}{F_s} (1 - p_a/p_{c0}) \right] \Delta \bar{p}_d - \Delta \bar{p}_c \right]} + 1 - 1. \quad (25)$$

Кононюк А.Е. Графы

причем $\Delta Q_f = f(\Delta p_d)$ (характеристика компрессора на рис. 4), и для удобства приращение зазора Δh представлено в виде

$$\bar{\Delta h} = \bar{\Delta V}_c - (1 - A_c) \bar{h}_c. \quad (26)$$

Уравнения, описывающие линейную модель воздушной подушки, можно выразить через те же самые переменные, что и в нелинейном случае. Уравнения (18) — (21) уже линейны, а уравнения провода (22) — (25) должны быть линеаризованы; таким образом, линейная модель состоит из уравнений (18) — (21) и следующих уравнений, полученных линеаризацией (22) — (24):

$$\frac{d\bar{\Delta p}_d}{dt} = \frac{\kappa}{V_d \bar{\tau}_1} \frac{|\bar{\Delta Q}_f - \bar{\Delta Q}_c|}{|1 - p_c/p_{ce}|}, \quad (27)$$

$$\begin{aligned} \frac{d\bar{\Delta p}_c}{dt} &= \frac{\kappa}{|1 + \bar{V}_{ca}| \left[1 - \frac{p_c}{p_{ce}} \right]} \times \\ &\times \left\{ \frac{|\bar{\Delta Q}_c - \bar{\Delta Q}_0|}{\bar{\tau}_1} - |1 - A_c| |\bar{\Delta y}_2 - \bar{\Delta y}_1| - |\bar{\Delta y}_1 - \bar{\Delta y}_0| \right\} \end{aligned} \quad (28)$$

где расходы выражены в виде

$$\bar{\Delta Q}_c = \frac{-F_c}{2\delta F} \bar{\Delta p}_c + 1/2 [1 + F_c/\delta F] \bar{\Delta p}_d \quad (29)$$

$$\bar{\Delta Q}_0 = + \bar{\Delta h} + \frac{\Delta p_c}{2} \quad (30)$$

и где характеристика компрессора для линейной модели имеет вид

$$\bar{\Delta Q}_f = \left(\frac{1}{R_f} \right) \bar{\Delta p}_d \quad (31)$$

причем

$$\left(\frac{1}{R_f} \right) = \left. \frac{\partial \bar{\Delta Q}_f}{\partial \bar{\Delta p}_d} \right|_c$$

1.11. ПРИМЕНЕНИЕ ГРАФОВ СВЯЗЕЙ ПРИ РАЗРАБОТКЕ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ ПОРШНЕВЫХ ДВС

Поршневой ДВС (ПДВС) - сложный технический объект, функционирование которого определяет взаимосвязанная совокупность процессов различной физической природы: механических; тепловых; газодинамических; гидродинамических; физико-химических; информационных.

Большая часть из перечисленных процессов имеет сложное, трудно формализуемое описание, а потому попытка детального описания таких процессов тем более во взаимосвязи невозможна. Для этого требуются большие вычислительные мощности и мощное программное обеспечение. Все применяемые на практике модели ПДВС являются в той или иной мере «огрубленными» и, как правило, достаточно подробно описывают процессы в отдельных системах двигателя, не уделяя должного внимания учету их взаимодействия с другими системами и ПДВС «в целом».

На современном этапе развития теоретических исследований ПДВС очень важна унификация применяемых математических моделей для возможности их совместного применения в системах проектирования ПДВС CAE/CAD/CAE/CAM с использованием единой базы данных. При этом одной из важнейших задач является безошибочное и оперативное формулирование условий на границах взаимодействующих систем ПДВС. Эта задача может быть решена при помощи аппарата графов связей. Графы связей характеризуются наглядностью и относительной простотой представления информации о системе и основываются на анализе потоков энергии, передаваемых в технических системах. Важнейшим положением теории графов связей является тезис о ведущей роли в любом процессе (за исключением информационного) энергетического воздействия, величина которого определяется свойствами обеих взаимодействующих систем. При этом передаваемый поток энергии всегда определен произведением двух модельных величин, называемых фундаментальными (ток и напряжение для электрических, сила и скорость для механических, давление и объемный расход для гидравлических, удельная эксергия и массовый расход для газовых систем), одна из которых определена первой системой, а другая – второй. Процесс построения графов связей

Кононюк А.Е. Графы

для ПДВС основывается на рассмотрении обмена энергией различных подсистем двигателя с окружающей средой и между собой. Соединяя модели подсистем соответствующими связями, получают модель системы в целом. В дальнейшем модель может последовательно усложняться по мере учета важных с точки зрения исследователя, физических эффектов; одновременно определяются причинные зависимости каждого добавленного элемента и оценивается значимость дополнительно введенных эффектов. Для ПДВС методом графов связей можно построить различные графические модели в зависимости от поставленной задачи исследования. Поскольку главная функция ПДВС – преобразование химической энергии топлива в механическую работу в изменяющихся во времени условиях эксплуатации, то основная модель должна отражать именно эти, термодинамические по своей природе процессы. В ПДВС целесообразно выделить основное ядро, определяющее важнейшие свойства двигателя как энергодинамической установки, и вспомогательные звенья, уточняющие и детализирующие эти свойства. Для ПДВС характерно наличие 3 связей, по которым осуществляется обмен энергией: массо-, термо- и механоcontact. Массоcontact осуществляется за счет прихода и расхода рабочего тела, термоcontact – за счет подвода (отвода) к рабочему телу энергии в форме тепла, механоcontact – за счет отвода (подвода) от рабочего тела энергии в форме работы. Объединение потоков разнородной энергии можно осуществить, используя для их описания лишь единственную функцию состояния – эксергию, выделяющую в различных видах энергии однородную, полностью преобразуемую в другие виды составляющую. Используя термины графов связей, получим описание перечисленных выше энергетических взаимодействий, характерных для ПДВС.

На рис 1 приведен граф связей, построенный для одноцилиндрового дизельного двигателя, а в таблице – уравнения энергетических базовых звеньев, входящих в граф.

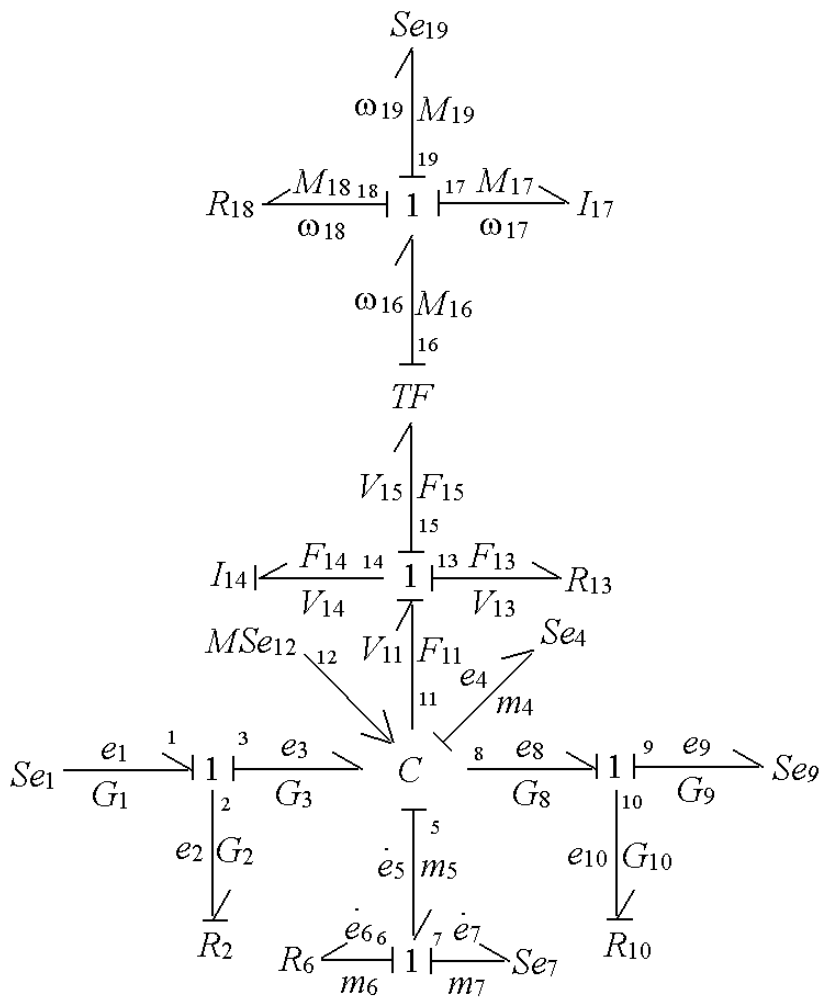


Рис. 1. Граф связей ПДВС

Уравнения энергетических базовых звеньев ПДВС

Таблица

Кононюк А.Е. Графы

№ п/п	Энергетические базовые звенья	Уравнения энергетических базовых звеньев
1	2	3
1	Источник Se_1 (окружающая среда)	$e_1 = i_1(p_1, T_1) - i_{oc} - T_{oc}[s_1(p_1, T_1) - s_{oc}]$
2	Приемник Se_4 (воспринимает энергию, накопленную рабочим телом)	$e_4 = u_4(p_4, T_4) - u_{oc} - T_{oc}[s_4(p_4, T_4) - s_{oc}] + p_{oc}(v_4 - v_{oc})$

1	2	3
3	Приемник Se_7 (окружающая среда)	$\dot{e}_7 = \frac{f_{cT} \cdot \alpha (T_4 - T_7)(T_7 - T_{oc})}{m_7 T_7}$
4	Приемник Se_9 (окружающая среда)	$e_9 = i_9(p_9, T_9) - i_{oc} - T_{oc}[s_9(p_9, T_9) - s_{oc}]$
5	Модулирующая (управляющая) информационная связь MSe_{12}	$\dot{e}_{12} = \dot{e}_{12}(m_x, \omega_T)$ $m_x = m_x(\omega_{17}), \omega_T = \omega_T(\omega_{17}, \varphi_{17}, \varphi_z, \bar{m})$
6	Приемник Se_{19} (потребитель)	$M_{19} = M_{19}(t)$
7	Приемник Se_7 (окружающая среда)	$\dot{e}_7 = \frac{f_{cT} \cdot \alpha (T_4 - T_7)(T_7 - T_{oc})}{m_7 T_7}$

Кононюк А.Е. Графы

8	1 – узел приходного массоконтакта	$G_1 = G_2, G_3 = G_2, e_2 = e_1 - e_3$
9	1 – узел расходного массоконтакта	$G_8 = G_{10}, G_9 = G_{10}, e_{10} = e_8 - e_9$
10	1 – узел термоконтакта	$m_6 = m_5, m_7 = m_5, \dot{e}_5 = \dot{e}_7 - \dot{e}_6$
11	1 – узел механоконтакта (поступательное движение)	$V_{11} = V_{14}, V_{13} = V_{14}, V_{15} = V_{14}, F_{14} = F_{11} - (F_{15} + F_{13})$
12	1 – узел механоконтакта (вращательное движение)	$\omega_{16} = \omega_{18}, \omega_{17} = \omega_{18}, \omega_{19} = \omega_{18}, M_{18} = M_{16} - (M_{17} + M_{19})$
13	C – поле (основное ядро)	$\frac{de_4}{dt} = \frac{1}{m_4} \left(e_3 \cdot G_3 - e_8 \cdot G_8 - \dot{e}_4 \cdot m_4 - \dot{e}_5 \cdot m_5 + \right. \\ \left. + \dot{e}_{12} \cdot m_{12} + F_{11} \cdot V_{11} \right)$ $\frac{dm_4}{dt} = G_3 - G_8$ $e_3 = i_3(p_3, T_3) - i_{oc} - T_{oc} [S_3(p_3, T_3) - S_{oc}]$ $e_8 = i_8(p_8, T_8) - i_{oc} - T_{oc} [S_8(p_8, T_8) - S_{oc}]$ $F_{11} = f_{\Pi}(p_4 - p_{oc}), m_5 = m_4, m_{12} = m_4$
14	TF – трансформирующий элемент (кривошипно-шатунный механизм)	$\omega_{16} = \frac{V_{15}}{r_k \left(\sin \varphi_{16} + \frac{\lambda}{2} \sin 2\varphi_{16} \right)}$

Кононюк А.Е. Графы

1	2	3
15	R – элемент приходящей массоcontactной связи	$G_2 = G_2(p_1, T_1, p_3, T_3), \quad e_2 = T_{oc}[s_3(p_3, T_3) - s_1(p_1, T_1)]$
16	R – элемент расходящей массоcontactной связи	$G_{10} = G_{10}(p_8, T_8, p_9, T_9),$ $e_{10} = T_{oc}[s_9(p_9, T_9) - s_8(p_8, T_8)]$
17	R – элемент термоcontactной связи	$\dot{e}_6 = \frac{f_{ct} \cdot \alpha \cdot T_{oc} (T_7 - T_5)^2}{m_6 T_7 \cdot T_5}$
18	R – элемент механоcontactной связи (поступательное движение)	$F_{13} = F_{13}(t)$
19	R – элемент механоcontactной связи (вращательное движение)	$M_{18} = M_{18}(t)$
20	I – элемент механоcontactной связи (поступательное движение)	$V_{14} = \frac{1}{m_{пр}} \int_0^t F_{14}(t) dt$
21	I – элемент механоcontactной связи (вращательное движение)	$\omega_{17} = \frac{1}{I_{пр}} \int_0^t M_{17}(t) dt$

В уравнениях таблицы использованы следующие условные обозначения: u, e, i, s – удельные внутренняя энергия, эксергия, энтальпия и энтропия рабочего тела соответственно; \dot{e} – плотность потока энергии (\dot{e}_{12} – поступающей в процессе сгорания топлива, \dot{e}_6 – теряемой при теплообмене); α – коэффициент теплообмена; G – расход; p, T, \mathcal{V} – давление, температура и удельный объем рабочего тела; m – масса рабочего тела; m_x – цикловая масса топлива; ω_t – скорость сгорания топлива; φ_z и \bar{m} – показатели сгорания топлива; F – сила; V – скорость поршня; ω, φ – частота вращения и угол поворота коленчатого вала; M – крутящий момент; t – время; $m_{пр}$ – приведенная масса частей двигателя, совершающих возвратно-поступательное движение; $I_{пр}$ – приведенный момент инерции вращающихся частей двигателя; f_{ct} – площадь поверхности; $f_{п}$ – площадь поршня; r_k – радиус кривошипа; λ – длина шатуна; индекс «ос» – окружающая среда.

В практических расчетах вместо фундаментальной переменной – эксергии – используются дополнительные – давление и температура.

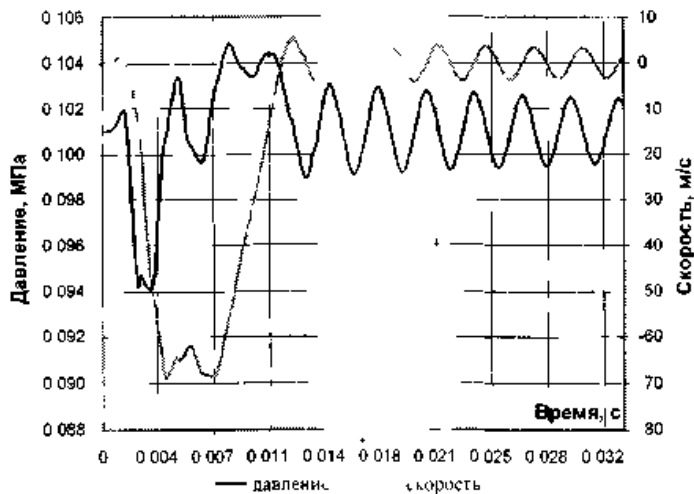
Кононюк А.Е. Графы

Уравнения пересчета, позволяющие осуществить пересчет фундаментальных переменных в дополнительные, приведены в литературе.

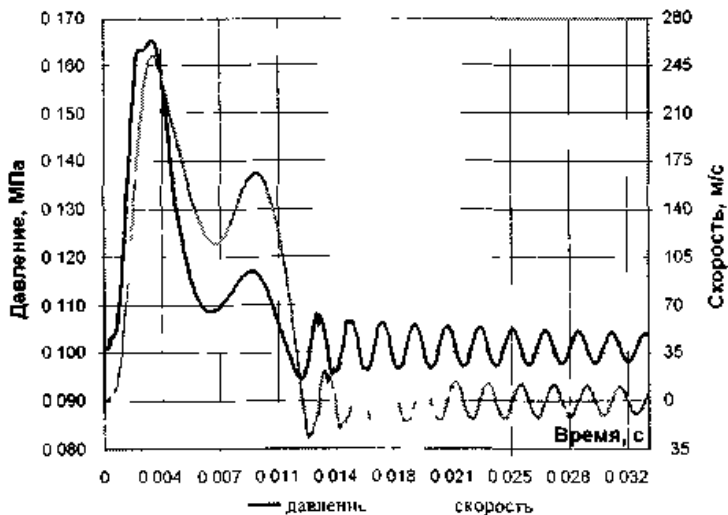
Изложенные выше принципы формулирования условий на границах взаимодействующих систем ПДВС заложены в основу иерархической системы моделей ПДВС.

В качестве примера рассмотрим особенности формулирования условий на границах впускной и выпускной систем. Элементы рассеивания R в данном случае определяют интегральные потери эксергии при движении газа во впускной и выпускной системах. При использовании для описания названных систем математических моделей нестационарного течения газа, базирующихся на численных методах, граничные условия будут формулироваться следующим образом. На каждом временном шаге основное ядро определяет текущие значения давления и температуры газа в цилиндре, впускная и выпускная системы определяют текущие значения расхода (прихода) рабочего тела.

На рис. 2 и 3 приведены результаты расчетов впускной и выпускной систем дизеля ТМЗ-450Д, базирующиеся на изложенных выше принципах.



a



b

Рис. 2. Изменение параметров потока во впускной (а) и выпускной (б) системах в течение цикла

Кононюк А.Е. Графы

Сравнение расчетных и экспериментальных данных, показанное на рис. 3, свидетельствуют об адекватности применяемого математического описания.

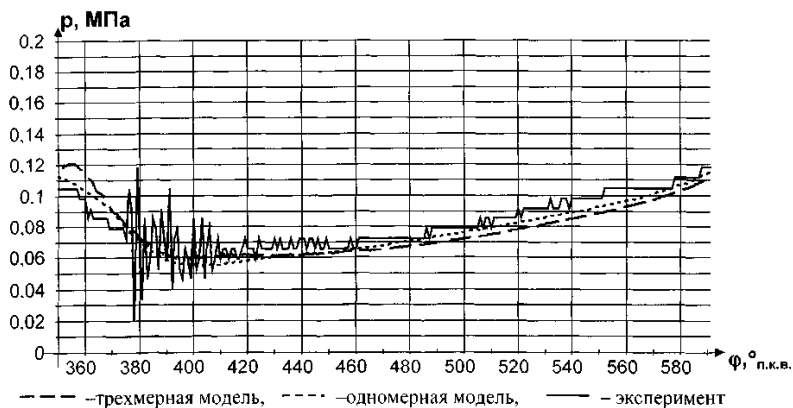


Рис. 3. Сравнение расчетных и экспериментальных значений давления в цилиндре в период впуска ($n = 3600$ об/мин, $M_c = 17$ Н-м)

Таким образом, применение аппарата графов связей позволяет раскрыть структурные особенности ПДВС; определить причинно-следственные отношения в ПДВС, устанавливающие вход-выход для отдельных элементов и ПДВС в целом; реализовать единый подход к формулированию условий на границах взаимодействующих систем ПДВС с целью совместного применения существующих математических моделей в системах проектирования ПДВС.

1.12. Графы связей и размещение узлов в «сетях – на - кристалле»

Е. А. Суворова, Ю. Е. Шейнин

Рост числа элементов на кристалле при переходе к субмикронным интегральным технологиям позволяет создавать системы-на-кристалле (СнК) с большим числом вычислительных узлов – Мультипроцессорные СнК (МПСнК, MPSoC). Основным направлением развития СнК с большим количеством процессоров становится создание систем с распределенной архитектурой, в которых вычислительные узлы взаимодействуют путем обмена сообщениями.

При переходе от мультиядерных СнК с несколькими процессорными ядрами к системам с большим числом процессоров, основным направлением развития становится создание «сетей-на-кристалле» (*Network-on-Chip, NoC*). Сети-на-кристалле представляют собой системы с распределенной архитектурой, включающие в себя десятки вычислительных узлов, использующих для взаимодействия между собой механизм обмена сообщениями. Каждый вычислительный узел включает в себя процессор, локальную память, контроллер связей с другими узлами и, возможно, еще ряд блоков специального назначения.

При разработке NoC может использоваться богатый научный задел в области параллельных систем с распределенной архитектурой, систем с обменом сообщениями в СВТ других классов. Выбор и оптимизация конфигурации и топологии системы связей между узлами параллельной ВС является классической задачей в этой области, где получено большое число теоретически важных и практически значимых результатов [1], [2], [3]. Однако новые возможности и специфические требования и ограничения реализации СнК требуют анализа эффективности классических методов проектирования и оптимизации, ревизии системы показателей графов связей параллельной СнК, разработки новых методов их оптимизации.

Можно выделить следующие специфические требования и ограничения реализации системы связей между узлами в NoC:

Кононюк А.Е. Графы

ограниченное число слоев металлизации; ограниченная площадь, занимаемая рисунком схемы; согласованность, ограниченное число межуровневых переходов линий связи, допустимая длина линий связи, ограниченное количество связей кристалла с внешними каналами СБИС [4].

Характеристики системы рассматриваются на двух уровнях представления – на структурном уровне и на уровне рисунка схемы.

Традиционно на структурном уровне используются следующие характеристики сложности: количество вычислительных узлов на кристалле, количество связей между ними и вектор валентности. Для СпК важной характеристикой сложности, определяющей возможность реализации в соответствии с выбранной технологией, становится минимальное количество слоев металлизации, необходимое для размещения структуры связей на кристалле. Минимальное количество слоев металлизации определяется «книжной» толщиной графа связей. При определении книжной толщины граф разбивается на «страницы» - планарные подграфы. Часть ребер исходного графа может не принадлежать страницам, а переходит от страницы к странице в любом месте рисунка схемы. Это наиболее точно отражает особенности технологии, [5]. Количество страниц определяет книжную толщину графа.

Для оценки времени обмена сообщениями на структурном уровне используется средний диаметр графа связей.

На уровне рисунка схемы выполняется оценка показателей позволяющих определить возможность и эффективность реализации структурной схемы для выбранной топологии. Для определения сложности реализации конкретного рисунка схемы на кристалле существенны следующие численные характеристики, [4]:

- 1) количество слоев металлизации, необходимое для размещения рисунка схемы,
- 2) геометрические размеры кристалла, необходимые для размещения рисунка схемы,
- 3) количество переходов линий связи из слоя в слой.

Кононюк А.Е. Графы

Для представления рисунка схемы в целом широко используется модель Томпсона, [6]. В соответствии с этой моделью граф связей внедряется в двумерную решетку. Рисунки вычислительных узлов представляются квадратами со стороной h . Линии связи могут прокладываться горизонтально и вертикально вдоль линий решетки. Различные линии могут пересекаться под прямым углом, но не могут перекрываться. Точки изменения направления линий не могут совпадать. Считается, что линии связи между узлами имеют единичную толщину. Площадь, занимаемая схемой в целом, определяется как площадь прямоугольника минимального размера, на котором можно разместить все вычислительные узлы и линии связи. Если для линий связи отводится два слоя металлизации, то в одном из них располагаются фрагменты линий, расположенные вертикально, а в другом – фрагменты линий, расположенные горизонтально. Каждый изгиб линии связи соответствует ее переходу из слоя в слой. Поскольку в ряде случаев может быть сгенерирован рисунок схемы, количество переходов линий связи в котором будет меньше, модель Томпсона позволяет оценить верхнюю границу количества переходов.

При разработке и анализе характеристик рисунка схемы до этапа синтеза с использованием САПР в соответствии с конкретной технологией не возможно учесть ряд факторов, необходимых для корректного использования классической или трехмерной модели Томпсона: ширина линий связи, размеры отверстий межслойных переходов, расстояние между линиями связи, наличие экранирующих линий, возможность размещения фрагментов шин в разных слоях, расположение портов узлов, влияние на рисунок схемы расположения линий в активных слоях.

В соответствии с этим для оценки длины линий связи и количества переходов линий связи из слоя в слой нами предлагается использовать упрощенную *модель условных линий связи*.

В модели условных линий связи так же, как и в модели Томпсона, множество вычислительных узлов системы внедряется в двумерную решетку. Но узлы в решетке расположены вплотную друг к другу. С учетом того, что геометрические размеры рисунков узлов существенно больше возможных промежутков между ними, это не внесет существенных искажений в оценку длины линии. Условные линии связей соединяют геометрические центры рисунков модулей. Это позволяет абстрагироваться от реальных положений портов.

Кононюк А.Е. Графы

Ограничения на прокладку линий по линиям сетки такие же, как и в модели Томпсона.

Обозначим длину условной линии связи L_u . Узлы могут располагаться в соседних ячейках решетки по вертикали или горизонтали. В этом случае длина условной линии связи между модулями равна линейному размеру модуля h ($L_u=h$). Узлы могут располагаться не рядом, но в одном столбце или в одной строке решетки. В этом случае $L_u=(k+1) \times h$, где k – количество узлов, находящихся между связываемыми. Узлы могут располагаться в разных столбцах и в разных строках. В этом случае $L_u=(k+1+m+1) \times h$, где k – количество узлов, находящихся между связываемыми в строке, m – количество узлов, находящихся между связываемыми в столбце.

Обозначим количество переходов одной условной линии связи из слоя в слой – N_u . Рассмотрим те же варианты, что и при определении условной длины линий связи. Будем считать, что если узлы расположены в соседних ячейках решетки по горизонтали или вертикали, линия связи между ними целиком расположена в одном слое металлизации. Для такой связи $N_u=0$. Когда узлы расположены в одном столбце или в одной строке, но не рядом, их центры также могут быть соединены вертикальной или горизонтальной линией. Однако, если какие-либо из модулей, расположенных между рассматриваемыми, также связаны, то линии связи в этом случае перекроются, что не допустимо. Рассмотрим, как этого можно избежать на примере узлов, расположенных в одной строке. Будем считать, что линия связи состоит из трех участков. Первый и третий (имеют пренебрежимо малую длину) – расположены вертикально, позволяют сместить линию связи относительно центров узлов вверх или вниз. Второй участок горизонтальный. Таким образом, на линии связи будет два перехода из слоя в слой – $N_u=2$. Эта оценка может быть несколько завышенной по сравнению с реальной ситуацией, поскольку в синтезированном рисунке геометрические места соединяемых портов различных модулей могут находиться не на одной линии, связи в этом случае перекрываются не будут. Аналогично линия связи между модулями, находящимися в разных столбцах и строках максимально может состоять из четырех отрезков. Первый (пренебрежимо короткий) предназначен для отведения линии в сторону от центра узла, второй и третий предназначены для приближения линии к нужному узлу, четвертый, как и первый предназначен для приведения линии к центру узла. Таким образом,

Кононюк А.Е. Графы

линия связи насчитывает три перехода из слоя в слой – $Nu=3$. Эта оценка так же несколько завышена. Таким образом, предлагаемая модель позволяет получить оценку верхней границы количества переходов линий связи из слоя в слой.

В соответствии с определенной системой критериев и параметров можно определить следующую последовательность действий, позволяющую выбрать топологию связей и рисунок схемы в соответствии с требованиями, предъявляемыми разработчиком.

1) Определение толщины графа связей для рассматриваемых топологий. Позволяет определить возможность использования топологий для выбранной технологии.

2) Построение рисунков схем для отобранных в соответствии с пунктом 1 топологий. Вычисление для них критериев сложности, на базе которых определяется возможность реализации этих рисунков по выбранной технологии. Оценка возможности масштабирования рисунков схем.

3) Определение характеристик скорости передачи сообщения (среднего диаметра графа связей) для выбранных в соответствии с пунктом 2 топологий. Оценка возможности масштабирования топологий. Выбирается топология с наименьшим значением среднего диаметра.

4) Определение размещения множества внешних узлов. Выбирается топология, для которой среднее время доступа к информации за пределами кристалла минимально.

С учетом новых показателей, исследованы классические регулярные (близкие к регулярным) топологии систем связей параллельных ВС. Доказано, что книжная толщина тора, трехмерной решетки, гиперкуба, δ сетей не превосходит 2. Таким образом, эти графы связей потенциально могут быть использованы для NoC. Определены ограничения на количества узлов в графах связей указанных выше типов, позволяющие разместить их в малом количестве слоев металлизации (3 – 4 слоя).

Рассмотрены различные варианты формирования рисунков схем для трехмерной решетки, тора и гиперкуба. Для тора и трехмерной

Кононюк А.Е. Графы

решетки исследовано влияние на характеристики рисунка схемы метода чередования узлов. Для тора исследовано влияние на характеристики рисунка схемы метода кластеризации при различном размере кластера. Определена возможность и эффективность применения различных методов построения рисунков схем при различных технологических ограничениях.

Рассмотрим, например, варианты формирования рисунков схем для трехмерной решетки $M \times N \times L$. Эту структуру можно рассмотреть как совокупность двумерных матриц размера $M \times N$. В дальнейшем будем называть их уровнями решетки. Если на кристалле для размещения рисунков узлов отводится одна группа слоев металлизации, уровни решетки могут быть размещены в нем так, как показано на рисунке 1 а. Обозначим такой способ формирования рисунка – *способ 1*.

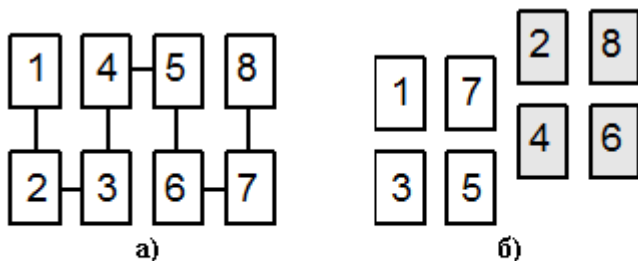


Рис. 1. Варианты расположения модулей трехмерной решетки а) в одной группе слоев металлизации, б) в двух группах слоев металлизации

Рассмотрим, как повлияет на характеристики рисунка применение метода чередования узлов – *способ 2*. Элементы уровней решетки 1 и 2 (рисунок 1,б 1 – окрашены белым, 2 – серым) можно чередовать, например, по горизонтали. Это иллюстрируется рисунком 2. Если рисунок одного узла занимает три слоя металлизации, то один из них может использоваться для прокладки вертикальных линий связи между узлами, второй – для прокладки горизонтальных линий связи между узлами, принадлежащими первому уровню, и третий – для прокладки линий связи, принадлежащих второму уровню. Такой рисунок схемы может быть использован, если в соответствии с технологией не может

Кононюк А.Е. Графы

быть выделено дополнительных слоев металлизации для прокладки линий связи. (Книжная толщина

Рис. 2. Размещение узлов решетки с чередованием

графа связей равна 2) Если $L > 2$, то рисунки пар уровней могут располагаться относительно друг друга так, как это было рассмотрено на рисунке 1,б. Модули уровней решетки, окрашенных серым, чередуются с модулями неокрашенных уровней решетки. При $N > 1/2 \times ((3 \times M + 2) / M)$ использование этого способа позволяет сократить суммарную длину условных линий связи, если для прокладки линий связи могут быть использованы активные слои, 2 способ позволяет сократить количество переходов линий связи из слоя в слой в два раза по сравнению с первым способом

Кононюк А.Е. Графы

Гиперкуб Для генерации рисунка схемы гиперкуба может быть использован подход кластеризации. В рамках этого подхода исходный гиперкуб может быть разбит на гиперкубы меньшей степени по-разному.

Первый способ. Гиперкуб степени 2 может быть легко внедрен в двумерную решетку. Он определяется в качестве кластера минимального размера. На базе четырех таких кластеров может быть вновь построен кластер степени 2, он будет соответствовать гиперкубу степени 4. Так продолжается до тех пор, пока не будет построен гиперкуб требуемой размерности. Это иллюстрируется рисунком 3.

Второй способ. Граф, соответствующий гиперкубу степени 4 является изоморфным графу, соответствующему тору размером 4x4. Это иллюстрируется рисунком 4. В соответствии с этим рисунок гиперкуба может быть построен на базе кластеров, соответствующих гиперкубам размерности 4'4. В этом случае для модулей, находящихся в середине решетки 4'4 будут задействованы все линии связи соседними узлами, что позволяет сократить количество переходов из слоя в слой. Суммарная длина линий связи для второго способа будет такой, же, как и для первого. Использование второго способа позволяет сократить суммарное количество переходов линий связи из слоя в слой по

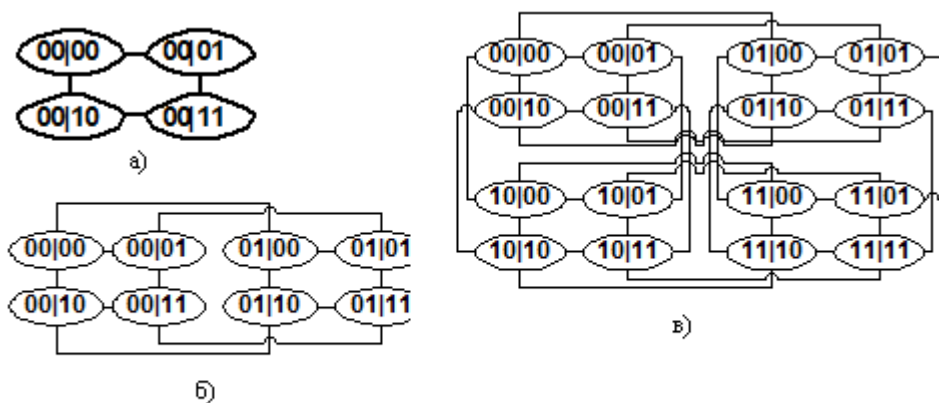


Рис. 3. Формирование рисунка схемы гиперкуба на базе кластеров 2^2

Кононюк А.Е. Графы

сравнению с использованием первого способа в $2n$ раз.

Рассмотрим проблему размещения узлов, связанных с внешними каналами СБИС, в графе NoС. Число внешних выводов СБИС растет существенно медленнее, чем сложность СБИС, число элементов на кристалле. Верхняя граница количества каналов связи с внешним миром определяется количеством выводов кристалла и количеством выводов, необходимых для организации одного канала связи в соответствии с выбранным стандартом.

Множество узлов, которые имеют связи с внешними каналами, будем называть множеством *внешних узлов*. Полагается, что каждый из внутренних узлов обменивается информацией с внешним миром через тот (те) внешний узел, расстояние до которого минимально (внутренний узел «прикреплен» к соответствующему внешнему узлу). Если внутренний узел прикреплен к n внешним узлам (расстояние до них от внутреннего узла одинаково и минимально), то поток обращений от этого внутреннего узла равномерно распределяется между ними.

Размещение множества внешних узлов определяет среднее время доступа к информации за пределами кристалла. Время обращения к информации за пределами кристалла - T_v - может быть оценено по формуле 1.

$$T_v = ((D_v - 1) \times T_o + D_v \times T_r) + (T_b + T_d); \quad (1)$$

где

D_v

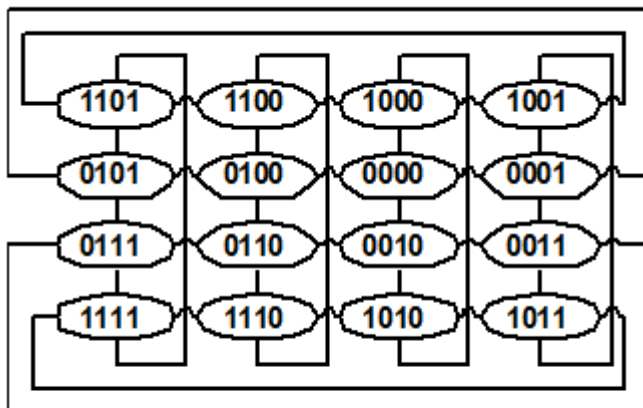


Рис. 4. Изоморфизм тора 4×4 и гиперкуба 2^4

Кононюк А.Е. Графы

среднее расстояние от узла системы до внешнего узла

T_0 – среднее время пребывания в очереди на обработку в транзитном внутреннем узле,

T_g – среднее время обмена сообщениями между внутренними узлами,

T_b – среднее время пребывания в очереди на обработку во внешнем узле,

T_d – среднее время обмена сообщением с внешним устройством.

Для минимизации времени обращения к информации за пределами кристалла необходимо минимизировать D_v , T_0 , T_b (T_g и T_d зависят от выбранной технологии). Тенденции развития технологии таковы, что T_d на порядок превосходит T_g . T_0 и T_b зависят от равномерности распределения потоков сообщений по внутренним узлам и по внешним узлам [4], [7]. В соответствии с этим сформулировано и доказано следующее:

1. Для минимизации времени доступа к информации за пределами кристалла, наряду с минимизацией D_v , необходимо, чтобы потоки обращений к внешней среде равномерно распределялись между внешними узлами.

2. Не во всех случаях множество внешних узлов можно выбрать таким образом, чтобы интенсивности потоков заявок через все внешние модули были одинаковы.

Обозначим среднюю интенсивность потока сообщений от одного узла I_f (если система находится в стационарном режиме и обращения равномерно распределяются между узлами, то средняя интенсивность потока сообщений через внутренний модуль - I_f), среднюю интенсивность потока сообщений к внешней среде от одного узла I_v , среднюю интенсивность потока сообщений через один внешний узел I_u . Будем считать, что характеристики потоков сообщений близки к экспоненциальным.

Средняя интенсивность потока заявок от внутреннего узла i к внешнему узлу j (к которому прикреплен узел i) можно определить как

Кононюк А.Е. Графы

lv/ki , где ki – количество внешних узлов, к которым прикреплен внутренний узел i . Средняя интенсивность потока заявок через внешний узел j может быть определена следующим образом:

$$\lambda_{u^j} = \sum_i \lambda_v \cdot \frac{1}{k_i} = \lambda_v \cdot \sum_i \frac{1}{k_i} \quad (2)$$

(Суммирование выполняется по всем внутренним узлам, прикрепленным к внешнему узлу j) В идеальном случае потоки заявок ко всем внешним узлам одинаковы и

$$\lambda_u = \frac{V}{V_0} \cdot \lambda_v \quad (3)$$

Обозначим допустимое значение отклонения от среднего значения ld .

$$\lambda_d = \max(|\lambda_{u_i} - \lambda_u|) \quad (4)$$

В терминах теории графов задача размещения множества внешних узлов может быть сформулирована следующим образом. Введем следующие обозначения.

J – множество вершин графа ($|J|=V$),

J_n – множество вершин, соответствующих внутренним узлам, ($|J_n|=V_n$),

J_0 – множество вершин графа, соответствующих внешним узлам, ($|J_0|=V_0$)

Обозначим кратчайшее расстояние между вершинами v_i и v_j – $d(v_i, v_j)$. На структурном уровне все ребра идентичны – $d(v_i, v_j)$ равно количеству ребер, входящих в кратчайший путь между i и j . Граф неориентированный, поэтому $d(v_i, v_j)=d(v_j, v_i)$.

Кононюк А.Е. Графы

Пусть J_x – произвольное подмножество вершин $J \setminus i$. Обозначим

$$r(v_i, J_x) = \min_{v_t \in J_x} (d(v_i, v_t))$$

– минимальное расстояние от вершины v_i до множества вершин J_x . Обозначим $S(J_x)$ – суммарное расстояние от всех вершин графа G до множества вершин J_x .

(5)

На базе $S(J_x)$ можно определить среднее расстояние до множества вершин J_x $D_v(J_x)$ следующим образом:

$$D_v(J_x) = S(J_x) / V; \quad (6)$$

Исходя из сделанного выше предположения о [программном обеспечении](#), функционирующем в системе, узел $v_i \in J_n$ будет обмениваться информацией с внешним миром через узел $v_j \in J_0$, для которого $d(v_i, v_j) = \min(d(v_i, J_0))$. Будем говорить в этом случае, что узел v_i прикреплен к узлу v_j .

Если для узла $v_i \in J_n$ $v_{j1}, v_{j2}, \dots, v_{jn} \in J_0$, $d(v_i, v_{j1}) = d(v_i, v_{j2}) = \dots = d(v_i, v_{jn}) = \min(d(v_i, J_0))$, то поток информации, которой узел v_i обменивается с внешним миром, равномерно распределяется между $v_{j1}, v_{j2}, \dots, v_{jn}$. Обозначим это как кратность прикрепления v_i к J_0 , равная n . Обозначим $K_n(v_i, v_{jk} \in J_0) = 1/n$ – коэффициент кратности прикрепления узла v_i к узлу v_{jk} . Если $v_i \in J_n$ прикреплена только к одной вершине v_{j1} , то $K_n(v_i, v_{j1}) = 1$. Если $v_i \in J_n$ не прикреплена к вершине v_j , то $K_n(v_i, v_j) = 0$.

Вследствие сделанных выше предположений о потоках обращений к внешней среде, средняя интенсивность потока заявок через внешний узел прямо пропорциональна количеству прикрепленных к нему внутренних узлов с учетом коэффициента прикрепления. Обозначим $K_p(v_j \in J_0)$ – суммарный коэффициент кратности прикрепления к узлу v_j .

$$K_p(v_j \in J_0) = \sum_{i=1}^N K_n(v_i, v_j) \quad (7)$$

Кононюк А.Е. Графы

Интенсивность потока заявок во внешнюю среду через внешний узел v_j может быть определена по следующей формуле:

$$lv_j = K_p(v_j) \times lv; \quad (8)$$

Обозначим отношение общего количества узлов к количеству внешних узлов $KL = V/V_o$ – среднее количество узлов, связанных с одним внешним узлом. Тогда $lu = lv \times KL$. Отклонение интенсивности потока заявок через внешний узел от среднего значения прямо пропорционально отклонению суммарного количества узлов, подключенных к внешнему узлу с учетом коэффициента подключения, от среднего. Обозначим отклонение суммарного количества узлов, подключенных к v_j от среднего DLv_j

$$DLv_j = |K_p(v_j) \hat{I} J_o - KL| \text{ для } v_j \hat{I} J_o \quad (9)$$

Обозначим максимальное отклонение от среднего для всех $v_j \hat{I} J_o$ – $DL(J_o)$

$$DL(J_o) = \max_{v \in \mathcal{G}_r} D_r(v) \quad (10)$$

В системе отклонение интенсивности потока заявок к внешнему миру через один внешний узел от среднего значения не превосходит lL .

$$lL = lv \times DL(J_o); \quad (11)$$

Обозначим DM максимально допустимое отклонение от KL , такое, что $lM = lv \times DM$.

Пусть в множестве вершин графа G можно выделить X подмножеств J_x , $|J_x| = V_o$, для которых $DL(J_x) \leq DM$. Тогда в соответствии с утверждениями 1 и 2 для заданного графа G множество J_o определяется как

$$S(\mathcal{G}_o) = \min_{\mathcal{G}_r \in \mathcal{G}} (S(\mathcal{G}_r)) \quad (12)$$

Кононюк А.Е. Графы

$Dv = Dv(J_0)$;

$$|v_j - l_j| \leq |v_j - l_j| + M; \quad (13)$$

Формулировка задачи поиска множества внешних вершин сходна с формулировкой классической задачи кратных медиан (P-медиан) графа, [8], которая используется для решения так называемых минисуммных задач размещения. В этих задачах критерий оптимальности состоит в минимизации суммы (или другой функции) расстояний – так называемой стоимости – от вершин графа до некоторого подмножества вершин этого же графа – J_p , называемого медианным подмножеством или p-медианой графа. ($|J_p| = P$). Граф может содержать одну или несколько p-медиан.

Если в классической задаче нахождения p-медианы применительно к проблеме размещения внешних узлов используется один критерий –

$$S(\mathcal{G}_p) = \min_{\mathcal{G}_r \subset \mathcal{G}} (S(\mathcal{G}_r))$$

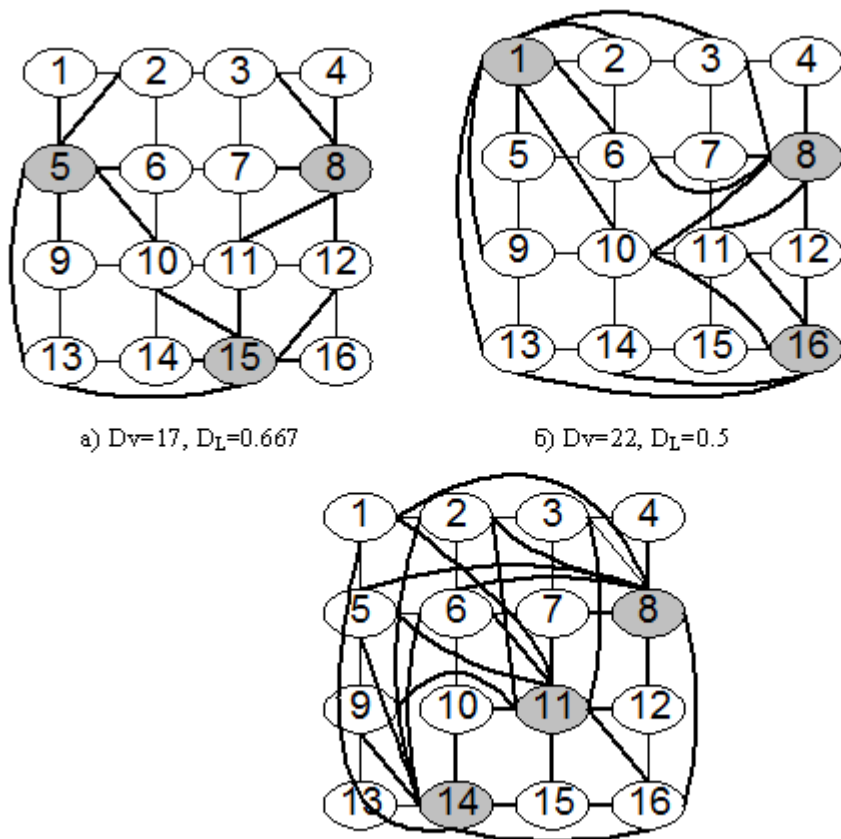
суммарное расстояние, соответственно $Dv = S(J_0) / N_v$ минимально возможное для данной топологии при выбранном количестве внешних узлов, то в задаче нахождения множества внешних вершин первичным критерием отбора множеств является то, что, в соответствии с утверждениями 1 и 2, DL (максимальное отклонение суммарного количества модулей, подключенных к одному внешнему модулю от среднего) не должно превышать заданной величины DM. В качестве вариантов решения рассматриваются только те подмножества вершин, которые удовлетворяют заданному значению этого критерия. Среди них отбираются те, для которых суммарное расстояние до них от всех вершин графа минимально. В результате в качестве решения этой задачи может быть получено множество, не являющееся p-медианой графа.

Для решения этой задачи разработан алгоритм, отличающийся от известного алгоритма нахождения p-медианы. Для решения минисуммных задач размещения используются метод направленного древовидного поиска и линейного программирования. Однако, при использовании линейного программирования могут быть получены не только целочисленные решения и переход от них к целочисленным решениям (необходимость которых определяется предметной областью) связан с большой вычислительной сложностью. В связи с

Кононюк А.Е. Графы

этим предложенный алгоритм является алгоритмом направленного древовидного поиска.

Множества, получаемые в результате использования этого алгоритма не всегда совпадают с медианными. Рассмотрим, например, топологию типа двумерная решетка, включающую в себя 16 вершин (4×4) при количестве вершин, имеющих связи с внешним миром равном 3, наблюдается аналогичная ситуация. Если определить $DM=0.5$, то для отбираемых множеств внешних вершин $Dv=22$, $DL=0.5$. В то время, как для медианных вершин $Dv=17$, DL находится в диапазоне от 0.667 до 1.833. Если для этой же топологии определить $DM=0.4$, то для множеств, удовлетворяющих этим условиям, $Dv=24$, $DL=0.3$. Примеры множеств внешних модулей для этих случаев приведены на рисунке 5. Равномерность загрузки внешних узлов достигается за счет увеличения суммарного расстояния до внешних узлов



а) $Dv=17, DL=0.667$

б) $Dv=22, DL=0.5$

Рис. 5. Варианты выбора множества внешних узлов

На этом рисунке серым цветом закрашены внешние узлы, тонкими линиями изображены физически существующие связи между узлами, жирными линиями отмечены прикрепления внутренних узлов к внешним. Для топологии, представленной на рисунке 5 при уменьшении DL в 2 раза, Dv увеличивается почти в 1,5 раза. Если время передачи сообщения между двумя соседними узлами в пределах кристалла в 10 раз меньше, чем время обмена сообщениями между внешним узлом и устройством за пределами кристалла (типичная для современных технологий ситуация [4]), то среднее время обмена сообщениями с внешним миром для варианта рис. 5, б в 1,3 раза меньше, чем для варианта рис. 5, а. Среднее время обмена

сообщениями с внешним миром для варианта рис. 5, в (приведенное на этом рисунке множество обеспечивает наиболее равномерное прикрепление для данной топологии) в 1,7 раза меньше, чем для варианта рис. 5, а. Таким образом, использование предложенного алгоритма позволило найти для данного примера множество внешних узлов, при котором время обмена сообщениями с внешним миром будет минимальным при данных условиях.

в) $Dv=24$, $DL=0.33$

2. Сети Петри

2.1. ОСНОВНЫЕ ПОНЯТИЯ

Реальные дискретные системы состоят из разнообразных компонентов, различающихся физическими свойствами, функциональным назначением, сложностью внутренней структуры. Для того чтобы сконструировать адекватный математический аппарат, предназначенный для моделирования систем, необходимо установить круг вопросов, которые должны решаться с помощью моделей, и осуществить переход от физических сущностей к их абстракциям, сначала в форме некоторого, (ограниченного) набора концептуальных понятий, затем — в точных математических терминах.

Следующие проблемы могут служить примерами тех задач, которые часто возникают при конструировании и исследовании дискретных систем: выполняет ли система те функции, для которых она предназначена; функционирует ли она эффективно; могут ли в ней возникнуть ошибки и аварийные ситуации; имеются ли в ней потенциально узкие места; можно ли упростить систему или заменить ее отдельные компоненты и подсистемы на более совершенные, не нарушая ее общего функционирования; можно ли из данных систем сконструировать более сложную, отвечающую заданным требованиям, и т.д. Эти задачи носят в основном "качественный", а не количественный характер.

В этой главе описаны основные принципы сетевого подхода к моделированию дискретных систем и вводятся базовые понятия теории сетей Петри.

2.1.1. Системы, процессы, сети

Первый шаг на пути к построению модели дискретной системы — это абстрагирование от конкретных физических и функциональных особенностей ее компонентов. Компоненты системы и их действия представляются абстрактными *событиями*, каковыми могут быть, например, исполнение оператора программы, переход триггера из состояния в состояние, прерывание в операционной системе, операция станка или конвейерной линии, завершение этапа проекта и т.п.

Событие может произойти (реализоваться) один раз, повториться многократно, порождая конкретные *действия* (реализации события), или не произойти ни разу. Совокупность действий, возникающих как реализации событий при функционировании дискретной системы, образует *процесс*, порождаемый этой системой. В общем случае одна и та же система может функционировать в одних и тех же условиях по-разному, порождая некоторое множество процессов, т.е. функционировать недетерминированно.

Реальная система функционирует во времени, события происходят в некоторые моменты времени и длятся некоторое время. В синхронных моделях дискретных систем события явно привязаны к определенным моментам или интервалам времени, в которые происходит одновременное изменение состояний всех компонентов системы, трактуемое как изменение общего состояния системы. Смена состояний происходит последовательно. Этот подход к моделированию больших параллельных систем имеет ряд недостатков. Во-первых, в большой системе приходится учитывать состояние всех компонентов при каждой смене ее общего состояния, что делает модель громоздкой, особенно в тех случаях, когда локальные изменения касаются небольшого фрагмента системы.

Во-вторых, при таком подходе исчезает информация о причинно-следственных связях между событиями в системе. Например, если два события при функционировании системы произошли одновременно, то мы не знаем, произошло ли это случайно или в этом факте скрыт какой-то функциональный смысл. Такие понятия, как конфликты между компонентами системы (из-за ресурсов) или ожидание одним из компонентов результатов работы других компонентов, трудно выражаются в терминах смены состояний системы.

В-третьих, в так называемых асинхронных системах события могут происходить внутри неопределенно больших интервалов времени, заранее трудно или нельзя указать более точно время их начала, конца и длительность.

Кононюк А.Е. Графы

Выходом может служить отказ от введения в модели дискретных систем времени и тактированных последовательностей изменений состояний, а замена их — причинно-следственными связями между событиями. Модели такого типа (в том числе сети Петри) называют *асинхронными*. (Если возникает необходимость осуществить привязку ко времени, то моменты или интервалы времени представляют как события. Таким образом, существенно синхронные системы могут описываться в терминах асинхронных моделей.) Замена временных связей причинно-следственными дает возможность более наглядно описать структурные особенности функционирования систем.

Отказ от времени приводит к тому, что события в асинхронной модели рассматриваются или как элементарные (неделимые, "мгновенные"), или как составные, имеющие некоторую внутреннюю структуру, образованную из "подсобытий". При неформальном описании функционирования асинхронных моделей, в частности, сетей Петри, мы будем привлекать временные отношения (раньше, позже, не одновременно и т.п.), когда это удобно или привычно, но они представляют собой лишь результаты причинно-следственных отношений.

Взаимодействие событий в больших асинхронных системах имеет, как правило, сложную динамическую структуру. Эти взаимодействия описываются более просто, если указывать не непосредственные связи между событиями, а те ситуации, при которых данное событие может реализоваться. При этом глобальные ситуации в системе формируются с помощью локальных операций, называемых *условиями* реализации событий.

Условие имеет емкость: условие не выполнено (емкость равна 0), условие выполнено (емкость равна 1), условие выполнено с n -кратным запасом (емкость равна n , где n - целое положительное число). Условие соответствует таким ситуациям в моделируемой системе, как наличие данного для операции в программе, состояние некоторого регистра в устройстве ЭВМ, наличие деталей на конвейере и т.п. Определенные сочетания условий разрешают реализоваться некоторому событию (предусловия события), а реализация события изменяет некоторые условия (постусловия события), т.е. события взаимодействуют с условиями, а условия — с событиями.

Таким образом, предполагается, что для решения указанных в начале главы задач достаточно представлять дискретные системы как структуры, образованные из элементов двух типов — событий и условий.

В сетях Петри события и условия представлены абстрактными символами из двух непересекающихся алфавитов, называемых

Кононюк А.Е. Графы

соответственно множеством *переходов* и множеством *мест*. В графическом представлении сетей переходы изображаются "барьерами", а места — кружками (рис. 1, 2 и 3).

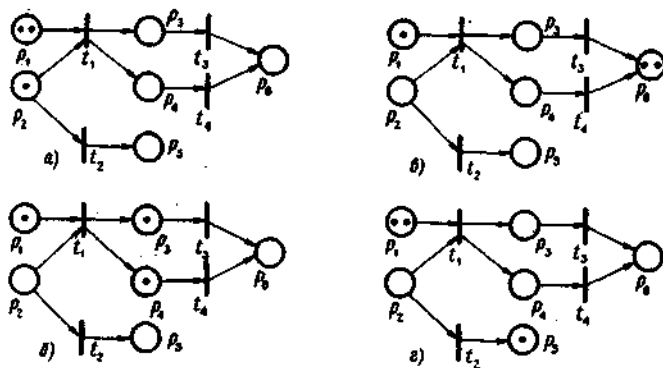


Рис. 1.

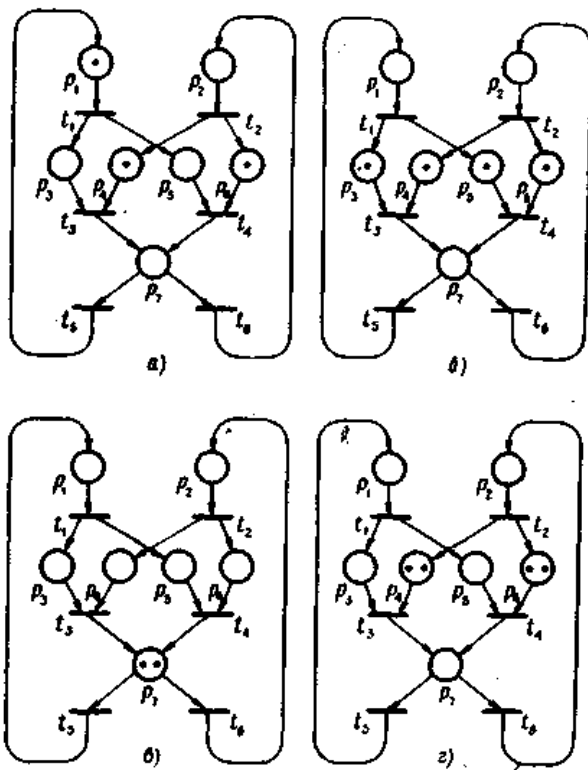


Рис. 2

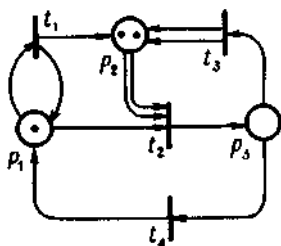


Рис. 3

Условия-места и события-переходы связаны отношением непосредственной зависимости (непосредственной причинно-следственной связи), которое изображается с помощью направленных дуг, ведущих из мест в переходы и из переходов в места. Места, из которых ведут дуги на данный переход, называются его *входными* местами. Места, на

Кононюк А.Е. Графы

которые ведут дуги из данного перехода, называются его *выходными* местами. В сети на рис. 1.1,а места p_1 и p_2 являются входными для перехода t_1 , а места p_3 и p_4 — выходными. В этом примере событие-переход t_1 непосредственно зависит от условий-мест p_1 и p_2 , а места p_3 и p_4 непосредственно зависят от t_1 . В этой же сети место p_2 является входным одновременно для двух переходов t_1 и t_2 , место p_6 является выходным одновременно для двух переходов t_3 и t_4 . Выполнение условия изображается *разметкой* соответствующего места, а именно помещением числа n или n фишек (маркеров) в это место, где $n > 0$ - емкость условия:

- p ○ — условие p не выполнено,
- p ○ — условие p выполнено,
- p ○ — условие p имеет емкость 3,
- p ⊖ — условие p имеет емкость 5.

Динамика поведения моделируемой системы находит свое отражение в функционировании (работе) сети Петри. Неформально работу сети можно представить как совокупность локальных действий, которые называются *срабатываниями* переходов. Они соответствуют реализациям событий и приводят к изменению разметки мест, т.е. к локальному изменению условий в системе.

Переход может сработать, если выполнены все условия реализации соответствующего события. Например, для так называемых ординарных сетей Петри (частный случай принятой в настоящее время версии сетей Петри, введенный им в первой работе) все входные места перехода должны содержать хотя бы по одной фишке.

Срабатывание перехода - неделимое действие, изменяющее разметку его входных и выходных мест следующим образом из каждого входного места изымается по одной фишке, а в каждое выходное место добавляется по одной фишке. Тем самым реализация события, изображаемого переходом, изменяет состояние (емкость) непосредственно связанных с ним условий так, что емкость предусловий, вызвавших реализацию этого события, уменьшается, а емкость постусловий, на которые оно влияет, увеличивается. Переход t_1 на рис. 1, а может сработать, так как оба его входных места p_1 и p_2 содержат фишки, а после срабатывания t_1 разметка его входных и выходных мест изменяется так, как показано на рис. 1, б.

Если два (и более) перехода могут сработать и они не имеют общих входных мест, то их срабатывания являются независимыми

Кононюк А.Е. Графы

действиями, осуществляемыми в любой последовательности или параллельно.

Если несколько переходов могут сработать и имеют общее входное место (как переходы t_1 и t_2 на рис. 1,а), то срабатывает только один, любой из них. При этом может оказаться, что, сработав, этот переход лишит возможности сработать другие переходы (рис. 1, б и з). Таким способом в сети моделируется конфликт между событиями, когда реализация одного события может исключить возможность реализации других. В сети никак не указывается, каким образом конфликт следует фактически разрешить. Считается, что решение о том, какое из конфликтующих событий следует реализовать, принимается вне формализма сети, т.е. поведение сети носит недоопределенный недетерминированный характер. Аналогичный конфликт возникает в том случае, когда несколько переходов могут сработать и они имеют общие выходные места, как переходы t_3 и t_4 (рис. 1, б и в).

В процессе функционирования сети происходит смена разметок мест как результат срабатывания ее переходов. Сеть останавливается, если ни один из ее переходов не может сработать (рис. 1, в и г).

На рис. 2, а показан еще один пример сети Петри с некоторой начальной разметкой мест, при которой может сработать только переход t_1 , так как его единственное входное место p_1 содержит фишку. Переходы t_2 , t_5 и t_6 имеют по одному входному месту, не содержащему фишки, поэтому они не могут сработать. Переходы t_3 и t_4 имеют по два входных места. Это означает, что общее условие реализации события, представленного переходом t_3 или t_4 , является конъюнкцией из двух условий. Для каждого из переходов-событий выполнено лишь одно условие, поэтому ни t_3 ни t_4 не могут сработать.

В результате срабатывания перехода t_1 место p_1 лишится фишки, а места p_3 и p_5 получат по одной фишке (рис. 2, б). Изменилась разметка сети и при новой разметке могут сработать два перехода - t_3 и t_4 . Срабатывание любого из них помещает фишку в место p_7 , после чего возможно срабатывание переходов t_5 и t_6 . При этом, если срабатывают оба перехода t_3 и t_4 до того, как сработают t_5 и t_6 , то место p_7 будет содержать две фишки (рис. 2, в), в противном случае - одну. Место p_7 - общее входное место для переходов t_5 и t_6 . Если p_7 содержит одну фишку, то сможет сработать только один из переходов t_5 или t_6 , так как сработавший переход заберет единственную фишку. Если же p_7 содержит две фишки, то возможны различные продолжения работы сети:

- (1) работает переход t_5 , затем - переход t_6 ;
- (2) работает переход t_6 , затем — переход t_5 ;

Кононюк А.Е. Графы

(3) дважды сработает t_5 , а t_6 не может реализовать возможность срабатывания;

(4) дважды сработает t_6 , а t_5 не сможет сработать ни разу.

В первых двух случаях места p_1 и p_2 получают по фишке, в последних двух случаях одно из мест p_1 или p_2 будет содержать две фишки, а второе не будет иметь фишек. Если p_1 и p_2 содержат по фишке, то срабатывание переходов t_1 и t_2 приведет к разметке, которая уже возникла ранее в процессе функционирования сети (рис. 2, б), и последующая работа сети будет повторять описанную выше. Если же одно место содержит две фишки (например, p_2), то сработать может (дважды) только один переход (в данном случае t_2), после чего каждый из переходов t_3 и t_4 имеет в одном входном месте две фишки, а в другом - ни одной (рис. 2, г). Возникает разметка, при которой ни один из переходов сети не может сработать, и сеть Петри останавливается.

Анализ работы сети Петри, показанный на рис. 2, позволяет сделать некоторые выводы о функционировании моделируемой сетью системы (это может быть, например, фрагмент операционной системы, состоящей из параллельных циклических процессов, взаимодействующих друг с другом). В частности, можно отметить, что система способна функционировать циклически как угодно долго, но может и остановиться в "тупиковой" ситуации, показанной на рис. 2, г. Таким образом, сети Петри формализуют понятие абстрактной асинхронной системы - динамической структуры из событий и условий. В общей теории сетей сеть Петри рассматривается как один из способов сетевого моделирования систем. Вводятся более общие сетевые модели. Их единую основу образует понятие неинтерпретированной ориентированной сети из условий и событий, которая описывает только статическое строение системы (формальное определение сети будет дано позже). Самой общей в спектре динамических сетевых моделей является, по-видимому, условно-событийная система, которая представляет собой сеть, дополненную правилами изменений условий в результате реализации событий. Сеть Петри можно считать конкретизацией условно-событийной системы.

Если сеть Петри описывает функциональную схему моделируемой системы, то работа сети моделирует процесс, происходящий при функционировании системы. Поскольку процесс протекает во времени, для его изучения нужно зафиксировать его в форме некоторой "истории процесса", которую обычно отождествляют с самим процессом. Недетерминированный характер функционирования асинхронной системы и соответствующей сети Петри приводит к тому, что система может порождать не единственный процесс, а множество

Кононюк А.Е. Графы

возможных процессов. Кроме того, процессы, порождаемые такими системами, являются параллельными.

Возникает вопрос, каким образом формализовать понятие параллельного процесса, в какой системе понятий можно удобно и полно описывать параллельные процессы (а также множества параллельных процессов) и изучать их. Другими словами, возникает необходимость в разработке моделей параллельных процессов. Поскольку параллельный процесс можно рассматривать как дискретную динамическую систему, то в этом случае можно использовать сетевую модель, которая является частным случаем условно-событийной системы. Модели такого типа будем называть реализационными сетями, или сетями-процессами. Они представляют собой сети, в общем случае бесконечные, с дополнительными ограничениями на структуру связей между условиями и событиями и на начальную разметку условий.

Возможность описывать системы и порождаемые ими процессы в рамках одного и того же формализма сетей позволяет не только унифицировать математический аппарат теории систем и процессов, но и более наглядно выявлять связи между функциональными и операционными свойствами систем.

2.1.2. Формальное определение сети Петри

Зафиксируем основные математические обозначения, которые будем использовать в формальных определениях.

$\{a, b, \dots, c\}$ — множество, состоящее из элементов a, b, \dots, c ;

(a, b, \dots, c) — упорядоченное множество (набор), состоящее из элементов a, b, \dots, c ;

$\{x/A(x)\}$ — множество значений переменной x , удовлетворяющих условию $A(x)$;

$x, \dots, y \in X$ — элементы x, \dots, y принадлежат множеству X ;

$X, \dots, Y \subseteq Z$ — множества X, \dots, Y являются частью (подмножествами) множества Z ;

$\notin, \not\subseteq$ — отрицание отношений \in и \subseteq ;

$|X|$ — мощность (число элементов) множества X ;

\emptyset — пустое множество;

\cup, \cap, \setminus — теоретико-множественные операции объединения, пересечения и разности множеств;

$A_1 \times \dots \times A_n$ — декартово произведение множеств A_1, \dots, A_n ;

$A^n = \underbrace{A \times \dots \times A}_n, n \geq 2$;

Кононюк А.Е. Графы

\mathbb{N} - множество всех натуральных чисел (включая 0);

$\vee, \wedge, \neg, \Rightarrow$ - логические операции дизъюнкции, конъюнкции, отрицания и импликации;

$A \Leftrightarrow B$ — A истинно тогда и только тогда, когда истинно B ;

\exists, \forall — кванторы существования и всеобщности (например,

$\forall x \in X, \exists y \in Y: A(x, y)$ — для любого x , принадлежащего X , существует y , принадлежащее Y , такое, что истинно условие $A(x, y)$).

Пусть X, Y — множества, $R \subseteq X \times X$ — бинарное отношение на X , и $x, y \in X$.

xRy — x и y находятся в отношении R ;

R^{-1} — обращение отношения R , т.е. $yR^{-1}x$, если и только если xRy ;

\bar{R} - дополнение отношения R , т.е. $\bar{R} = (X \times X) \setminus R$;

$R \circ S$ - суперпозиция отношений R и S , т.е.

$$R \circ S = \{ (x, y) \mid \exists z \in X: xRz \wedge zSy \};$$

$$R^n = \underbrace{R \circ \dots \circ R}_n, \quad n \geq 2;$$

$$R^+ = \bigcup_{n=0}^{\infty} R^n, \quad \text{где } n \in \mathbb{N}, \text{ — транзитивное замыкание отношения } R.$$

Для функции f запись $f: X \rightarrow Y$ указывает область определения функции X и область ее значений Y .

Конечное множество символов $A = \{a, b, \dots, c\}$ образует алфавит. Словом в алфавите A называется некоторая последовательность выписанных подряд символов из алфавита A (например, $abbacb$).

Конкатенацией двух слов α и β в алфавите A называется слово $\alpha\beta$, т.е. слово, полученное выписыванием подряд слов α и β (слова α и β являются в этом случае полсловами слова $\alpha\beta$). Через A^* обозначим множество всех слов в алфавите A , через λ — пустое слово, т.е. слово, не содержащее ни одного символа. Через a^n и α^n , где a — символ, α — слово или подслово, обозначают слова (подслова)

$$\underbrace{aa \dots a}_n \text{ и } \underbrace{\alpha\alpha \dots \alpha}_n. \text{ Если слово представляет собой конкатенацию } \beta\gamma,$$

то β называют префиксом, а γ — суффиксом слова $\beta\gamma$. Если A и B — алфавиты такие, что $A \supseteq B$, то слово β в алфавите B , полученное вычеркиванием из слова α в алфавите A всех символов из $A \setminus B$, будем называть проекцией слова α в алфавите A на алфавит B .

Через $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$ обозначим расширенное множество натуральных чисел, где ω — элемент, удовлетворяющий для любого $n \in \mathbb{N}$ следующим свойствам:

$$1) \omega > n,$$

$$2) n + \omega = \omega + n = \omega + \omega = \omega - n = \omega - \omega = \omega,$$

$$3) \omega \cdot n = n \cdot \omega = \omega,$$

$$4) 0 \cdot \omega = \omega \cdot 0 = 0.$$

Кононюк А.Е. Графы

Пусть $K = (k_1, \dots, k_r)$, $L = (l_1, \dots, l_r) \in \mathbf{N}^r$ — векторы из r чисел. Арифметические операции распространим на векторы, считая их покомпонентными операциями. Например, $K + L = (k_1 + l_1, \dots, k_r + l_r)$. Отношения же переопределим для векторов как глобальные отношения, с использованием кванторов существования и всеобщности. Например,

$$(K \leq L) = \begin{cases} \text{истина, если } \forall i, 1 \leq i \leq r: k_i \leq l_i, \\ \text{ложь в противном случае,} \end{cases}$$

$$(K = L) = \begin{cases} \text{истина, если } \forall i, 1 \leq i \leq r: k_i = l_i, \\ \text{ложь в противном случае,} \end{cases}$$

$$(K < L) = \begin{cases} \text{истина, если } (\forall i, 1 \leq i \leq r: k_i \leq l_i) \wedge (\exists i, 1 \leq i \leq r: k_i < l_i), \\ \text{ложь в противном случае.} \end{cases}$$

Сетью будем называть тройку (P, T, F) где

P — непустое множество элементов сети, называемых *местами*,

T — непустое множество элементов сети, называемых *переходами*,

$F \subseteq P \times T \cup T \times P$ — отношение *инцидентности*, и для (P, T, F)

выполнены следующие условия:

A1) $P \cap T = \emptyset$ (множества мест и переходов не пересекаются);

A2) $(F \neq \emptyset) \wedge (\forall x \in P \cup T, \exists y \in P \cup T: xFy \vee yFx)$ (т.е. любой элемент сети инцидентен хотя бы одному элементу другого типа);

A3) если для произвольного элемента сети $x \in X$ обозначить через x

множество его *входных элементов* $\{y \mid yFx\}$, а через x' — множество его *выходных элементов* $\{y \mid xFy\}$, то

$$\forall p_1, p_2 \in P: (p_1 = p_2) \wedge (p_1' = p_2') \Rightarrow (p_1 = p_2).$$

(т.е. сеть не содержит пары мест, которые инцидентны одному и тому же множеству переходов).

Графическим представлением сети служит двудольный ориентированный граф (в общем случае бесконечный) с двумя типами вершин; вершины-места изображаются кружочками, вершины-переходы - барьерами (а также прямоугольниками или квадратами, если переходы являются неэлементарными объектами). Из вершины x в вершину y ведет дуга, если и только если xFy .

На основе понятия сети, которая описывает только статическую топологию моделируемого процесса или системы, вводятся динамические сетевые структуры, в которых местам приписываются специальные разметки, моделирующие выполнение условия, и с сетью связывается понятие ее функционирования, изменяющего эти разметки (условия) в результате так называемых *срабатываний переходов*. К

Кононюк А.Е. Графы

таким динамическим сетям относятся сети Петри, их различные варианты, обобщения и частные случаи.

Сеть Петри - это набор $N = (P, T, F, W, M_0)$, где (P, T, F) - конечная сеть (множество $X = P \cup T$ конечно), а $W: F \rightarrow \mathbb{N} \setminus \{0\}$ и $M_0: P \rightarrow \mathbb{N}$ - две функции, называемые соответственно *кратностью дуг* и *начальной разметкой*. Первая сопоставляет каждой дуге число $n > 0$ (кратность дуги). Если $n > 1$, то в графическом представлении сети число n выписывается рядом с короткой чертой, пересекающей дугу. Часто такая дуга будет также заменяться пучком из n дуг, соединяющих соответствующие элементы сети. Условимся никак не отмечать кратность дуг, равную 1. Такую сеть будем называть *ординарной*). Вторая функция сопоставляет каждому месту $p \in P$ некоторое число $M_0(p) \in \mathbb{N}$ (разметка места). В графическом представлении сети разметка места p изображается помещением в вершину-кружок числа $M_0(p)$ или, если это число невелико, соответствующего числа точек (фишек).

Функционирование сети Петри описывается формально с помощью множества последовательностей срабатываний и множества достижимых в сети разметок. Эти понятия определяются через правила срабатывания переходов сети.

Разметка сети N — это функция $M: P \rightarrow \mathbb{N}$. Если предположить, что все места сети N строго упорядочены каким-либо образом, т.е.

$P = (p_1, \dots, p_n)$, то разметку M сети (в том числе начальную разметку) можно задать как вектор чисел $M = (m_1, \dots, m_n)$ такой, что для любого

$i, 1 \leq i \leq n, m_i = M(p_i)$. Если $P' = \{p_{i_1}, \dots, p_{i_k}\}$ — подмножество мест из P , то условимся через $M(P')$ обозначать множество разметок

$\{M(p_{i_1}), \dots, M(p_{i_k})\}$. Если P' представить как вектор $P' = (p_{i_1}, \dots, p_{i_k})$,

то $M(P')$ обозначает вектор из множества \mathbb{N}^k , называемый *проекцией* разметки M на P' . На основе отношения инцидентности F и функции кратности дуг W можно ввести *функцию инцидентности*

$F: P \times T \cup T \times P \rightarrow \mathbb{N}$, которая определяется как

$$F(x, y) = \begin{cases} n, & \text{если } xFy \wedge (W(x, y) = n), \\ 0, & \text{если } \neg(xFy). \end{cases}$$

Если места сети упорядочены, то можно каждому переходу t сопоставить два целочисленных вектора $F(t)$ и $F^*(t)$ длиной n , где $n = |P|$:

$$F(t) = (b_1, \dots, b_n), \quad \text{где } b_i = F(p_i, t),$$

$$F^*(t) = (b_1, \dots, b_n), \quad \text{где } b_i = F(t, p_i).$$

Кононюк А.Е. Графы

Переход t может сработать при некоторой разметке M сети N , если

$\forall p \in \cdot t: M(p) \geq F(p, t)$, т.е. каждое входное место p перехода t имеет разметку, не меньшую, чем кратность дуги, соединяющей p и t . Это условие можно переписать в векторной форме следующим образом:

$$M \geq F(t).$$

Для ординарной сети Петри условие срабатывания перехода означает, что любое входное место этого перехода содержит хотя бы одну фишку, т.е. имеет ненулевую разметку.

Срабатывание перехода t при разметке M порождает разметку M' по следующему правилу:

$$\forall p \in P: M'(p) = M(p) - F(p, t) + F(t, p), \text{ т.е.}$$

$$M' = M - F(t) + F^*(t).$$

Таким образом, срабатывание перехода t изменяет разметку так, что разметка каждого его входного места p уменьшается на $F(p, t)$, т.е. на кратность дуги, соединяющей p и t , а разметка каждого его выходного места увеличивается на $F(t, p)$, т.е. на кратность дуги, соединяющей t и p .

На множестве разметок можно ввести отношение $[>$ непосредственного следования разметок:

$$M [> M' \iff \exists t \in T: (M \geq F(t)) \wedge (M' = M - F(t) + F^*(t)).$$

Будем использовать уточняющее обозначение $M [t > M'$, если M' непосредственно следует после M в результате срабатывания перехода t .

Говорят, что разметка M' достижима от разметки M , если существует последовательность разметок M, M_1, M_2, \dots, M' и слово $\tau = t_1 t_2 \dots t_k$ в алфавите T такие, что

$$M [t_1 > M_1 [t_2 > M_2 \dots [t_k > M'.$$

Слово τ в этом случае называется последовательностью срабатываний, ведущих от M к M' . Обобщим отношения непосредственного следования до отношения " M' достижима от M ", используя обозначение $M [> M'$ или $M [\tau > M'$, если уточняется последовательность срабатываний. (Последовательность τ может быть пустой, т.е. M достижима от M .)

Множество $\{M' \mid M [> M'\}$ разметок, достижимых в сети N от разметки M , обозначим через $R(N, M)$. Множество $R(N) = R(N, M_0)$, т.е. множество всех разметок, достижимых в N от начальной разметки M_0 , называют множеством достижимых разметок сети N . (Заметим, что $M \in R(N, M)$ и $M_0 \in R(N)$.)

Множеством последовательностей срабатываний сети N , или свободным языком сети N , называется множество

$$L(N) = \{ \tau \in T^* \mid \exists M \in R(N): M_0 [\tau > M \}.$$

Кононюк А.Е. Графы

т.е. множество всех последовательностей срабатываний, ведущих от M_0 к каждой достижимой в N разметке.

На рис. 3 изображена сеть Петри, на примере которой поясним данные выше определения. В этой сети $P = (p_1, p_2, p_3)$, $T = \{t_1, t_2, t_3, t_4\}$. Функция инцидентности F задается с помощью следующих двух таблиц, в которых на пересечении строки x и столбца y стоит число $F(x, y)$:

	p_1	p_2	p_3
t_1	1	1	0
t_2	0	0	1
t_3	0	2	0
t_4	1	0	0

	t_1	t_2	t_3	t_4
p_1	1	1	0	0
p_2	0	2	0	0
p_3	0	0	1	1

Начальная разметка M_0 задается следующим образом: $M_0(p_1) = 1$, $M_0(p_2) = 2$, $M_0(p_3) = 0$ или, в векторной форме: $M_0 = (1, 2, 0)$.

При разметке M_0 могут сработать переходы t_1 и t_2 , так как

$M_0 = (1, 2, 0) \geq F(t_1) = (1, 0, 0)$, $M_0 \geq F(t_2) = (1, 2, 0)$. Переходы t_3 и t_4 не могут сработать, так как $M_0 \not\geq F(t_3) = (0, 0, 1)$, $M_0 \not\geq F(t_4) = (0, 0, 1)$.

В результате срабатывания перехода t_1 разметка M_0 сменяется на разметку $(1, 3, 0)$, а в результате срабатывания перехода t_2 разметка M_0 сменяется на разметку $(0, 0, 1)$. Обе новые разметки непосредственно следуют после M_0 в рассматриваемой сети. Можно представить возможные изменения разметок сети N , происходящие в результате срабатывания ее переходов, в виде *графа разметок* — ориентированного графа, множество вершин которого образовано множеством $R(N)$ достижимых в N разметок. Из вершины M в вершину M' ведет дуга, помеченная символом перехода t , если и только если $M[t > M'$. На рис. 4 показан начальный фрагмент графа разметок сети на рис. 3.

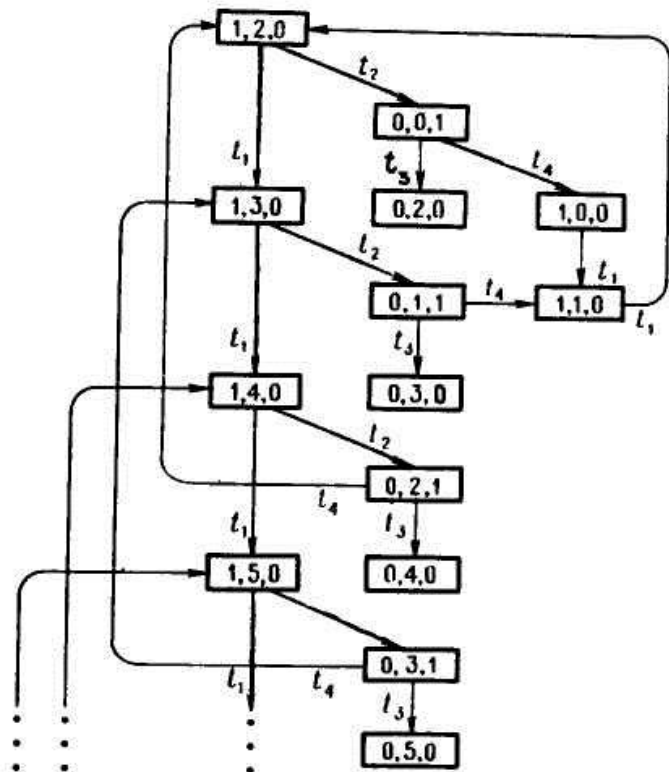


Рис. 4

Этот граф бесконечен, так как множество $R(N)$ достижимых разметок бесконечно для рассматриваемой сети.

Разметка $M \in R(N)$ называется *тупиковой*, если в сети N не существует ни одного перехода, который может сработать при этой разметке. Для рассматриваемой сети тупиковыми являются разметки $(0, 2, 0), (0, 3, 0), (0, 4, 0), \dots, (0, n, 0), \dots$

Легко видеть, что если выделить путь по дугам графа разметок, начинающийся в вершине M и заканчивающийся в вершине M' , и выписать подряд все встречающиеся символы переходов, то полученное слово образует последовательность срабатываний, ведущих от M к M' . Множество всех слов, полученных выписыванием символов переходов вдоль путей, начинающихся в M_0 , образует множество последовательностей срабатываний сети, или ее свободный язык. Так, язык рассматриваемой сети включает слова

Кононюк А.Е. Графы

$(\lambda, t_1, t_2, t_1t_1, t_1t_2, t_2t_3, t_2t_4, t_1t_1t_1, t_1, t_1t_2, t_1t_2t_3, t_1t_2t_4, t_2t_4t_1, t_1t_1t_1t_1, t_1t_1t_1t_2, t_1t_1t_2t_3, t_1t_1t_2t_4, t_1t_2t_4t_1, \dots)$. Следующая теорема характеризует эффект увеличения начальной разметки в сети.

Теорема 1. Пусть $N = (P, T, F, W, M_0)$, $N' = (P, T, F, W, M_0 + K)$, где $K \in \mathbb{N}^{|P|}$. Тогда

- а) $M \in R(N) \Rightarrow (M + K) \in R(N')$;
- б) $L(N) \subseteq L(N')$;
- в) $M_1 [t] M_2 \Rightarrow (M_1 + K) [t] (M_2 + K)$.

Доказательство. Воспользуемся индукцией по длине срабатываемых последовательностей. Прежде всего $M_0 \in R(N)$ и $(M_0 + K) \in R(N')$.

Заметим также, что $M_0 [\lambda] M_0$, где λ — пустая последовательность, $\lambda \in L(N)$ и $\lambda \in L(N')$. Предположим, что

$$M_0 [t] M \text{ в } N \text{ и } (M_0 + K) [t] (M + K).$$

Покажем, что

$$\forall t \in T: M [t] M' \Rightarrow (M + K) [t] (M' + K).$$

Если t может сработать при разметке M в сети N , т.е. $M \geq F(t)$, то t может сработать и при разметке $(M + K) \geq M$ в сети N' . Это означает, что последовательность срабатываний tt принадлежит как $L(N)$, так и $L(N')$. (Если t не может сработать при M в N , то это не исключает возможности срабатывания t при $(M + K) \geq M$ в сети N' .) После срабатывания перехода t в N и соответственно в N' разметка в обеих сетях изменяется следующим образом:

$$M - F(t) + F'(t) = M',$$

$$(M + K) - F(t) + F'(t) = (M - F(t) + F'(t)) + K = M' + K.$$

Тем самым установлена справедливость соотношений а) и б). Аналогично можно убедиться в том, что имеет место соотношение в).

2.1.3. Структура сети Петри

Сеть Петри состоит из четырех элементов: множество *позиций* P , множество *переходов* T , *входная* функция I и *выходная* функция O . Входная и выходная функции связаны с переходами и позициями. Входная функция I отображает переход t_j в множество позиций $I(t_j)$, называемых входными позициями перехода. Выходная функция O отображает переход t_j в множество позиций $O(t_j)$, называемых выходными позициями перехода.

Структура сети Петри определяется ее позициями, переходами, входной и выходной функциями.

Определение 1. Сеть Петри C является четверкой, $C = (P, T, I, O)$.

Кононюк А.Е. Графы

$P = \{p_1, p_2, \dots, p_n\}$ — конечное множество *позиций*, $n \geq 0$. $T = \{t_1, t_2, \dots, t_m\}$ — конечное множество *переходов*, $m \geq 0$. Множество позиций и множество переходов не пересекаются, $P \cap T = \emptyset$. $I : T \rightarrow P^\infty$ является входной функцией — отображением из переходов в комплекты позиций. $O : T \rightarrow P^\infty$ есть выходная функция — отображение из переходов в комплекты позиций.

Мощность множества P есть число n , а мощность множества T есть число m . Произвольный элемент P обозначается символом p_i , $i=1, \dots, n$, а произвольный элемент T — символом t_j , $j=1, \dots, m$.

Примеры сетей Петри даны на рис. 1—3.

Позиция p_i является *входной позицией* перехода t_j в том случае, если $p_i \in I(t_j)$; p_i является *выходной позицией*, если $p_i \in O(t_j)$. Входы и выходы переходов представляют собой *комплекты* позиций. *Комплект* является обобщением множества, в которое включены многократно повторяющиеся элементы — тиражированные элементы. В приложении содержится описание теории комплектов. Использование комплектов, а не множеств для входов и выходов перехода позволяет позиции быть кратным входом либо кратным выходом перехода. *Кратность* входной позиции p_i для перехода t_j есть число появлений позиции во входном комплекте перехода, $\#(p_i, I(t_j))$. Аналогично кратность выходной позиции p_i для перехода t_j есть число появлений позиции в выходном комплекте перехода, $\#(p_i, O(t_j))$. Если входная и выходная функции являются множествами (а не комплектами), то кратность каждой позиции есть либо 0, либо 1.

Входные и выходные функции используются для отображения позиций в комплекты переходов, а также их можно использовать для отображения переходов в комплекты позиций. Определим, что переход t_j является входом позиции p_i , если p_i есть выход t_j . Переход t_j есть выход позиции p_i , если p_i есть вход t_j .

Определение 2. Определим расширенную входную функцию I и выходную функцию

$$I : P \rightarrow T^\infty, \quad O : P \rightarrow T^\infty$$

таким образом, что

$$\#(t_j, I(p_i)) = \#(p_i, O(t_j)), \quad \#(t_j, O(p_i)) = \#(p_i, I(t_j)).$$

Для сети Петри на рис. 1 расширенными входной и выходной функциями являются:

$$\begin{array}{ll}
 I(p_1) = \{ \}, & O(p_1) = \{t_1\}, \\
 I(p_2) = \{t_1, t_4\}, & O(p_2) = \{t_2\}, \\
 I(p_3) = \{t_1, t_4\}, & O(p_3) = \{t_2, t_3\}, \\
 I(p_4) = \{t_3\}, & O(p_4) = \{t_4\}, \\
 I(p_5) = \{t_1, t_2\}, & O(p_5) = \{t_2\}.
 \end{array}$$

$$\begin{array}{l}
 C = (P, T, I, O), \\
 P = \{p_1, p_2, p_3, p_4, p_5\}, \\
 T = \{t_1, t_2, t_3, t_4\}, \\
 I(t_1) = \{p_1\}, \quad O(t_1) = \{p_2, p_3, p_5\}, \\
 I(t_2) = \{p_2, p_3, p_5\}, \quad O(t_2) = \{p_5\}, \\
 I(t_3) = \{p_3\}, \quad O(t_3) = \{p_4\}, \\
 I(t_4) = \{p_4\}, \quad O(t_4) = \{p_2, p_3\}.
 \end{array}$$

Рис. 1. Структура сети Петри представлена в виде четверки, которая состоит из множества позиций (P), множества переходов (T), входной функции $I: T \rightarrow P^\infty$ и выходной функции ($O: T \rightarrow P^\infty$).

$$\begin{array}{l}
 C = (P, T, I, O), \\
 P = \{p_1, p_2, p_3, p_4, p_5, p_6\}, \\
 T = \{t_1, t_2, t_3, t_4, t_5\}, \\
 I(t_1) = \{p_1\}, \quad O(t_1) = \{p_2, p_3\}, \\
 I(t_2) = \{p_3\}, \quad O(t_2) = \{p_3, p_5, p_5\}, \\
 I(t_3) = \{p_2, p_3\}, \quad O(t_3) = \{p_2, p_4\}, \\
 I(t_4) = \{p_4, p_5, p_5, p_5\}, \quad O(t_4) = \{p_4\}, \\
 I(t_5) = \{p_2\}, \quad O(t_5) = \{p_6\}.
 \end{array}$$

Рис. 2. Структура сети Петри.

$$C = (P, T, I, O),$$

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\},$$

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6\},$$

$$I(t_1) = \{p_1\},$$

$$O(t_1) = \{p_2, p_3\},$$

$$I(t_2) = \{p_8\},$$

$$O(t_2) = \{p_1, p_7\},$$

$$I(t_3) = \{p_2, p_5\},$$

$$O(t_3) = \{p_6\},$$

$$I(t_4) = \{p_3\},$$

$$O(t_4) = \{p_4\},$$

$$I(t_5) = \{p_6, p_7\},$$

$$O(t_5) = \{p_9\},$$

$$I(t_6) = \{p_4, p_8\},$$

$$O(t_6) = \{p_5, p_8\}.$$

Рис. 3. Структура сети Петри.

2.1.4. Графы сетей Петри

В значительной степени теоретическая работа по сетям Петри основана на формальном определении сетей Петри, изложенном выше. Тем не менее для иллюстрации понятий теории сетей Петри гораздо более удобно графическое представление сети Петри. Теоретико-графовым представлением сети Петри является двудольный ориентированный мультиграф.

Структура сети Петри представляет собой совокупность позиций и переходов. В соответствии с этим граф сети Петри обладает двумя типами узлов. *Кружок* \circ является позицией, а *планка* $|$ — переходом.

Ориентированные дуги (стрелки) соединяют позиции и переходы, при этом некоторые дуги направлены от позиций к переходам, а другие — от переходов к позициям. Дуга, направленная от позиции p_i к переходу t_j определяет позицию, которая является входом перехода. Кратные входы в переход указываются кратными дугами из входных позиций в переход. Выходная позиция указывается дугой от перехода к позиции. Кратные выходы также представлены кратными дугами.

Сеть Петри есть *мультиграф*, так как он допускает существование кратных дуг от одной вершины графа к другой. Следует добавить, что так как дуги являются направленными, то это ориентированный мультиграф. Мы знаем, что вершины графа можно разделить на два

Кононюк А.Е. Графы

множества (позиции и переходы) таким образом, что каждая дуга будет направлена от элемента одного множества (позиций или переходов) к элементу другого множества (переходов или позиций); следовательно, такой граф является *двудольным* ориентированным мультиграфом. В дальнейшем для простоты будем называть его просто графом сети Петри.

Определение 3. *Граф G сети Петри* есть двудольный ориентированный мультиграф, $G = (V, A)$, где $V = \{u_1, u_2, \dots, u_s\}$ — множество вершин, а $A = \{a_1, a_2, \dots, a_r\}$ — комплект направленных дуг, $a_i = (v_j, v_k)$, где $v_j, v_k \in V$. Множество V может быть разбито на два непересекающихся подмножества P и T , таких, что $V = P \cup T$,

$P \cap T = \emptyset$, и для любой направленной дуги $a_i \in A$, если $a_i = (v_j, v_k)$, тогда либо $v_j \in P$ и $v_k \in T$, либо $v_j \in T$, а $v_k \in P$.

Графы сети Петри, изображенные на рис. 4 — 6, эквивалентны структурам сети Петри на рис. 1 — 3.

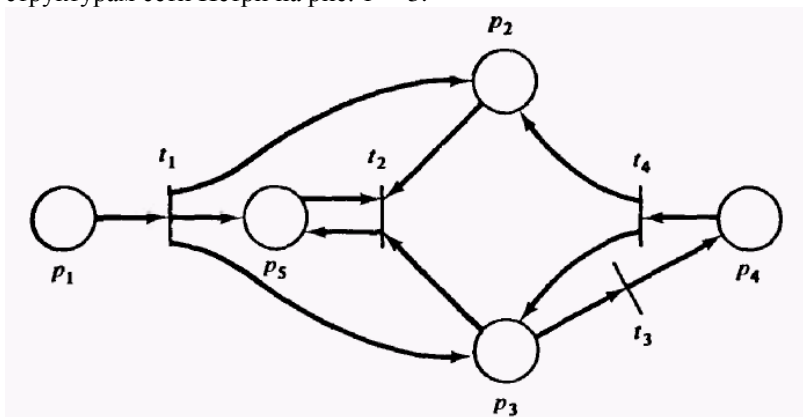


Рис. 3. Граф сети Петри, эквивалентный структуре, показанной на рис. 1.

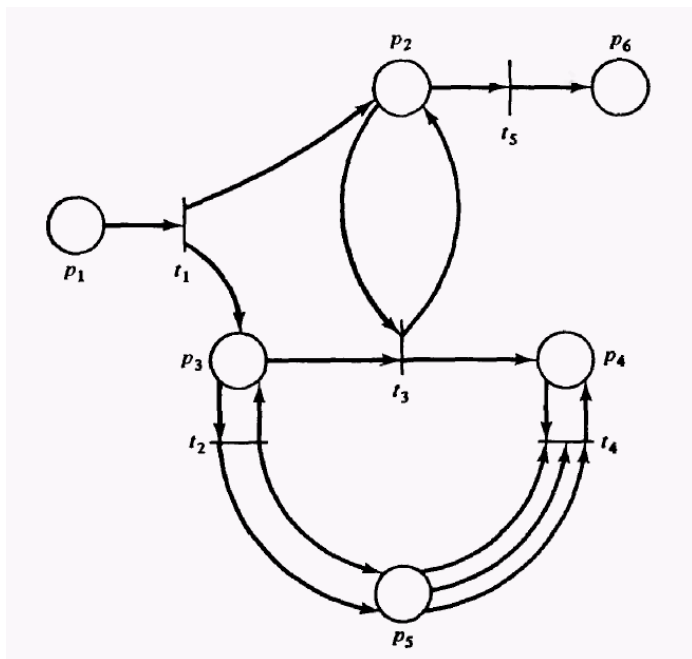


Рис. 5. Граф сети Петри, эквивалентный структуре, изображенной на рис.2.

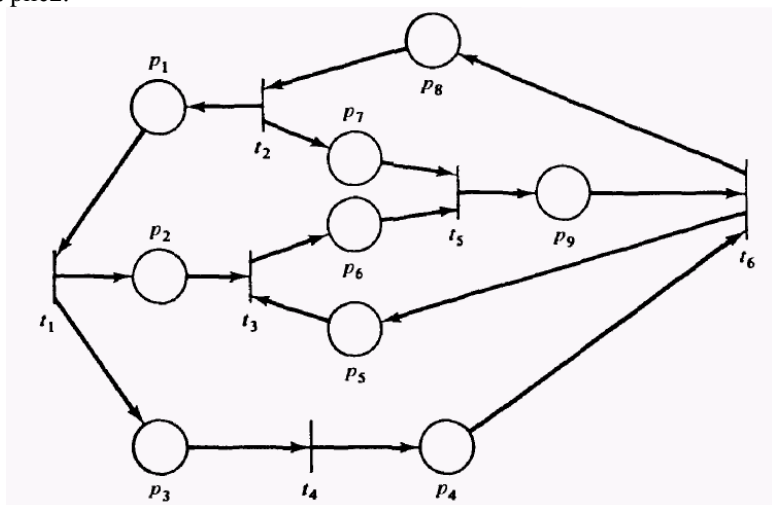


Рис. 6. Граф сети Петри, эквивалентный структуре, показанной на рис. 3.

Кононюк А.Е. Графы

Для демонстрации эквивалентности этих двух представлений сети Петри — структуры сети Петри и графа сети Петри — покажем, каким образом можно преобразовать один в другой. Предположим, нам дана структура сети Петри $C = (P, T, I, O)$ с $P = \{p_1, p_2, \dots, p_n\}$ и $T = \{t_1, t_2, \dots, t_m\}$. Тогда граф сети Петри можно определить следующим образом.

Определение 4. Определим $V = P \cup T$. Определим A как комплект направленных дуг, такой, что для всех $p_i \in P$ и $t_j \in T$

$$\#((p_i, t_j), A) = \#(p_i, I(t_j)),$$

$$\#((t_j, p_i), A) = \#(p_i, O(t_j)).$$

$G = (V, A)$ есть граф сети Петри, эквивалентный структуре сети Петри $C = (P, T, I, O)$.

Обратное преобразование (от графа сети Петри к структуре) осуществляется подобным образом, и поэтому оставим более детальное описание такого преобразования читателю. Однако при переходе от графа сети Петри к структуре сети Петри возникает следующая задача: если множество вершин можно разделить на два подмножества S и R , то какое из этих подмножеств должно быть позициями, а какое — переходами? Оба возможных варианта позволяют определить сеть Петри, хотя в получающихся в результате структурах позиции и переходы меняются местами.

Двойственной к сети Петри $C = (P, T, I, O)$ является сеть Петри

$\bar{C} = (T, P, I, O)$, которая получается в результате перестановки позиций и переходов. Структура графа сохраняется, просто меняются местами кружки и планки. На рис. 7 показана сеть, двойственная к сети Петри на рис. 4.

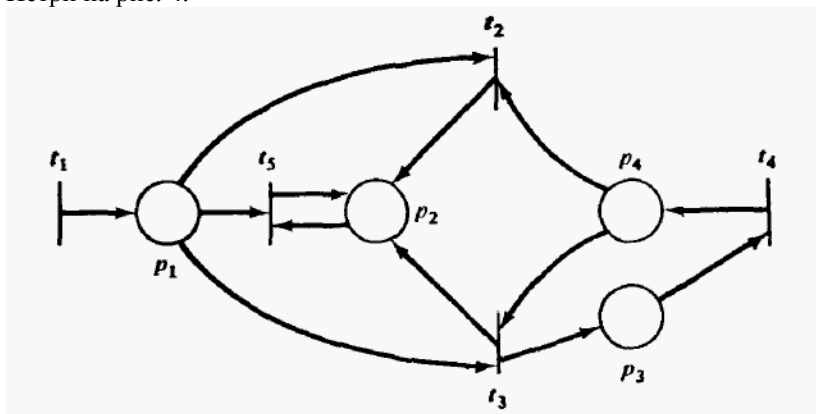


Рис. 7. Сеть, двойственная к сети Петри, показанной на рис. 4.

Двойственность — обычно полезный прием в теории графов и кажется интересным понятием для сетей Петри. Однако никакой пользы извлечь из понятия двойственной сети Петри в исследовании этих сетей не представляется возможным. Это объясняется в основном трудностью определения сети, двойственной к маркированной сети Петри. Маркированные сети Петри мы обсудим позднее.

Графы сети Петри являются мультиграфами, так как позиция может быть кратным входом или выходом перехода. В графе это показывается несколькими дугами между позицией и переходом. В то время как такой способ удовлетворителен для дуг с малой кратностью (не более трех), он неудобен для дуг очень большой кратности. Таким образом, в качестве альтернативного представления структур с большой кратностью используется *пучок* дуг. Пучок — это специальная дуга, которая рисуется жирной линией и помечается кратностью.

Рис. 8 иллюстрирует переход с входной кратностью 7 и выходной кратностью 11.

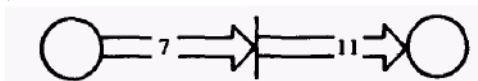


Рис. 8. Пучок дуг. Для графов с большой кратностью используется пучок дуг, помеченный числом кратности, а не изображение всех кратных дуг.

2.1.5. Маркировка сетей Петри

Маркировка μ , есть присвоение фишек позициям сети Петри. Фишка — это примитивное понятие сетей Петри (подобно позициям и переходам). Фишки присваиваются (можно считать, что они принадлежат) позициям. Количество и положение фишек при выполнении сети Петри могут изменяться. Фишки используются для определения выполнения сети Петри.

Определение 5. *Маркировка* μ сети Петри $S = (P, T, I, O)$ есть функция, отображающая множество позиций P в множество неотрицательных целых чисел N .

$$\mu : P \rightarrow N.$$

Маркировка μ может быть также определена как n -вектор $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, где $n = |P|$ и каждое $\mu_i \in N$, $i = 1, \dots, n$. Вектор μ определяет для каждой позиции p_i сети Петри количество фишек в этой позиции. Количество фишек в позиции p_i есть μ_i , $i = 1, \dots, n$. Связь между

Кононюк А.Е. Графы

определениями маркировки как функции и как вектора очевидным образом устанавливается соотношением $\mu(p_i)=\mu_i$. Обозначение ее в виде функции является несколько более общим и поэтому употребляется гораздо чаще.

Маркированная сеть Петри $M = (C, \mu)$ есть совокупность структуры сети Петри $C = (P, T, I, O)$ и маркировки μ , и может быть записана в виде $M = (P, T, I, O, \mu)$.

На графе сети Петри фишки изображаются маленькой точкой в кружке, который представляет позицию сети Петри. На рис. 9 и 10 приведены примеры графического представления маркированной сети Петри.

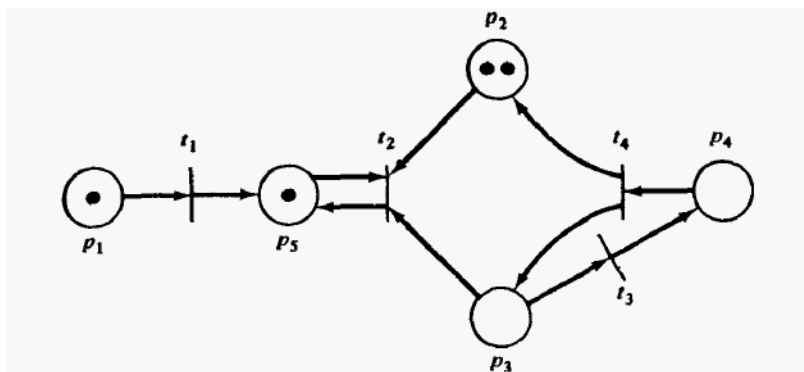


Рис. 9. Маркированная сеть Петри. Структура сети Петри совпадает со структурами на рис. 1 и 4. Маркировка — (1, 2, 0, 0, 1).

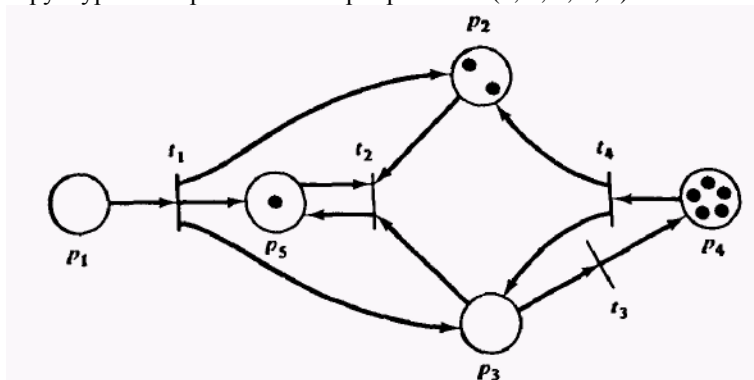


Рис. 10. Маркированная сеть Петри. Структура аналогична структуре, изображенной на рис. 9, но маркировка отличается.

Так как количество фишек, которое может быть определено для каждой позиции, неограниченно, то в целом для сети Петри существует бесконечно много маркировок.

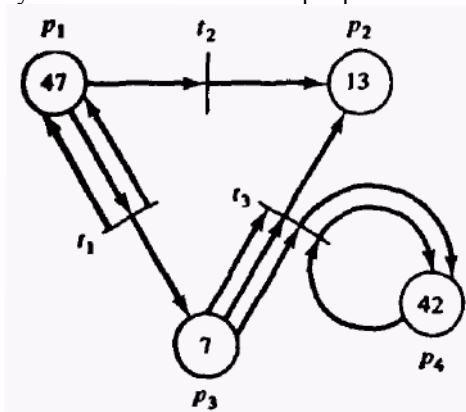


Рис. 11. Граф сети Петри с очень большой маркировкой (47, 13, 7, 42).

Множество всех маркировок сети Петри, обладающей n позициями, есть множество всех n -векторов, N^n . Это множество, хотя и бесконечно, является счетным.

2.1.6. Правила выполнения сетей Петри

Выполнением сети Петри управляют количество и распределение фишек в сети. Фишки находятся в кружках и управляют выполнением переходов сети. Сеть Петри выполняется посредством *запусков* переходов. Переход запускается удалением фишек из его входных позиций и образованием новых фишек, помещаемых в его выходные позиции.

Переход может запускаться только в том случае, когда он *разрешен*. Переход называется разрешенным, если каждая из его входных позиций имеет число фишек по крайней мере равное числу дуг из позиции в переход. Кратные фишки необходимы для кратных входных дуг. Фишки во входной позиции, которые разрешают переход, называются его *разрешающими фишками*. Например, если позиции p_1 и p_2 служат входами для перехода t_4 , тогда t_4 разрешен, если p_1 и p_2 имеют хотя бы по одной фишке. Для перехода t_7 с входным комплектом $\{p_6, p_6, p_6\}$ позиция p_6 должна обладать по крайней мере тремя фишками, для того чтобы t_7 был разрешен.

Кононюк А.Е. Графы

Определение 6. Переход $t_j \in T$ в маркированной сети Петри $C = (P, T, I, O)$ с маркировкой μ разрешен, если для всех $p_i \in P$

$$\mu(p_i) \geq \#(p_i, I(t_j)).$$

Переход *запускается* удалением всех разрешающих фишек из его входных позиций и последующим помещением в каждую из его выходных позиций по одной фишке для каждой дуги. Кратные фишки создаются для кратных выходных дуг. Переход t_3 с $I(t_3) = \{p_2\}$ и $O(t_3) = \{p_7, p_{13}\}$ разрешен всякий раз, когда в p_2 будет хотя бы одна фишка. Переход t_3 запускается удалением одной фишки из позиции p_2 и помещением одной фишки в позицию p_7 и в p_{13} (его выходы). Дополнительные фишки в позиции p_2 не влияют на запуск t_3 (хотя они могут разрешать дополнительные запуски t_3). Переход t_2 , в котором $I(t_2) = \{p_{21}, p_{23}\}$ и $O(t_2) = \{p_{23}, p_{25}, p_{25}\}$, запускается удалением одной фишки из p_{21} и одной фишки из p_{23} , при этом одна фишка помещается в p_{23} и две в — p_{25} (так как p_{25} имеет кратность, равную двум).

Запуск перехода в целом заменяет маркировку μ сети Петри на новую маркировку μ' . Заметим также, что так как можно запустить только разрешенный переход, то при запуске перехода количество фишек в каждой позиции всегда остается неотрицательным. Запуск перехода никогда не удалит фишку, отсутствующую во входной позиции. Если какая-либо входная позиция перехода не обладает достаточным количеством фишек, то переход не разрешен и не может быть запущен.

Определение 7. Переход t_j в маркированной сети Петри с маркировкой μ может быть запущен всякий раз, когда он разрешен. В результате запуска разрешенного перехода t_j образуется новая маркировка μ' , определяемая следующим соотношением:

$$\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j)).$$

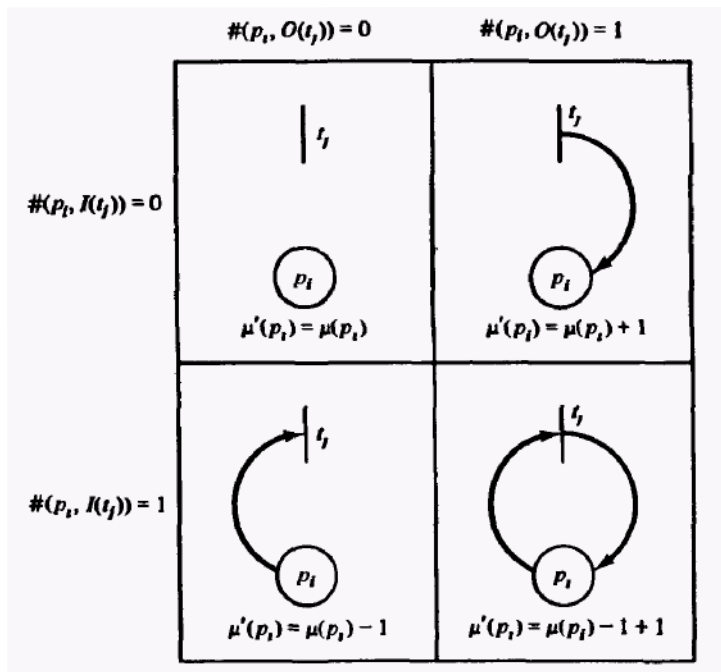


Рис. 12. Иллюстрация того, как меняется маркировка позиций, когда запущен переход t_j . Каждая позиция может или не может быть входом либо выходом перехода. Здесь показан случай для кратности 0 или 1.

В качестве примера рассмотрим маркированную сеть Петри, изображенную на рис. 13. При такой маркировке разрешены только три перехода: t_1 , t_3 и t_4 . Переход t_2 не разрешен, так как ни позиция p_2 , ни позиция p_3 , являющиеся входами перехода t_2 , не содержат ни одной фишки. Так как переходы t_1 , t_3 и t_4 разрешены, любой из них может быть запущен. Если запущен переход t_4 , то происходит удаление фишки из каждого входа и помещение фишки в каждый выход. При этом одна фишка удаляется из p_5 , одна фишка помещается в p_3 , а количество фишек в p_4 увеличивается с двух до трех. Новая маркировка, полученная в результате запуска перехода t_4 , показана на рис. 14.

В маркированной сети Петри, изображенной на рис. 14, разрешены только переходы t_1 и t_3 . При запуске перехода t_1 осуществляется

Кононюк А.Е. Графы

удаление фишки из p_1 и помещение фишек в p_2 , p_3 и p_4 (в p_4 — две фишки, так как эта позиция является кратным выходом перехода t_1).

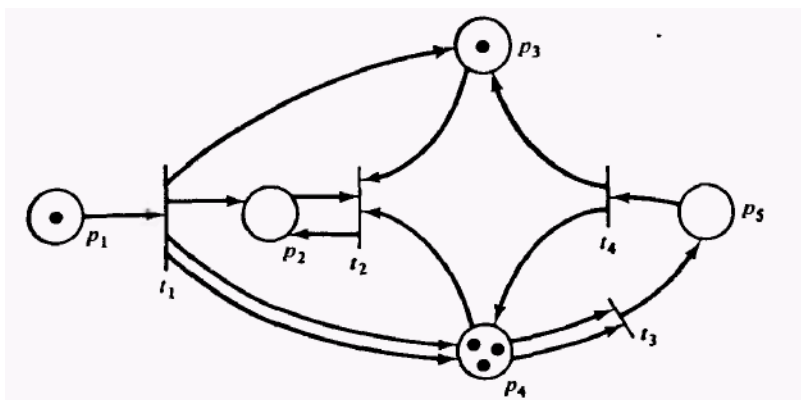


Рис. 2.14. Маркировка, полученная в результате запуска перехода t_4 в сети на рис. 13.

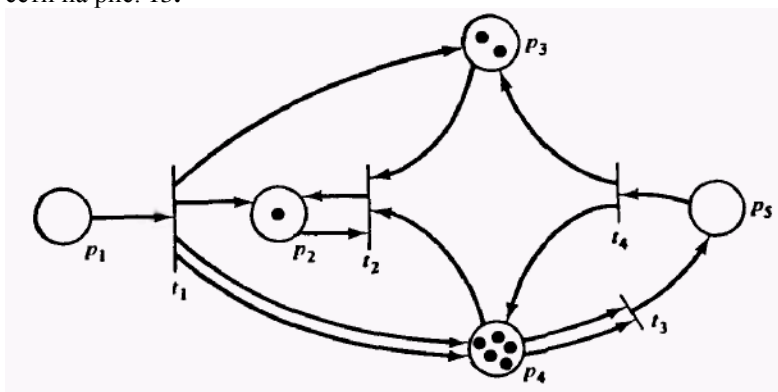


Рис. 15.. Маркировка, полученная при запуске перехода t_1 в сети на рис. 14.

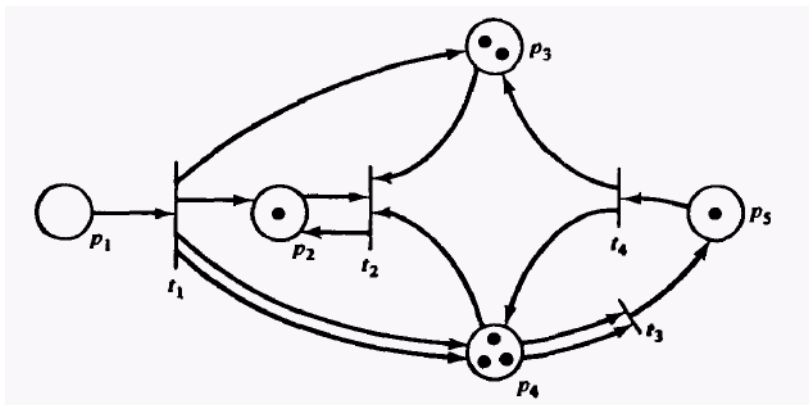


Рис. 16. Маркировка, полученная при запуске перехода t_3 в сети на рис. 15.

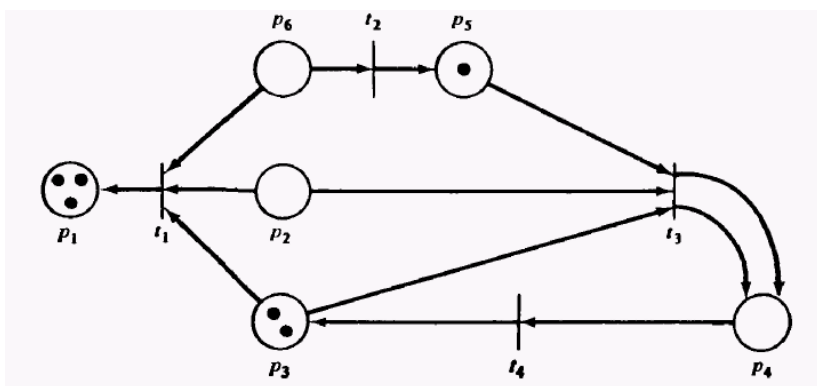


Рис. 17. Маркированная сеть Петри.

Эта операция образует маркировку, приведенную на рис. 15. В такой маркированной сети Петри переходы t_2 и t_3 разрешены. Запуск перехода t_3 образует новую маркировку (рис. 16), где две фишки удалены из p_4 , а одна добавлена в p_5 .

Запуски могут осуществляться до тех пор, пока существует хотя бы один разрешенный переход. Когда не останется ни одного разрешенного перехода, выполнение *прекращается*.

2.1.7. Пространство состояний сети Петри

Состояние сети Петри определяется ее маркировкой. Запуск перехода изменяет состояние сети Петри посредством изменения маркировки сети. Пространство состояний сети Петри, обладающей n позициями, есть множество всех маркировок, т. е. N^n . Изменение в состоянии, вызванное запуском перехода, определяется функцией изменения δ , которую мы назовем *функцией следующего состояния*. Когда эта функция применяется к маркировке p (состоянию) и переходу t_j она образует новую маркировку (состояние), которая получается при запуске перехода t_j в маркировке μ . Так как t_j может быть запущен только в том случае, когда он разрешен, то функция $\delta(\mu, t_j)$ не определена, если t_j не разрешен в маркировке μ . Если же t_j разрешен, то $\delta(\mu, t_j) = \mu'$, где μ' есть маркировка полученная в результате удаления фишек из входов t_j и добавления фишек в выходы t_j .

Определение 8. Функция следующего состояния $\delta: N^n \times T \rightarrow N^n$ для сети Петри $C = (P, T, I, O)$ с маркировкой μ и переходом $t_j \in T$ определена тогда и только тогда, когда $\mu(p_i) \rightarrow \#((p_i, I(t_j)))$ для всех $p_i \in P$. Если $\delta(\mu, t_j)$ определена, то $\delta(\mu, t_j) = \mu'$, где $\mu'(p_i) = \mu(p_i) - \#((p_i, I(t_j))) + \#((p_i, O(t_j)))$ для всех $p_i \in P$.

Пусть дана сеть Петри $C = (P, T, I, O)$ с начальной маркировкой μ^0 . Эта сеть может быть выполнена последовательными запусками переходов. Запуск разрешенного перехода t_j в начальной маркировке образует новую маркировку $\mu^1 = \delta(\mu^0, t_j)$. В этой новой маркировке можно запустить любой другой разрешенный переход, например t_k , образующий новую маркировку $\mu^2 = \delta(\mu^1, t_k)$. Этот процесс будет продолжаться до тех пор, пока в маркировке будет существовать хотя бы один разрешенный переход. Если же получена маркировка, в которой ни один переход не разрешен, то никакой переход не может быть запущен, функция следующего состояния не определена для всех переходов, и выполнение сети должно быть закончено.

При выполнении сети Петри получаются две последовательности: *последовательность маркировок* $(\mu^0, \mu^1, (\mu^2, \dots))$ и *последовательность переходов*, которые были запущены $(t_{j_0}, t_{j_1}, t_{j_2}, \dots)$. Эти две последовательности связаны следующим соотношением: $\delta(\mu^k, t_{j_k}) = \mu^{k+1}$ для $k = 0, 1, 2, \dots$. Имея последовательность переходов и μ^0 , легко получить последовательность маркировок сети Петри, а имея последовательность маркировок, легко получить последовательность переходов, за исключением нескольких вырожденных случаев. Таким образом, обе эти последовательности представляют описание выполнения сети Петри.

Кононюк А.Е. Графы

Пусть некоторый переход в маркировке μ разрешен и, следовательно, может быть запущен. Результат запуска перехода в маркировке μ есть новая маркировка μ' . Говорят, что μ' является *непосредственно достижимой* из маркировки μ , иными словами, состояние μ' непосредственно получается из состояния μ .

Определение 9. Для сети Петри $S = (P, T, I, O)$ с маркировкой μ маркировка μ' называется *непосредственно достижимой* из μ , если существует переход $t_j \in T$, такой, что $\delta((\mu, t_j) = \mu'$.

Можно распространить это понятие на определение множества Достижимых маркировок данной маркированной сети Петри. Если μ' непосредственно достижима из μ , а μ'' — из μ' , говорят, что μ'' *достижима* из μ . Определим *множество достижимости* $R(C, \mu)$ сети Петри S с маркировкой μ как множество всех маркировок, достижимых из μ . Маркировка μ' принадлежит $R(C, \mu)$, если существует какая-либо последовательность запусков переходов, изменяющих μ на μ' . Отношение «достижимости» (имеется в виду бинарное отношение на множестве маркировок, такое, что (μ, μ') принадлежит ему тогда и только тогда, когда $\mu' \in R(C, \mu)$) является рефлексивным, транзитивным замыканием отношения «непосредственной достижимости».

Определение 10. *Множество достижимости* $R(C, \mu)$ для сети Петри $S = (P, T, I, O)$ с маркировкой μ есть наименьшее множество маркировок, определенных следующим образом:

1. $\mu \in R(C, \mu)$;
2. Если $\mu' \in R(C, \mu)$ и $\mu'' = \delta(\mu', t_j)$ для некоторого $t_j \in T$, то $\mu'' \in R(C, \mu)$.

Для сети Петри, изображенной на рис. 18, и маркировки $\mu = (1, 0, 0)$ непосредственно достижимыми являются две маркировки: $(0, 1, 0)$ и $(1, 0, 1)$. Из $(0, 1, 0)$ нельзя достичь ни одной маркировки, так как ни один переход не разрешен. Из $(1, 0, 1)$ можно получить $(0, 1, 1)$ и $(1, 0, 2)$. Можно показать, что множество достижимости $R(C, \mu)$ имеет следующий вид: $\{(1, 0, n), (0, 1, n) | n \geq 0\}$.

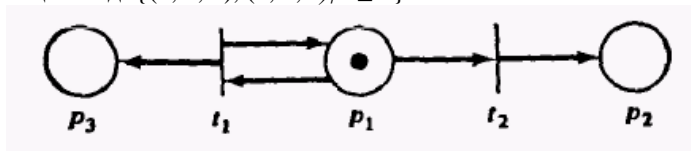


Рис. 18. Маркированная сеть Петри.

Удобно распространить функцию следующего состояния на отображение маркировки и последовательности переходов в новую маркировку. Для последовательности переходов $(t_{j1}, t_{j2}, \dots, t_{jk})$ и

Кононюк А.Е. Графы

маркировки μ маркировка $\mu' = \delta(\mu, t_{j1}, t_{j2}, \dots, t_{jk})$ есть результат запусков: сначала — t_{j1} затем — t_{j2} и т. д. до t_{jk} . (Такая операция возможна только в том случае, если каждый переход к моменту его запуска разрешен.)

Определение 11. Расширенная функция следующего состояния определяется для маркировки μ и последовательности переходов $\sigma \in T^*$ (T^* — множество всех подмножеств множества (булеан) переходов T) следующими соотношениями: $\delta(\mu, t_j, \sigma) = \delta(\delta(\mu, t_j), \sigma)$, $\delta(\mu, \lambda) = \mu$.

Обычно применяют эту расширенную функцию.

2.1.8. Альтернативные формы определения сетей Петри

Теория сетей Петри разрабатывалась рядом авторов. У них были различные мотивы и предпосылки. Вследствие такого разнообразия многие фундаментальные понятия были определены по-разному. Дадим некоторые из этих альтернативных определений, чтобы показать, что между ними нет существенных различий, и чтобы подготовить читателя к тем разнообразным представлениям, которые могут встретиться в литературе.

Например, оригинальные сети Петри не разрешают существование кратных дуг между позициями и переходами. Это эквивалентно определению входов и выходов переходов как множеств (а не как комплектов) позиций. Далее, правило запуска было ограничено следующим требованием: фишка есть в каждой входной позиции для перехода, и нет ни одной фишки в выходных позициях. Переход запускается путем удаления фишек из его входов (которые теперь становятся пустыми) и размещением фишек в (ранее пустых) выходах (которые теперь заполняются). Переход не может быть запущен, если фишка уже принадлежит выходной позиции. Таким образом, маркировка каждой позиции присваивает либо 0 фишек, либо 1 фишку, $\mu : P \rightarrow \{0, 1\}$. Теперь становится очевидным тот факт, что сеть с n позициями имеет точно 2^n возможных маркировок, т. е. конечное число состояний.

Ранние разработки Хольта по проекту теории информационных систем использовали эти же определения, но по мере продвижения исследований была обнаружена ограниченность такой модели. В работе Хольта и Коммонера, представленной на конференции в Вудс Холле, класс маркировок и правила запуска распространены на

Кононюк А.Е. Графы

произвольные маркировки, $\mu: P \rightarrow \{0, 1, 2, \dots\}$. Таким образом была создана базовая модель сетей Петри в том виде, в каком она определена сегодня (за исключением возможности кратных дуг).

Многие из первых исследователей не давали формального определения своих моделей, а описывали неформально относящиеся к их работе компоненты, такие, как позиции, переходы, фишки и правило запуска. Одно из первых формальных определений было дано Патилом в его докторской диссертации, в которой сеть Петри определялась в виде четверки (T, P, A, B) , где T — множество переходов, A — множество дуг, P — множество позиций и B — начальная маркировка. Дуги множества A соединяли либо позицию с переходом, либо переход с позицией. Таким образом, $A \subseteq (P \times T)$ и $(T \times P)$. Многие статьи по сетям Петри основываются на этом определении и определяют сеть Петри как тройку (P, T, A) с выделенной функцией маркировки.

Преобразование из формы (P, T, A) к определению с выделением входной и выходной функциями сравнительно просто. Множество дуг A разбивается на множество входных дуг $\{(p_i, t_j) \mid (p_i, t_j) \in A\}$ и выходных дуг $\{(t_j, p_i) \mid (p_i, t_j) \in A\}$. Эта форма непосредственно приводит к обобщению, допускающему кратные входы и выходы. Необходимо только указывать кратность для каждой входной и выходной дуги.

Хэк остановился на определении сетей Петри в виде четверки (P, T, F, B) , где, как обычно, P — множество позиций, T — множество переходов, F и B — функции, ставящие в соответствие позициям и переходам количество фишек, необходимых для входа (F) или образованных для выхода (B). Следовательно, переход t_j можно запустить только в том случае, если по крайней мере $F(t_j, p_i)$ фишек находятся в каждой позиции $p_i \in P$. Переход запускается путем удаления $F(t_j, p_i)$ фишек из каждой входной позиции и помещения $B(t_j, p_i)$ фишек в каждую выходную позицию. Функции F и B могут быть представлены матрицами.

Питерсон в своей диссертации попытался объединить переходы и их входы и выходы, определяя переход как упорядоченную пару комплектов позиций: $t_j \in P^{\alpha} \times P^{\beta}$. Первая компонента пары — комплект входов перехода; вторая — комплект выходов перехода. Это сводит множество примитивных понятий теории к позициям и фишкам, так как переходы являются структурами, составленными из позиций. Такой подход особенно полезен для простого определения переходов при построении сети Петри.

Эти определения отличаются от представленных здесь только разницей в обозначениях. В большинстве работ по сетям Петри именно это имеет место: различие в определениях проявляется только в

Кононюк А.Е. Графы

обозначениях. Однако в некоторых случаях определения могут ограничивать класс сетей Петри тем, что не допускаются кратные входные и выходные дуги или существует ограничение на форму переходов, т. е. требуется, чтобы переходы имели непустые множества входных и выходных позиций или входные и выходные позиции перехода не должны совпадать (без петель).

2.2. СЕТИ ПЕТРИ ДЛЯ МОДЕЛИРОВАНИЯ

Сети Петри были разработаны и используются в основном для моделирования. С их помощью могут быть промоделированы многие системы, и особенности системы с независимыми компонентами, например, аппаратное и программное обеспечение ЭВМ, физические системы, социальные и др. Сети Петри применяются для моделирования возникновения различных событий в системе. В частности, сети Петри могут моделировать поток информации или другие ресурсы системы.

В этом разделе мы приведем примеры систем, моделируемых при помощи сетей Петри. Эти примеры дадут представление о большом классе систем, которые можно моделировать сетями Петри, об используемом методе моделирования и о свойствах, которыми должны обладать моделируемые системы.

2.2.1. События и условия

Простое представление системы сетью Петри основано на двух основополагающих понятиях: *событиях* и *условиях*. События — это действия, имеющие место в системе. Возникновением событий управляет состояние системы. Состояние системы может быть описано множеством условий. **Условие** — **есть предикат или логическое описание состояния системы**. Условие может принимать либо значение «истина», либо значение «ложь».

Так как события являются действиями, то они могут происходить. Для того чтобы событие произошло, необходимо выполнение соответствующих условий. Эти условия называются *предусловиями* события. Возникновение события может вызвать нарушение предусловий и может привести к выполнению других условий, *постусловий*.

В качестве примера рассмотрим задачу моделирования простого автомата-продавца. Автомат-продавец находится в состоянии ожи-

Кононюк А.Е. Графы

дания до тех пор, пока не появится заказ, который он выполняет и посылает на доставку. Условиями для такой системы являются:

а) автомат-продавец ждет; б) заказ прибыл и ждет; в) автомат-продавец выполняет заказ; г) заказ выполнен. Событиями будут:

1. Заказ поступил. 2. Автомат-продавец начинает выполнение заказа.

3. Автомат-продавец заканчивает выполнение заказа. 4. Заказ посылается на доставку. Предусловия события 2 (автомат-продавец начинает выполнение заказа) очевидны: (а) автомат-продавец ждет; (б) заказ прибыл и ждет. Постусловие для события 2: (в) автомат-продавец выполняет заказ. Аналогично мы можем определить предусловия и постусловия для других событий и составить следующую таблицу событий и их пред- и постусловий:

Событие	Предусловия	Постусловия
1	нет	б
2	а, б	в
3	в	г, а
4	г	нет

Такое представление системы легко моделировать сетью Петри. В сети Петри условия моделируются позициями, события — переходами. При этом входы перехода являются предусловиями соответствующего события; выходы — постусловиями. Возникновение события равносильно запуску соответствующего перехода. Выполнение условия представляется фишкой в позиции, соответствующей этому условию. Запуск перехода удаляет разрешающие фишки, представляющие выполнение предусловий и образует новые фишки, которые представляют выполнение постусловий.

Сеть Петри на рис. 1 иллюстрирует модель приведенного выше автомата-продавца.

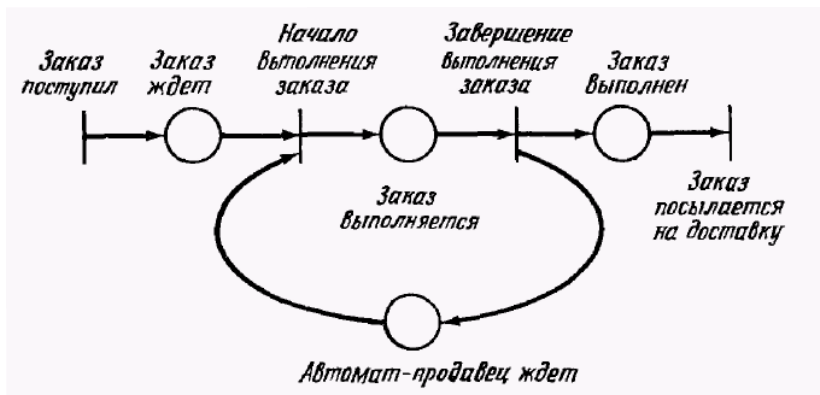


Рис. 1. Сеть Петри для простого автомата-продавца

Мы указали каждому переходу и позиции соответствующие событие и условие.

Можно моделировать и более сложную систему. Система автомат-продавец состоит из трех различных автоматов M_1 , M_2 и M_3 и двух операторов F_1 и F_2 . Оператор F_1 воздействует на автоматы M_1 и M_2 , а оператор F_2 — на M_1 и M_3 . Заказы требуют двух стадий обработки. Сначала они должны быть обработаны автоматом M_1 , затем либо автоматом M_2 , либо M_3 . Эта более сложная система будет иметь следующие условия:

- а) заказ прибыл и ждет обработки автоматом M_1 ;
- б) заказ обработан автоматом M_1 и ждет обработки либо автоматом M_2 , либо M_3 ;
- в) заказ выполнен;
- г) автомат M_1 незанят;
- д) автомат M_2 незанят;
- е) автомат M_3 незанят;
- ж) оператор F_1 незанят;
- з) оператор F_2 незанят;
- и) автомат M_1 находится под воздействием оператора F_1 ;
- к) автомат M_1 находится под воздействием оператора F_2 ;
- л) автомат M_2 находится под воздействием оператора F_1 ;
- м) автомат M_3 находится под воздействием оператора F_2 .

При этом могут происходить следующие события:

1. Поступление заказа.
2. Оператор F_1 начинает выполнение заказа на автомате M_1 .
3. Оператор F_1 закончил выполнение заказа на автомате M_1 .

Кононюк А.Е. Графы

4. Оператор F_2 начинает выполнение заказа на автомате M_1
5. Оператор F_2 закончил выполнение заказа на автомате M_1 .
6. Оператор F_1 начинает выполнение заказа на M_2 .
7. Оператор F_1 закончил выполнение заказа на M_2 .
8. Оператор F_2 , начинает выполнение заказа на M_3 .
9. Оператор F_2 закончил выполнение заказа на M_3 .

10. Заказ посылается на доставку.

Пред- и постусловия каждого события:

События	Предусловия	Постусловия
1	нет	а
2	а, ж, г	и
3	и	ж, г, б
4	а, з, г	к
5	к	б, з, г
6	б, ж, д	л
7	л	в, ж, д
8	б, е, з	м
9	м	в, е, з
10	в	нет

Сеть Петри этой системы показана на рис. 2.

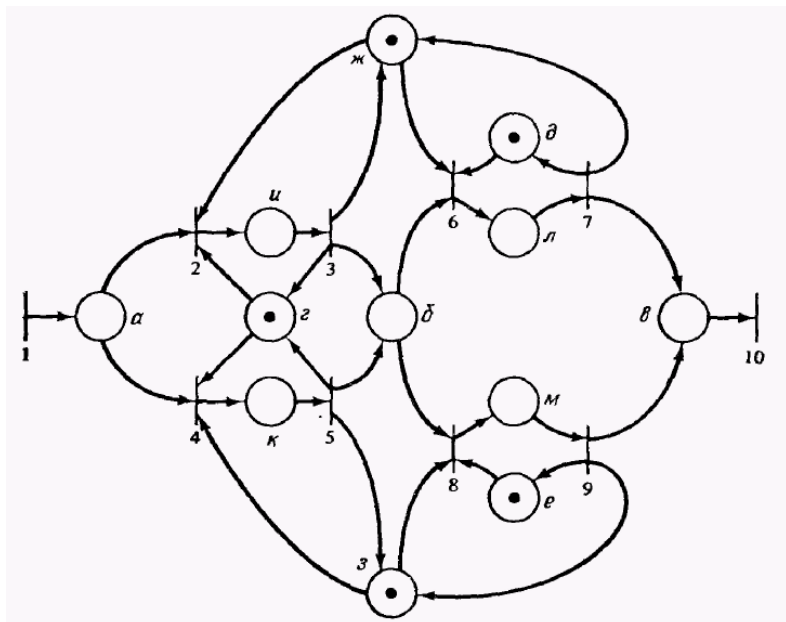


Рис. 2. Сеть Петри для сложного автомата-продавца.

Аналогичный пример можно привести для вычислительной системы, которая обрабатывает задания, поступающие с устройства ввода, и выводит результаты на устройство вывода. Задания посту--пают на устройство ввода. Когда процессор свободен и в устройстве ввода есть задание, процессор начинает обработку задания. Когда задание выполнено, оно посылается в устройство вывода; процессор же либо продолжает обрабатывать другое задание, если оно имеется, либо ждет прихода задания, если устройство ввода еще не получило такого. Эта система может быть промоделирована сетью Петри, показанной на рис. 3.

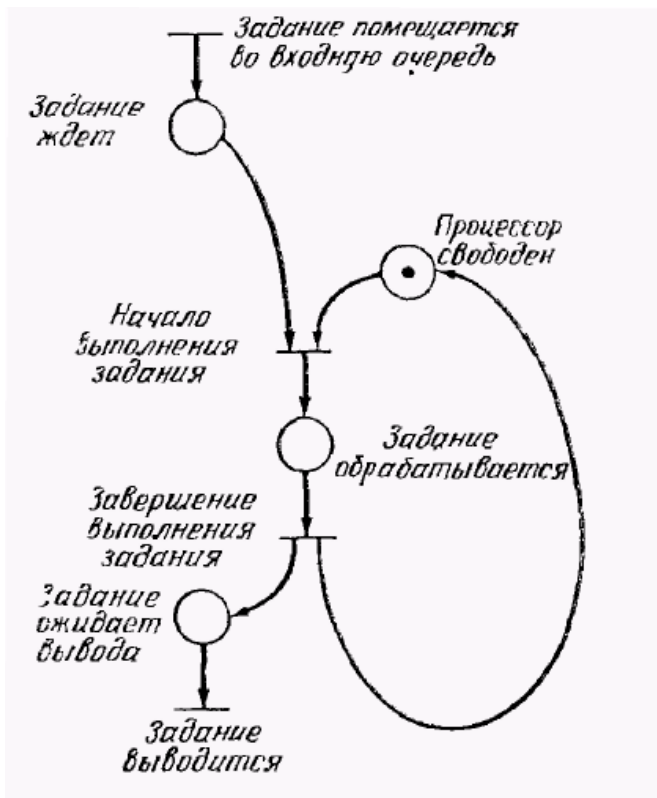


Рис. 3. Моделирование простой вычислительной системы.

2.2.2. Одновременность и конфликт

Приведенные примеры иллюстрируют некоторые особенности сетей Петри и систем, моделируемых с их помощью. Одной из особенностей является свойственный сетям и их моделям *параллелизм*, или *одновременность*. В модели сети Петри два разрешенных не взаимодействующих события могут происходить независимо друг от друга. Синхронизировать события, пока это не потребуется моделируемой системе, нет нужды. Но, когда синхронизация необходима, моделировать ее легко. Таким образом, сети Петри представляются

Кононюк А.Е. Графы

идеальными для моделирования систем с распределенным управлением, в которых несколько процессов выполняются одновременно.

Другая важная особенность сетей Петри — это их *асинхронная* природа. В сети Петри отсутствует измерение времени или течение времени. Это отражает философский подход к понятию времени, утверждающий, что одно из важнейших свойств времени, с логической точки зрения, — это определение частичного упорядочения событий. В реальной жизни различные события укладываются в различные интервалы времени, и это отражено в модели сети Петри независимостью от времени управления последовательностью событий. Структура сети Петри такова, что содержит в себе всю необходимую информацию для определения возможных последовательностей событий. Таким образом, на рис. 3 событие «завершение выполнения задания» должно следовать за соответствующим событием «начало выполнения задания». Однако нет и не требуется никакой информации, связанной с количеством времени, необходимым на выполнение задания. Выполнение сети Петри (или поведение моделируемой системы) рассматривается здесь как *последовательность* дискретных событий. Порядок появления событий является одним из возможных, допускаемых основной структурой. Это приводит к явной *недетерминированности* в выполнении сети Петри. Если в какой-то момент времени разрешено более одного перехода, то любой из нескольких возможных переходов может стать «следующим» запускаемым. Выбор запускаемого перехода осуществляется недетерминированным образом, т. е. случайно. Эта особенность сети Петри отражает тот факт, что в реальной жизненной ситуации, где несколько действий происходит одновременно, возникающий порядок появления событий — не однозначен; скорее может возникнуть любая из множества последовательностей событий. Однако частичный порядок появления события — единственен.

Проблемы, включающие в себя эти понятия, имеют философский характер. В своей точке зрения на Вселенную я, в частности, склонен к детерминизму: все действия предопределены состоянием Вселенной, и никакого беспорядка не существует. Кажущийся беспорядок — просто результат недостатка знаний о состоянии Вселенной и ее переходов из состояния в состояние. В этом смысле выбор для запуска одного из разрешенных переходов в моделируемой системе детерминирован, но не в модели, просто потому что модель не дает полной информации о системе.

Кононюк А.Е. Графы

Обратимся к теории относительности. Одним из ее основных положений является утверждение, что передача чего-либо не может быть мгновенной. Даже информация о возникновении события распространяется в пространстве со скоростью, ограниченной скоростью света c . Это означает, что если два события и произойдут одновременно (т. е. они не имеют причинной взаимосвязи), то порядок возникновения этих событий для двух разных наблюдателей может оказаться различным. Для двух событий A и B , происходящих в одно и то же время, наблюдатель, расположенный в непосредственной близости от события A , получит информацию, связанную с событием A , прежде, чем информацию, связанную с событием B . Таким образом, наблюдатель может сделать вывод, что событие A произошло до события B . С другой стороны, наблюдатель, расположенный в непосредственной близости от события B , отметит прямо противоположную последовательность возникновения событий.

Такое представление, хотя и является необходимым для полноты рассмотрения происходящего, однако влечет за собой значительные трудности при описании и анализе динамического поведения сети Петри, когда определяется последовательность запусков переходов. Для простоты обычно вводят следующее ограничение. Запуск перехода (и соответствующего события) рассматривается как *мгновенное* событие, занимающее нулевое время, и возникновение двух событий одновременно невозможно. Моделируемое таким образом событие называется *примитивным*; примитивные события мгновенны и *неодновременны*. (Иногда это обосновывается тем, что время — это непрерывная действительная переменная. Следовательно, если мы присвоим каждому событию время возникновения, то вероятность того, что любые две произвольно выбранные непрерывные действительные переменные совпадают, равна нулю, и, следовательно, события неодновременны.)

Непримитивными называются такие события, длительность которых отлична от нуля. Они не являются неодновременными и, следовательно, могут пересекаться во времени. Так как осуществление большинства событий в реальном мире занимает некоторое время, то они являются непримитивными событиями и поэтому не могут должным образом моделироваться переходами в сети Петри. Однако это не приводит к возникновению проблем при моделировании систем. Непримитивное событие может быть представлено в виде двух примитивных событий: «начало непримитивного события», «конец

Кононюк А.Е. Графы

непримитивного события» и условия «непримитивное событие происходит». Эта ситуация может быть промоделирована с помощью сети, показанной на рис. 4

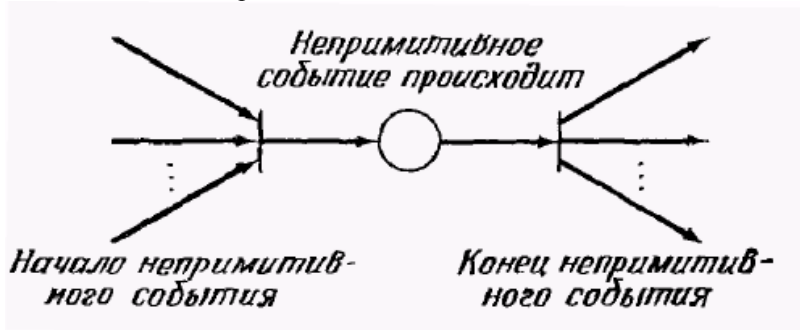


Рис. 4. Моделирование непримитивного события.

Петри и другие предложили представлять непримитивные события в сети Петри в виде прямоугольника, как показано на рис. 5, а примитивные события — планками, как мы делали это раньше.

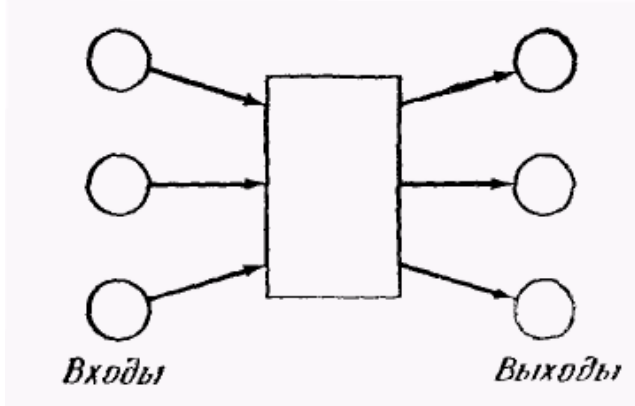


Рис. 5. Представление непримитивного события прямоугольником.

Это упростит некоторые сети Петри, например, такую, как на рис. 6, которая эквивалентна сети Петри, изображенной на рис. 3.

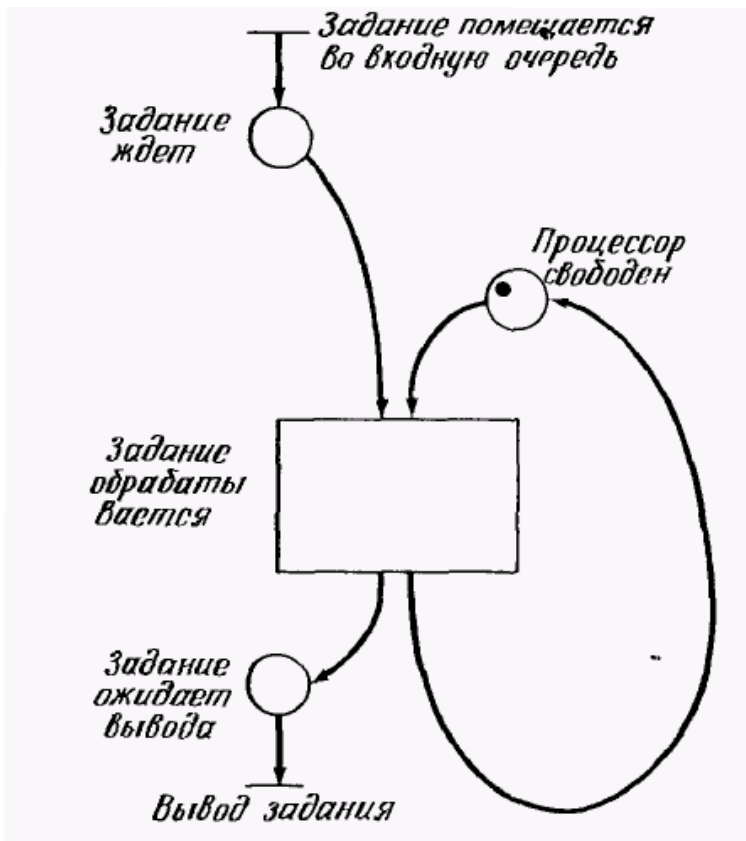


Рис. 6. Моделирование вычислительной системы с использованием непримитивного перехода.

Но так как в принципе предложенное понятие выражено в терминах более примитивных конструкций, мы не будем использовать обозначение в виде прямоугольника. Прямоугольник может иметь существенное значение при моделировании сложных систем на нескольких иерархических уровнях, так как он позволяет выделить в отдельный элемент сети целые подсети. Это в некотором смысле подобно понятиям подпрограммы или макроса в языках программирования.

Недетерминированность и неодновременность запусков переходов в моделировании параллельной системы показываются двумя способами. Один из них представлен на рис. 7.

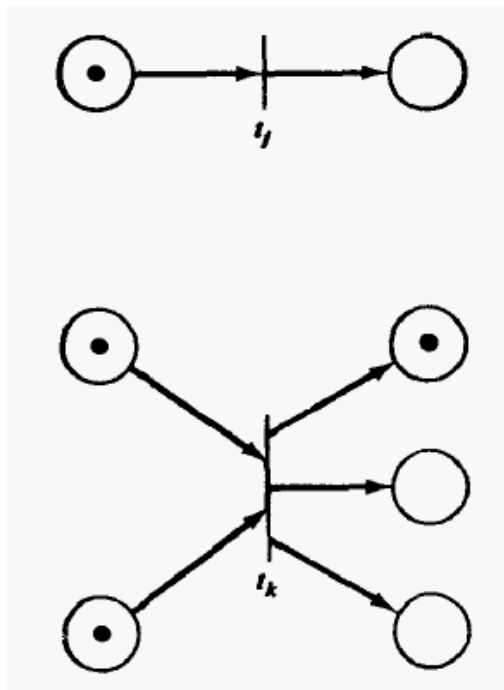


Рис. 7. Одновременность. Эти два перехода могут быть запущены в любом порядке.

В этой ситуации два разрешенных перехода в любом случае не влияют друг на друга, и в число возможных последовательностей событий входит последовательность, в которой первым срабатывает один переход, и последовательность, в которой первым будет запущен другой переход. Это называется *одновременностью*.

Другая ситуация, в которой одновременное выполнение затруднено и которая характеризуется невозможностью одновременного возникновения событий, показана на рис. 8.

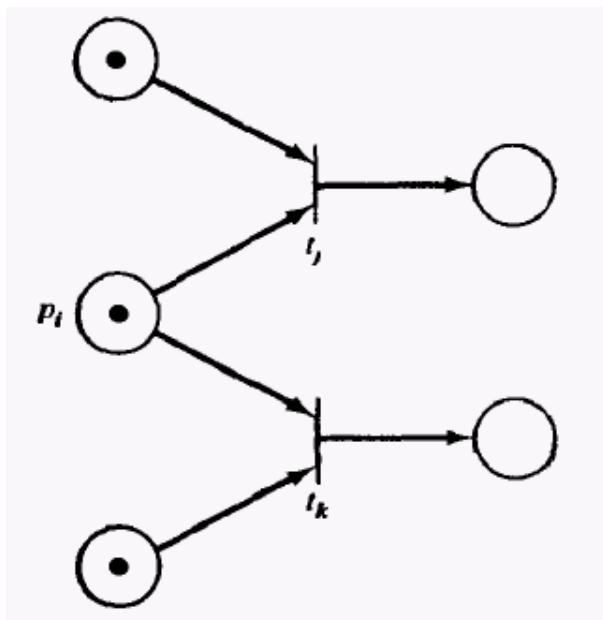


Рис. 8. Конфликт. Переходы t_j в t_k находятся в конфликте, т. е. запуск одного из них удаляет фишку из p_i и тем самым запрещает другой

Здесь два разрешенных перехода находятся в *конфликте*. Может быть запущен только один переход, так как при запуске он удаляет фишку из общего входа и запрещает другой переход.

Рассмотренные ситуации требуют внимательного изучения моделируемых сетями Петри систем для правильного отображения их поведения. К сожалению, многие работы по сетям Петри исследовали только свойства данной сети или класса сетей. Разработке же методов моделирования специально для сетей Петри внимания уделялось мало. Однако существуют определенные области, в которых сети Петри представляются идеальным инструментом для моделирования: это те области, в которых события происходят асинхронно и независимо. Для того чтобы дать представление о моделировании сетями Петри, в этой главе мы проиллюстрируем использование сетей Петри для моделирования аппаратного и программного обеспечения ЭВМ и других систем.

2.2.3. Аппаратное обеспечение ЭВМ

Аппаратное обеспечение ЭВМ можно рассматривать на нескольких уровнях, и сети Петри могут моделировать каждый из этих уровней. На одном уровне ЭВМ построены из простых устройств памяти и вентилях; на более высоком уровне в качестве основных компонент системы используются функциональные блоки и регистры. На еще более высоком уровне целые вычислительные системы могут быть компонентами сети ЭВМ. Одним из сильных свойств сетей Петри является их способность моделировать каждый из этих уровней. Обсудим эту способность и приведем некоторые примеры.

2.2.3.1. Конечные автоматы

На низком уровне вычислительные системы могут быть описаны *автоматами*. Автомат — это пятерка $(Q, \Sigma, \Delta, \delta, \Gamma)$, где

Q — конечное множество состояний $\{q_1, q_2, \dots, q_k\}$;

Σ — конечный входной алфавит;

Δ — конечный выходной алфавит;

$\delta: Q \times \Sigma \rightarrow Q$ — функция следующего состояния, отображающая текущее состояние и текущий вход в следующее состояние;

$\Gamma: Q \times \Sigma \rightarrow \Delta$ — функция выхода, отображающая текущее состояние и текущий вход в выходной символ.

Автоматы часто представляют в виде графов переходов, как, например, на рис. 9.

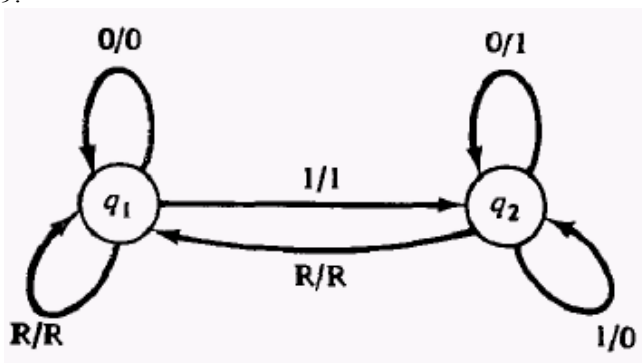


Рис. 9. Граф переходов для конечного автомата, вычисляющего дополнение до двух двоичного числа.

В графе переходов состояния представляются кружками, являющимися вершинами. Дуга из состояния q_i в q_j , помеченная a/b , означает, что, находясь в состоянии q_i , автомат при входе a перейдет в состояние q_j , выдавая при этом символ b . Формально следовало бы записать $\delta(q_i, a) = q_j$ и $\Gamma(q_i, a) = b$.

Входной алфавит определяет входы автомата из внешнего мира, а выходной алфавит — выходы автомата во внешний мир.

В качестве примера рассмотрим автомат, изображенный на рис. 9. Этот автомат преобразует двоичное число, представленное последовательностью бит, начиная с младшего, в дополнение до двух. В данном случае входной и выходной алфавит состоит из трех символов: 0, 1 и R. Автомат начинает работать в состоянии q_1 . Символ сброса (R) означает конец (или начало) числа и возвращает автомат в исходное состояние. Выход автомата для символа сброса является просто его повторением.

Аналогичный автомат показан на рис. 10.

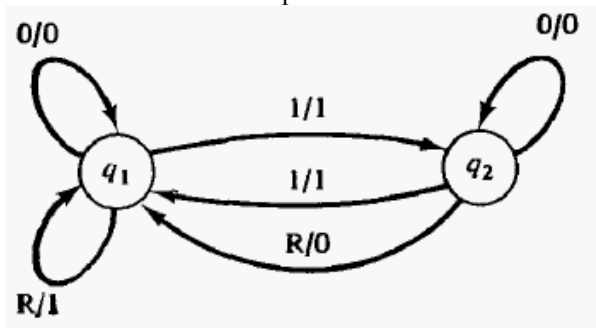


Рис. 10. Конечный автомат для определения четности входного двоичного числа.

При тех же самых входах этот автомат вычисляет четность входного числа. Он начинает работу в состоянии q_1 . Выход копирует вход до тех пор, пока входным символом не окажется символ сброса. Выходом для символа сброса будет 0 в случае нечетного числа и 1 — в случае четного числа.

При представлении конечного автомата сетью Петри следует обратить внимание на связь между сетью Петри и внешним миром, которая ранее не учитывалась. Моделирование взаимодействия с внешним миром можно реализовать многими способами. В нашей задаче мы моделируем это взаимодействие с помощью специального множества позиций. Позициями будут представлены каждый входной и выходной

Кононюк А.Е. Графы

символы. Мы допускаем, что из внешнего мира помещается фишка в позицию, соответствующую входному символу, а затем фишка, появившаяся в позиции, соответствующей выходному символу, удаляется оттуда. Эта последовательность действий будет повторяться необходимое число раз. Общая схема проиллюстрирована на рис. 11.

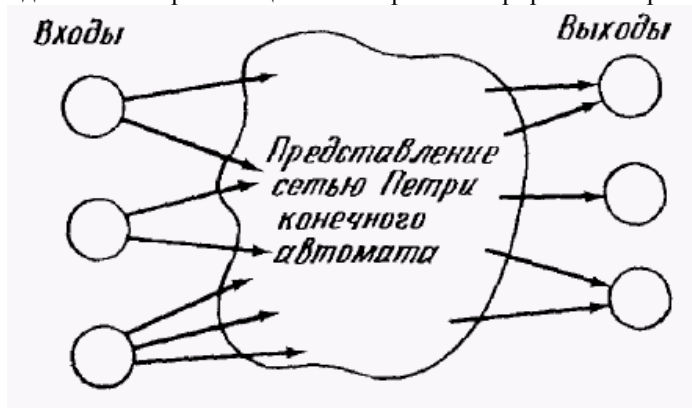


Рис. 11. Общий подход к моделированию связи между сетью Петри и внешним миром.

Заметим, что здесь не исключена возможность путаницы в понятиях, так как позиции, соответствующие входным и выходным символам, могут быть обоснованно названы входными и выходными позициями сети. Однако их не следует путать с входными или выходными позициями переходов. Несмотря на возможную путаницу, эти термины наиболее естественны для обоих понятий.

В качестве альтернативного подхода к моделированию входов и выходов сети могут быть использованы переходы. Для определения следующего входного символа из внешнего мира следует выбирать входной переход и запускать его. Сеть Петри ответит (в конце концов) запуском соответствующего перехода из множества выходных переходов, связанного с соответствующим выходом. Затем из внешнего мира будет запущен новый входной переход и т. д. Этот подход проиллюстрирован на рис. 12.

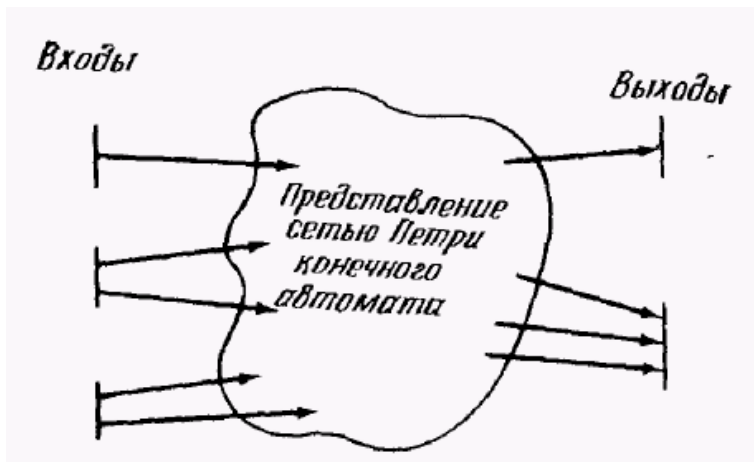


Рис. 12. Альтернативный подход представления связи между сетью Петри и внешним миром, где вместо позиций используются переходы.

Нетрудно показать, что оба этих подхода эквивалентны, поэтому будем использовать первый из них, в котором входные и выходные символы моделируются позициями.

Задав представление позиций, соответствующих символам входа и выхода, мы можем завершить построение модели системы конечных состояний. Представим каждое состояние автомата позицией в сети Петри. Текущее состояние отмечается фишкой; все остальные позиции пусты. Теперь для определения переходов из состояния в состояние можно ввести переходы сети следующим образом. Для каждой пары (состояние и входной символ) мы определяем переход, входными позициями которого являются позиции, соответствующие состоянию и входному символу, а выходными позициями — позиции, соответствующие следующему состоянию и выходу.

Для конечного автомата $(Q, \Sigma, \Delta, \delta, \Gamma)$ мы определили сеть Петри (P, T, I, O) таким образом:

$$\begin{aligned}
 P &= Q \cup \Sigma \cup \Delta, \\
 T &= \{t_{q, \sigma} \mid q \in Q \text{ и } \sigma \in \Sigma\}, \\
 I(t_{q, \sigma}) &= \{q, \sigma\}, \\
 O(t_{q, \sigma}) &= \{\delta(q, \sigma), \Gamma(q, \sigma)\}.
 \end{aligned}$$

Эта сеть Петри является моделью конечного автомата.

На рис. 13 изображена сеть Петри, соответствующая автомату с рис. 9.

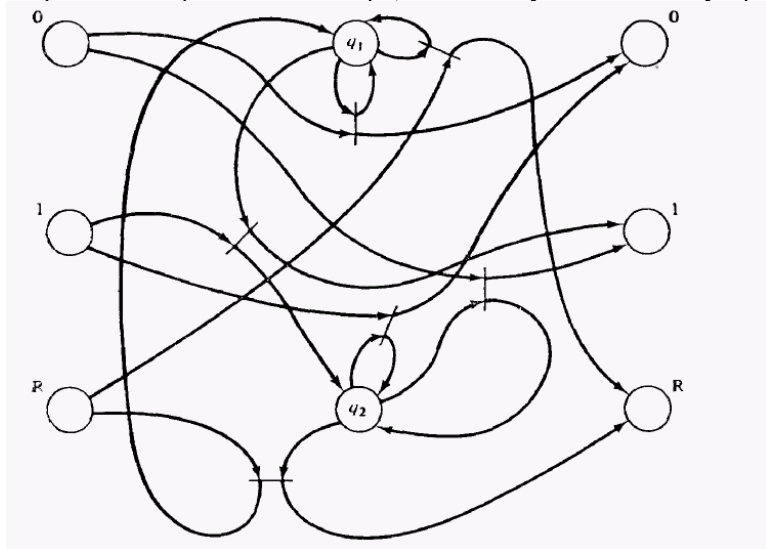


Рис. 13. Сеть Петри, эквивалентная автомату на рис. 3.9.

На рис. 14— сеть Петри, соответствующая автомату с рис. 10.

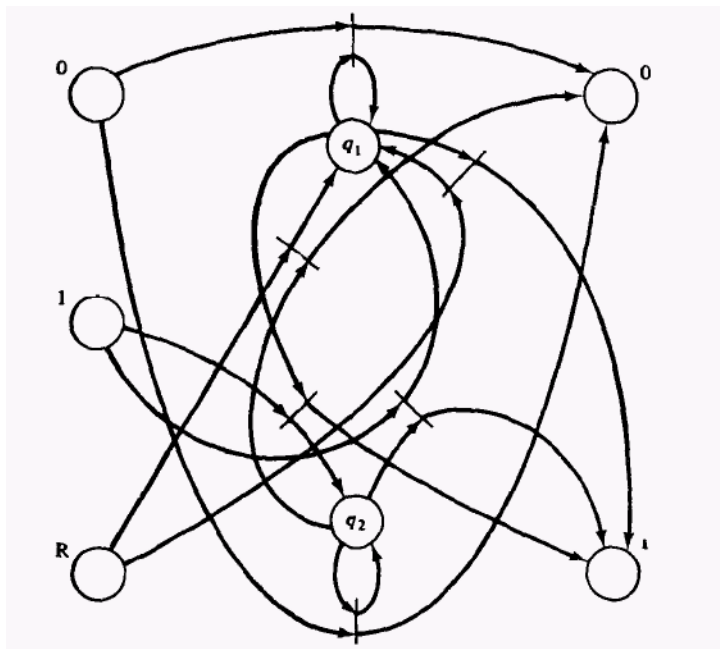


Рис. 14. Сеть Петри, эквивалентная автомату на рис. 10.

При сравнении сетей Петри на рис. 13, 14 с эквивалентными автоматами на рис. 9, 10 возникают некоторые вопросы. Прежде всего: почему модель сети Петри предпочтительнее описания конечным автоматом? Описание автоматом более понятно, чем описание сетью Петри, в которой 6 переходов, 24 дуги и 7 или 8 позиций. Это верно. Однако мы показали, что сети Петри могут представлять любую систему, представимую автоматом, и это свидетельствует о больших возможностях сетей Петри.

Кроме того, следует отметить, что модель сети Петри имеет определенное преимущество при композиции автоматов. Например, заметим, что выходной алфавит автомата с рис. 9 идентичен входному алфавиту автомата с рис. 10. Передавая выход автомата с рис. 9 на вход автомата с рис. 10, можно построить композицию автоматов, вычисляющую дополнение до двух и проверяющую четность. Такая композиция является автоматом с составными состояниями, компоненты которых — это состояния обоих подавтоматов. В сетях Петри такая композиция есть просто совмещение выходных позиций

первой сети с входными позициями второй. На рис. 15 показана композиция автоматов, а на рис. 16 — составная сеть Петри.

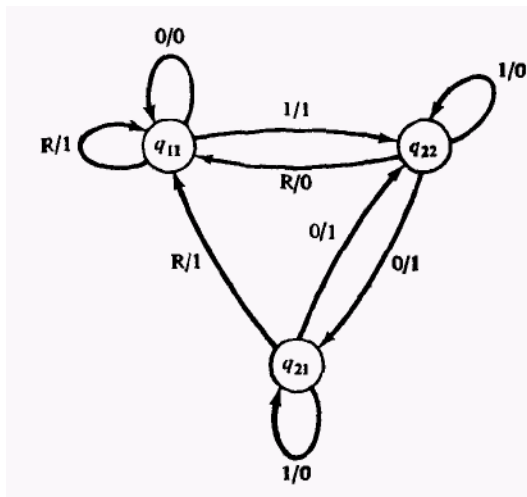


Рис. 15. Составной автомат, представляющий последовательную композицию автоматов, изображенных на рис. 9, 10.

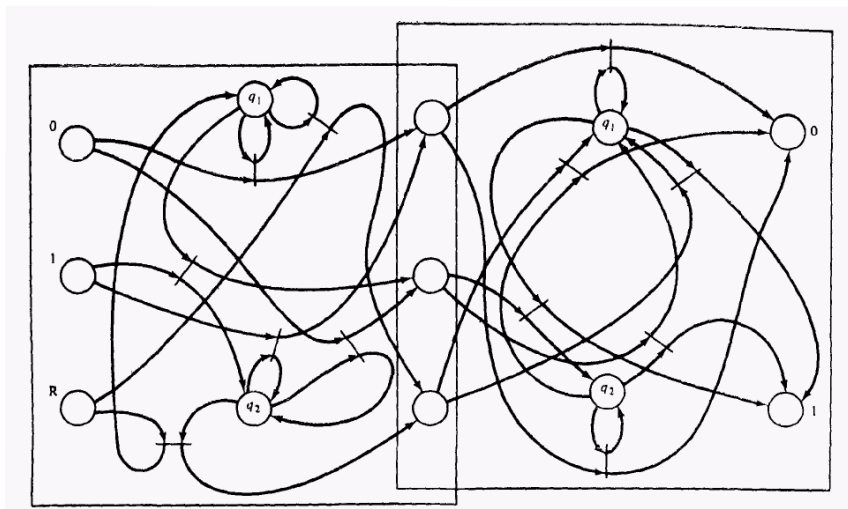


Рис. 16. Составная сеть Петри, являющаяся последовательной композицией сетей Петри на рис. 13, 14.

Другое преимущество представления сети Петри связано с иными формами композиции. Например, параллельная композиция позволяет компонентам *композиции автоматов* работать одновременно. В этом случае вновь получаем автомат с составными состояниями, в то время как для сети Петри — это просто дублирование фишек во входах, соответствующих входным символам, и использование их во всех компонентах сети Петри. Наконец, на выходе мы просто выбираем соответствующие позиции выхода. Например, если мы хотим соединить параллельно две сети Петри (рис. 13 и 14), то в результате получим сеть Петри, подобную изображенной на рис. 17, которая вычисляет дополнение числа до двух и его четность.

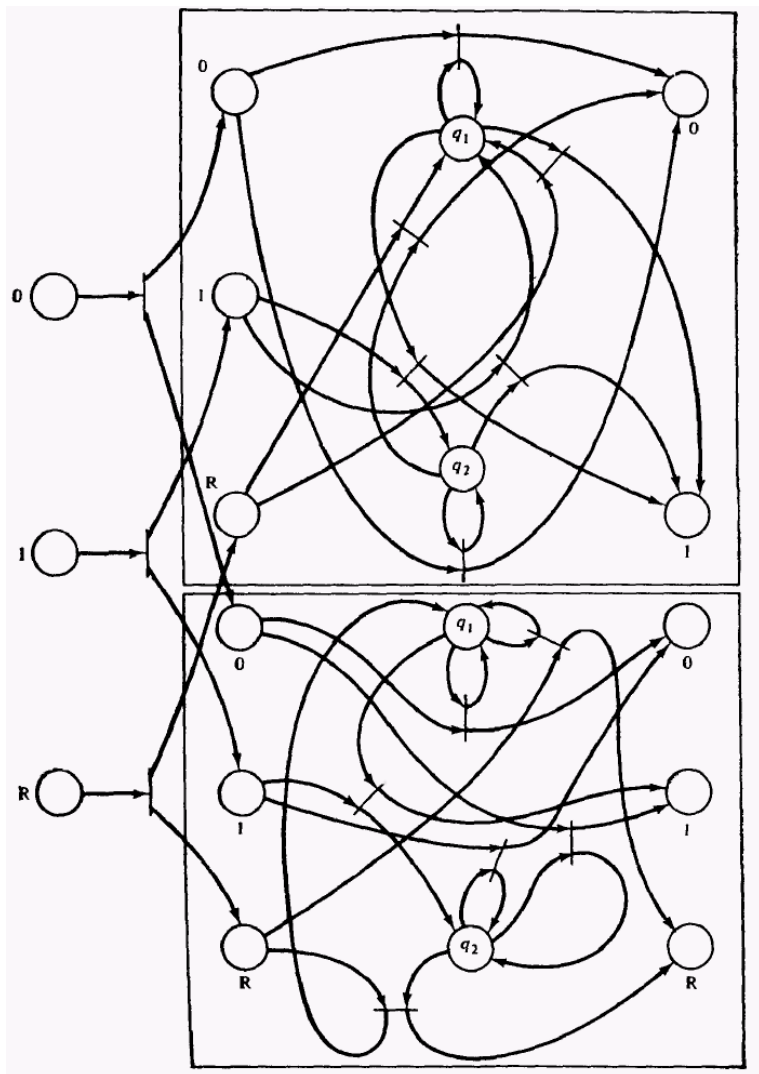


Рис. 17. Параллельная композиция сетей Петри с рис. 13, 14. Необходимо обеспечивать дублирование входов для обеих компонент с помощью специальной подсети.

Если на входе появляется символ сброса, то выходом является четность.

2.2.3.2. ЭВМ с конвейерной обработкой

Возможность моделировать параллелизм и довольно простого объединения подсистем, представленных сетями Петри, делают сети Петри весьма полезным инструментом моделирования сложной аппаратуры вычислительных систем. Вычислительные системы состоят из многих компонент. Это делает сеть Петри наиболее подходящим средством для их представления.

На протяжении ряда лет было предпринято много попыток увеличения производительности вычислительных систем путем параллельного выполнения нескольких функций. Примером такого подхода к построению высокопроизводительной ЭВМ является использование *конвейерной обработки*. Этот метод обработки подобен функционированию сборочного конвейера и особенно удобен для работы с векторами и массивами. Конвейер состоит из набора операций, которые могут выполняться одновременно. Когда операция k завершается, она передает свой результат операции $(k + 1)$ и ожидает от операции $(k - 1)$ нового задания. Если каждая операция занимает t единиц времени и всего существует n операций, то завершение обработки одного операнда потребует nt единиц времени. Однако, если на конвейерную обработку продолжают поступать новые операнды, результаты могут выдаваться со скоростью один каждые t единиц времени.

В качестве примера рассмотрим сложение двух чисел с плавающей точкой. Основные шаги этой операции предполагают:

1. Выделить экспоненты обоих чисел.
2. Сравнить экспоненты и изменить, если необходимо, должным образом порядок большей и меньшей экспонент.
3. Сдвинуть точку в числе с меньшей экспонентой для их уравнивания.
4. Сложить дроби.
5. Нормализовать результат.
6. Проверить экспоненту на переполнение и сформировать экспоненту и дробь результата.

Каждый из этих шагов может быть выполнен отдельным вычислительным блоком, где каждый отдельный операнд передается от блока к блоку для выполнения операции сложения. Это позволит выполнять до шести сложений одновременно.

Кононюк А.Е. Графы

Координацию различных блоков можно осуществлять несколькими способами. Обычно управление конвейерной обработкой является синхронным; время, отпущенное на выполнение каждого шага конвейера t , постоянно и фиксированно. Каждые t единиц времени результат каждого блока перемещается по конвейеру, чтобы стать входом для следующего блока. Однако при синхронном подходе обработка может быть приостановлена без необходимости, так как требуемое время может изменяться от блока к блоку, а также внутри данного блока для различных входов. Например, для выполнения шага нормализации результата при сложении чисел с плавающей точкой может потребоваться различное количество времени в зависимости от того, на сколько разрядов необходимо произвести нормализующий сдвиг и в какую сторону: вправо или влево. В этом случае, поскольку время t должно быть выбрано максимальным, необходимым для самого медленного блока конвейера, может оказаться так, что большую часть времени все единицы будут пребывать незагруженными, ожидая окончания t единиц времени.

В асинхронном конвейере в среднем процесс обработки может быть ускорен сигнализацией о завершении каждого шага конвейерной обработки и готовности передать свой операнд и получить новый. Результат шага k конвейерной обработки может быть послан на шаг $(k + 1)$, как только шаг k выполнен, а блок $(k + 1)$ свободен. Рассмотрим произвольный шаг конвейерной обработки. Очевидно, нужно иметь место, куда можно поместить входы и выходы в то время, как они используются или производятся. Обычно это предполагает наличие регистров: блок использует значение своего входного (буферного) регистра для вычисления значения выходного (буферного) регистра. После этого необходимо ждать, пока (1) — выходной регистр блока не будет очищен путем пересылки содержимого во входной регистр следующего блока и (2) — новое входное значение не появится в его входном регистре. Таким образом, для управления блоком k конвейера необходимо знать, когда выполняются следующие условия:

- входной регистр заполнен;
- входной регистр пуст;
- выходной регистр заполнен;
- выходной регистр пуст;
- блок занят;
- блок свободен;
- пересылка осуществлена.

Кононюк А.Е. Графы

На рис. 18 и 19 показано, как можно промоделировать асинхронный конвейер такого типа. На рис. 18 приведена блок-схема конвейера, моделируемого сетью Петри на рис. 19.

Отметим, что в этой модели мы промоделировали реальную работу блоков конвейера как непримитивных событий. Это позволяет нам игнорировать на этом уровне конкретные детали того, что делают блоки, и сосредоточиться на их правильном взаимодействии. Каждая операция также может быть промоделирована сетью Петри. Затем сети Петри для каждого блока можно внести в сеть Петри на рис. 19, получив более детальную сеть. Такая возможность моделирования системы на нескольких различных уровнях абстракции, т. е. иерархическим образом, может быть весьма полезна.

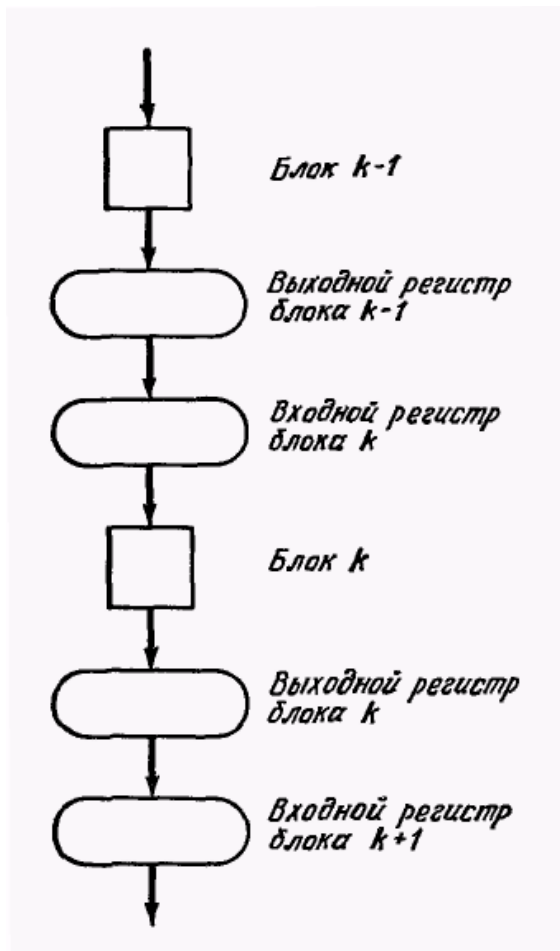


Рис. 18. Блок-схема устройства управления асинхронной ЭВМ с конвейерной обработкой.

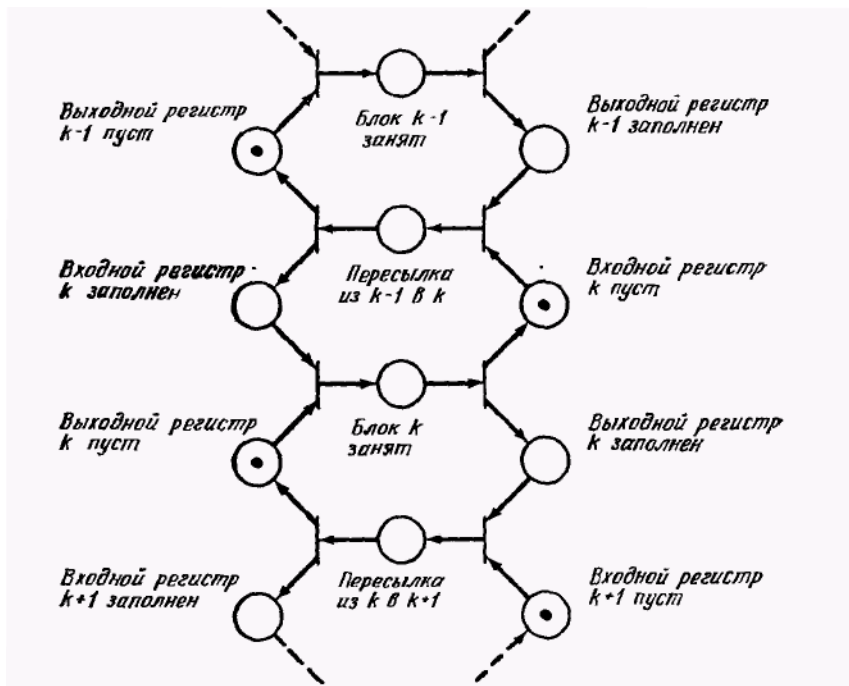


Рис. 19. Модель сети Петри устройства управления асинхронной ЭВМ с конвейерной обработкой.

2.2.3.3. Кратные функциональные блоки

Конвейерная структура управления, рассмотренная в предыдущем разделе, — это один из подходов к построению очень больших быстродействующих вычислительных систем. Другой подход использует *кратные функциональные блоки*. В ряде ЭВМ имеется 10 функциональных блоков: блок ветвления (для условных переходов); булев блок (для булевых операций); блок сдвига; блок сложения с плавающей точкой; блок сложения с фиксированной точкой; 2 блока умножения; блок деления и 2 блока приращения (для индексирования). Кроме того, имеется несколько регистров для хранения входных и выходных значений функциональных блоков. Устройство управления ЭВМ организует одновременное функционирование нескольких независимых блоков.

В качестве примера рассмотрим следующую последовательность, инструкций для вычислительной системы:

Кононюк А.Е. Графы

1. Умножить $X1$ на $X1$, результат поместить в $X0$.
2. Умножить $X3$ на $X1$, результат поместить в $X3$.
3. Сложить $X2$ с $X4$, результат поместить в $X4$.
4. Сложить $X0$ с $X3$, результат поместить в $X3$.
5. Разделить $X0$ на $X4$, результат поместить в $X6$.

Для выполнения инструкций устройство управления выдает первую инструкцию в блок умножения. Поскольку существуют два блока умножения, можно выдать также вторую инструкцию. При этом обоим блокам доступно содержимое $X1$. Инструкция 3 может быть выдана блоку сложения. Теперь, для того, чтобы выдать инструкцию 4, мы должны ждать, пока инструкции 1, 2 и 3 не будут завершены, так как инструкция 4 использует блок сложения (занятый выполнением инструкции 3), $X0$ (вычисляемый инструкцией 1) и $X3$ (вычисляемый инструкцией 2). Инструкция 5 должна ждать выполнения инструкции 1 (вычисление $X0$) и инструкции 3 (вычисление $X4$).

Организация параллелизма этого типа (выполнение нескольких инструкций программы одновременно) должна быть такой, чтобы результат выполнения программы с использованием параллелизма и без него был бы одинаков. Некоторые инструкции в программе требуют результата предыдущей инструкции до выполнения последующей. Система, которая организует параллелизм в последовательной программе таким способом для получения правильных результатов, обладает свойством *детерминированности*. Условия поддержания детерминированности были рассмотрены Бернстайном. Вот они: для двух операций a и b , таких, что a предшествует b на линейном участке программы, b может начать выполняться прежде, чем выполнена a , тогда и только тогда, когда входные данные b не требуют результата a и результаты b не изменяют ни входных данных, ни результатов a .

Один из методов применения таких ограничений для построения устройства управления, выдающего инструкции отдельным функциональным блокам, основан на *таблице резервирования*. Инструкция для функционального блока i , использующая регистры i , j и k , выдается только в том случае, если ни одна из четырех компонент не зарезервирована; при выдаче инструкции все эти компоненты становятся зарезервированными. Если же из-за того, что либо функциональный блок, либо один из регистров уже используются, инструкция не может быть выдана, то устройство управления перед переходом к следующей инструкции ожидает выполнения условий для выдачи этой инструкции.

Схему такого типа можно промоделировать сетью Петри. Каждому функциональному блоку и каждому регистру поставим в соответствие

Кононюк А.Е. Графы

позицию: если блок или регистр свободен — в позицию будет помещена фишка, если нет — фишки в позиции не будет. Кратные идентичные функциональные блоки показываются соответствующим числом фишек в позициях. На рис. 20 изображена часть сети Петри, которая может быть использована для моделирования выполнения инструкции, использующей блок u и регистры i, j и k . Моделирование всего устройства управления потребует гораздо большей сети Петри.

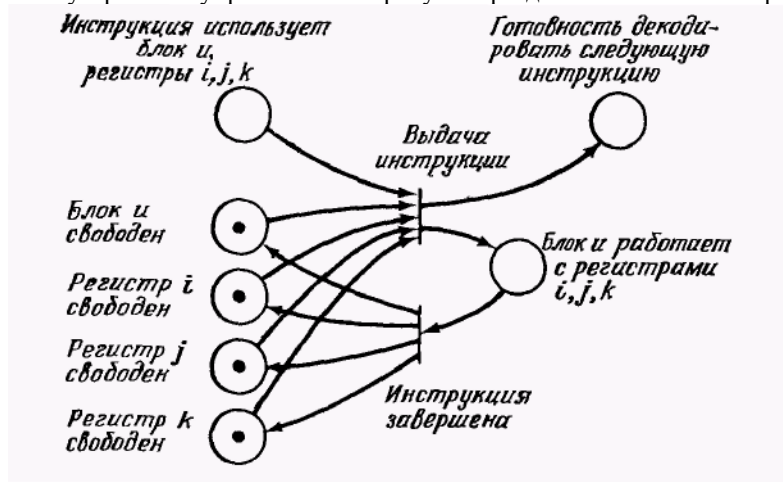


Рис. 20. Часть сети Петри, моделирующая устройство управления для ЭВМ, с кратными регистрами и кратными функциональными блоками.

Схема, описанная выше, представляет собой очень простой способ введения параллелизма и не рассматривает, например, тот факт, что кратные функциональные блоки могут использовать одновременно одинаковые входные регистры. Таким образом, максимальный параллелизм в этой схеме может быть не получен. Однако существуют другие схемы, с помощью которых это достигается. Такие более сложные схемы также моделируются (более сложными) сетями Петри. Эти сети могут быть очень велики. Если для каждой инструкции и тройки регистров требуется позиция, соответствующая условию «блок u оперирует регистрами i, j и k », то необходимо более полумиллиона позиций и переходов. Главная трудность заключается в представлении того факта, что содержимое внутренних регистров может определять, какие регистры и блоки будут использоваться (индексирование).

2.2.4. Программное обеспечение ЭВМ

В дополнение к аппаратному обеспечению ЭВМ сетями Петри можно моделировать и программное обеспечение. Чаще всего сети Петри используются именно для этого, и здесь они имеют наибольшие возможности для практического применения.

В этом разделе мы покажем, как сети Петри могут моделировать различные системы параллельных взаимодействующих процессов.

2.2.4.1. Блок-схемы

Вырожденным случаем параллельной системы обработки является система с одним процессом. Сначала рассмотрим, как сетью Петри может быть представлен отдельный процесс, а затем путем комбинации сетей Петри, представляющих несколько процессов, получим систему параллельных процессов.

Отдельный процесс описывается программой. Эта программа может быть написана на многих языках, но для удобства примем общецелевой язык, такой, как Алгол, Фортран, PL/1, Кобол, Паскаль, Бэйсик или даже язык ассемблера. Программа представляет два различных аспекта процесса: вычисление и управление. Вычисление связано с текущими арифметическими и логическими операциями, вводом и выводом, обычными манипуляциями над участками памяти и их содержимым. Управление же связано не со значениями или выполняемыми вычислениями, а только с порядком их выполнения.

Сети Петри удачно представляют структуру управления программ. Сети Петри предназначены для моделирования упорядочения инструкций и потока информации, но не для действительного вычисления самих значений. Модель системы по своей природе является *абстракцией* моделируемой системы. Поэтому она игнорирует все возможные специфические детали. Если бы моделировались все детали, тогда модель была бы дубликатом моделируемой системы, а не абстракцией.

Стандартный способ представления структуры управления программ — это *блок-схемы*. Блок-схема представляет поток управления в программе. Например, программа на рис. 21 представляется блок-схемой на рис. 22.

```
begin
  Input(  $y_1$ );
  Input(  $y_2$ );
   $y_3 := 1$ ;
  while  $y_1 > 0$ 
    do begin
      if odd (  $y_1$ )
        then begin
           $y_3 := y_3 * y_2$ ;
           $y_1 := y_1 - 1$ ;
        end;
       $y_2 := y_2 * y_2$ ;
       $y_1 := y_1 - 2$ ;
    end;
  Output (  $y_3$ );
end;
```

Рис. 21. Простая программа. Эта программа представлена блок-схемой на рис. 22 и сетью Петри на рис. 25.

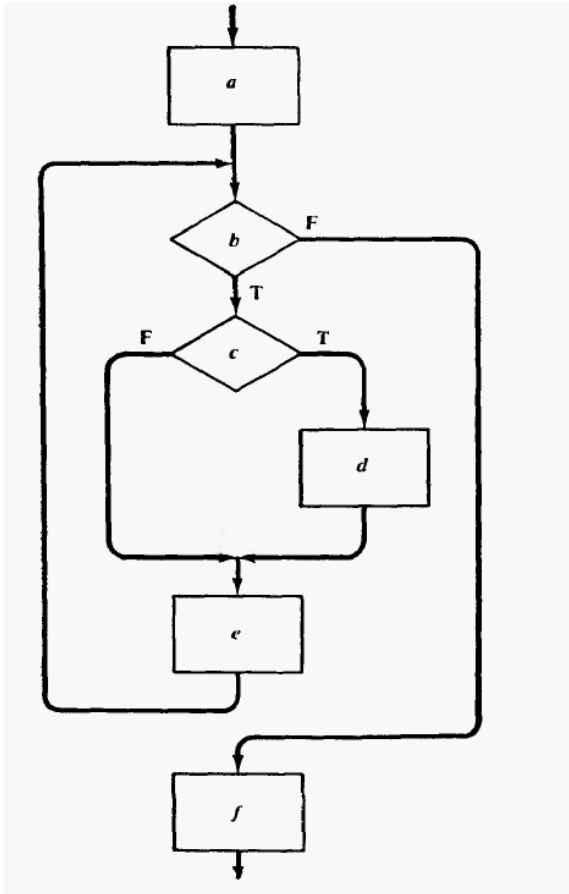


Рис. 22. Блок-схема программы, изображенной на рис. 21.

<i>Действие</i>	<i>Интерпретация</i>
<i>a</i>	$\text{Input}(y_1); \text{Input}(y_2); y_3 := 1;$
<i>b</i>	$y_1 > 0?$
<i>c</i>	$\text{odd}(y_1) ?$
<i>d</i>	$y_3 := y_3 * y_2; y_1 := y_1 - 1;$
<i>e</i>	$y_2 := y_2 * y_2; y_1 := y_1 / 2;$
<i>f</i>	$\text{Output}(y_3);$

Рис. 23. Интерпретация действий блок-схемы на рис. 22, представляющей программу с рис. 21.

Кононюк А.Е. Графы

Заметим, что блок-схема на рис. 22 не указывает конкретные вычисления, которые надо произвести, а только определяет структуру программы. Такая блок-схема является *абстрактной*. Рис. 23 показывает, как можно проинтерпретировать блок-схему (рис. 22) программы, представленной на рис. 21. Каждая последовательная программа может быть представлена в виде блок-схемы. Таким образом, показывая, как блок-схема может быть представлена сетью Петри, мы дадим представление сетью Петри программы.

Оказывается блок-схема во многом подобна сети Петри: блок-схема представима в виде узлов двух типов (принятия решения, показанные ромбами, и вычисления, показанные прямоугольниками) и дуг между ними. Удобный способ выполнения блок-схемы — введение фишки, которая представляет текущую инструкцию. По мере выполнения инструкций фишка передвигается по блок-схеме. Из сходства между графическими представлениями программы и сети Петри можно показать, что для получения эквивалентной сети Петри можно заменить узлы блок-схемы на позиции, а дуги — на переходы. Такой подход использовался для превращения конечного автомата в сеть Петри. Однако заметим, что в сети Петри действия моделируются переходами, тогда как в блок-схеме действия моделируются узлами. Кроме того, если движение нашей фишки текущей инструкции в блок-схеме нужно приостановить, это делается между узлами, на дугах, а не внутри блока. Таким образом, правильный перевод блок-схемы в сеть Петри заменяет узлы блок-схемы на переходы сети Петри, а дуги блок-схемы — на позиции сети Петри. Каждая дуга блок-схемы соответствует точно одной позиции в сети Петри. Узлы блок-схемы представляются по-разному в зависимости от типа узла: вычисления или принятия решения. На рис. 24 иллюстрируются оба способа перевода.

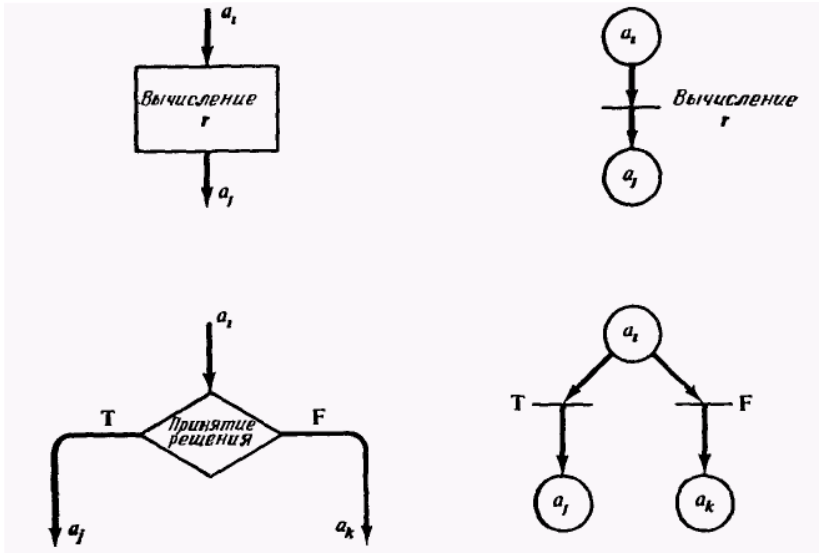


Рис. 24. Перевод узлов вычисления и принятия решения в блок-схеме в переходы в сети Петри.

На рис. 25 показан перевод блок-схемы на рис. 22 в эквивалентную сеть Петри.

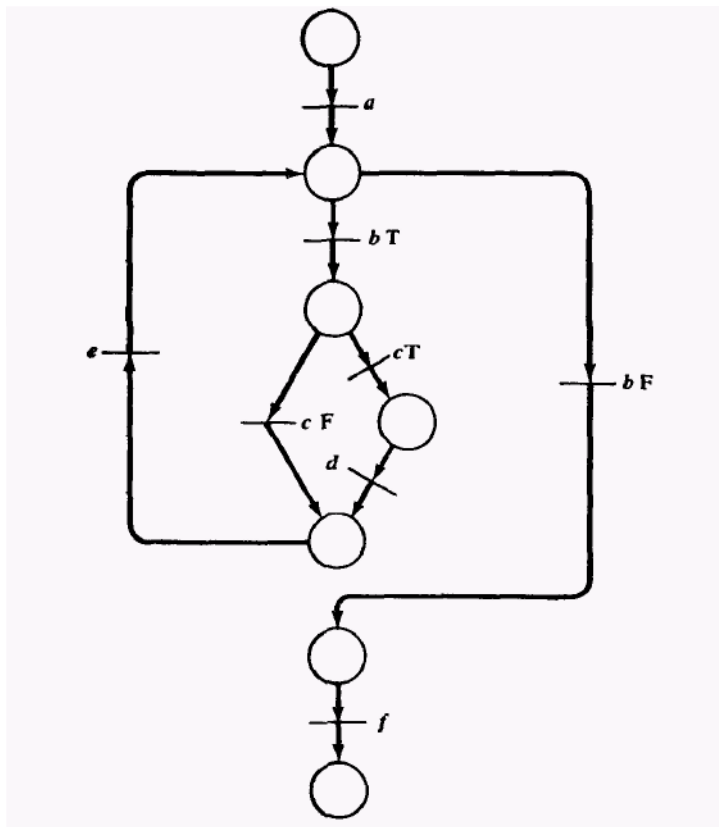


Рис. 25. Представление сетью Петри программы на рис. 21, полученное из блок-схемы на рис. 22.

Необходимо сделать замечания относительно того, что означают компоненты сети Петри на рис. 25. Что означают позиции? Для ответа рассмотрим интерпретацию фишек блок-схемы счетчиком команд. Фишка, находящаяся в позиции, означает, что счетчик команд установлен на готовность выполнения следующей инструкции. Каждая позиция имеет единственный выходной переход, за исключением позиции, которая предшествует принятию решения; такие позиции имеют по два выходных перехода, соответствующих истинному и ложному значению предиката.

Переходы, очевидно, связываются с действиями программы: вычислениями и принятия решений. Для интерпретации сети Петри необходимо интерпретировать каждый переход. Следует также

отметить, что переходы для вычислений имеют один вход и один выход; переход, представляющий вычисления, не может находиться в конфликте, поскольку его входная позиция не является входной для какого-либо другого перехода. Действие же, связанное с принятием решения, вводит в сеть конфликт. Выбор способа разрешения конфликта либо недетерминирован, либо им можно управлять извне (предсказателем, вычисляющим истинность или ложность предиката и вынуждающим запуск нужного перехода). Различие между этими двумя способами разрешения конфликта — методологический вопрос.

2.2.4.2. Параллелизм

Параллелизм (или одновременность) может быть введен несколькими способами. Рассмотрим случай двух параллельных процессов, каждый из которых может быть представлен сетью Петри. Таким образом, составная сеть Петри, являющаяся просто объединением сетей Петри для каждого из двух процессов, может представлять одновременное выполнение двух процессов. Начальная маркировка составной сети Петри имеет две фишки, по одной в каждой сети, представляя первоначальный счетчик команд процесса. Это вводит параллелизм, который, однако, не является еще полезным.

Обычный подход основан на введении параллелизма в процесс в вычислительной системе. Здесь имело место несколько предложений. Одно из них использует операции FORK и JOIN, впервые предложенные Деннисом и Ван Хорном. Операция FORK j , выполняемая в предложении i , приводит к одновременному выполнению текущего процесса, начиная с предложения $(i + 1)$, и вновь созданного процесса — с предложения j . Операция JOIN соединяет два процесса в один или, что равнозначно, уничтожает один из них. Эти операции моделируются в сети Петри, как показано на рис. 3.26.

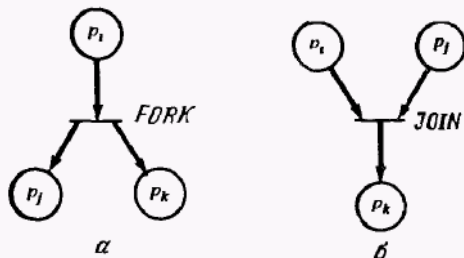


Рис. 26. Моделирование операций FORK и JOIN сетью Петри.

Кононюк А.Е. Графы

a — FORK (выполняемая на участке (создает два новых процесса на участки *i*, *j* и *k*), б — JOIN (соединяет два процесса, которые заканчиваются на участка *i* и *j*, в один процесс, продолжающийся на участке *k*)

Другое предложение по введению параллелизма основано на операциях *parbegin* и *parend*]. Структура управления была предложена Дейкстрой и имеет вид *parbegin S₁; S₂; ...; S_n parend*, где *S_i* — предложение. Смысл структуры *parbegin/parend* заключается в параллельном выполнении каждого из предложений *S₁, S₂,, S_n*. Эта конструкция может быть представлена сетью Петри, показанной на рис. 27.

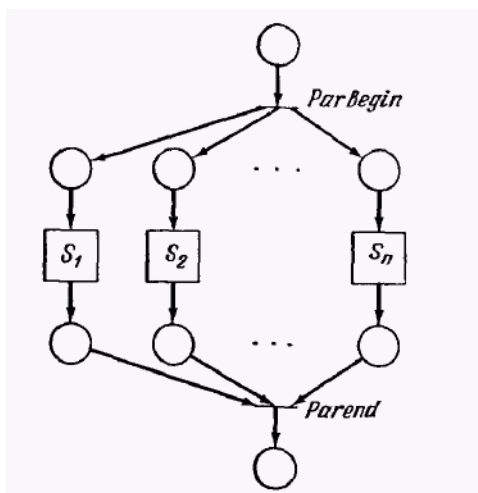


Рис. 27. Моделирование структуры *parbegin S₁; S₂; ...; S_n parend* в сети Петри. Каждый квадрат есть представление сетью Петри утверждений *S₁, S₂* и т. д. Здесь иллюстрируется также иерархическая природа моделирования сетями Петри.

2.2.4.3. Синхронизация

Введение параллелизма полезно только в том случае, когда компоненты процессов могут взаимодействовать при получении решения задачи. Такое взаимодействие требует распределения ресурсов между процессами. Для гарантии правильности работы системы в целом

Кононюк А.Е. Графы

распределением необходимо *управлять*. Проблемы синхронизации, возникающие при взаимодействии процессов, иллюстрируются многочисленными примерами, приведенными в литературе. Среди задач синхронизации: задача о взаимном исключении, производителе/потребителе, обедающих мудрецах, чтении/записи и др. Эти задачи стали классическими в области синхронизации; каждое новое предложение по механизму синхронизации должно решать их. И хотя сети Петри представляют собой схему моделирования, а не механизм синхронизации, они определенно способны моделировать механизмы синхронизации. Мы представляем здесь некоторые решения в виде сетей Петри.

2.2.4.4. Задача о взаимном исключении

Предположим, что несколько процессов разделяют общую переменную, запись, файл или другой элемент данных. Этот разделяемый элемент данных может использоваться процессами различными способами, упрощенно их можно классифицировать как *чтение* значения элемента данных или *запись* нового значения. Эти две операции являются часто единственными примитивными операциями. Это означает, что для обновления разделяемого элемента данных процесс должен сначала считать старое значение, затем вычислить новое и, наконец, записать его на то же место. Если два процесса в одно и то же время пытаются выполнить такую последовательность действий, то могут возникнуть трудности. Возможна следующая последовательность:

1. Первый процесс считывает значение x из разделяемого объекта;
2. Второй процесс считывает значение x из разделяемого объекта;
3. Первый процесс вычисляет новое значение $x' = f(x)$;
4. Второй процесс вычисляет новое значение $x'' = g(x)$;
5. Первый процесс записывает x' в разделяемый объект;
6. Второй процесс записывает x'' в разделяемый объект, уничтожая значение x' .

Результат вычисления первого процесса потерян, так как теперь значением разделяемого объекта является $g(x)$, в то время как им должно быть либо $g(f(x))$, либо $f(g(x))$. (Представьте себе, что $g(x)$ — «снять со счета x 1000 долл.», $f(x)$ — «поместить на счет x 1000 долл.», а процессы 1 и 2 — банковские операции.)

Для предотвращения проблем такого рода необходимо обеспечить механизм взаимного исключения. *Взаимное исключение* — это метод создания таких программ, что одновременно не более чем один процесс имеет доступ к разделяемому объекту данных. Участок кода, в

Кононюк А.Е. Графы

котором осуществляется доступ к разделяемому объекту и который требует защиты от вмешательства других процессов, называется *критической секцией*. Идея состоит в том, что когда процесс готов выполнить свою критическую секцию, он сначала ждет, пока другой процесс не выполнит свою собственную критическую секцию. Затем он «блокирует» доступ к критической секции, не давая возможности никакому другому процессу войти в свою критическую секцию. Он входит в критическую секцию, выполняет ее и, выйдя из нее, освобождает ее для доступа со стороны других процессов.

Эта задача может быть решена сетью Петри, как показано на рис. 28.

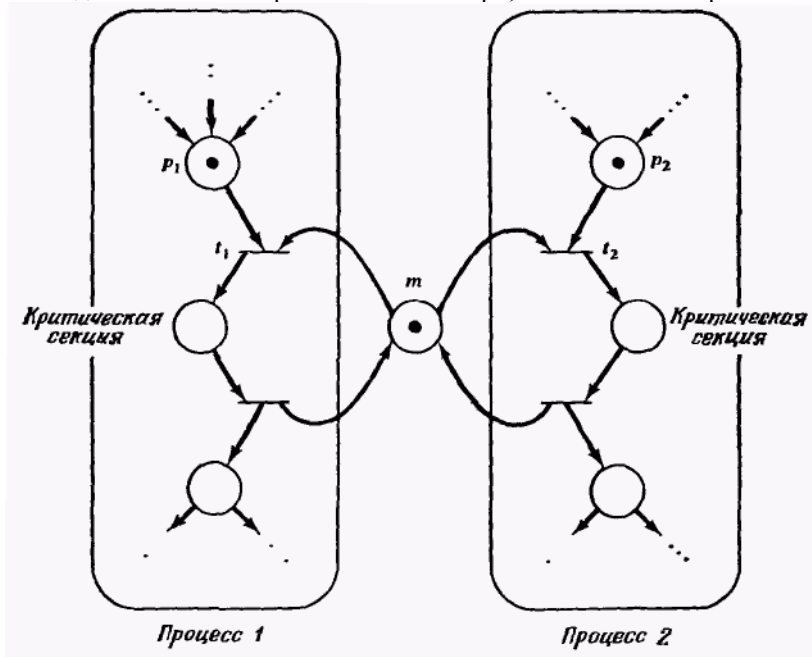


Рис. 28. Взаимное исключение. Управление доступом к критическим секциям двух процессов осуществляется таким образом, что оба процесса не могут одновременно выполнять свои критические секции.

Позиция m представляет собой разрешение для входа в критическую секцию. Для того чтобы какой-либо процесс вошел в критическую секцию, он должен иметь фишку в p_1 или в p_2 соответственно, свидетельствующую о желании попасть в критическую секцию, а также должна существовать фишка в m , дающая разрешение на вход. Если оба процесса пытаются войти в критическую секцию

одновременно, то переходы t_1 и t_2 вступят в конфликт, и только один из них сможет запуститься. Запуск t_1 запретит переход t_2 , вынуждая процесс 2 ждать, пока первый процесс выйдет из своей критической секции и возвратит фишку обратно в позицию m .

2.2.4.5. Задача о производителе/потребителе

В задаче о производителе/потребителе также присутствует совместно используемый объект, но в этом случае разделяемый объект точно определен и является буфером. Процесс-производитель создает объекты, которые помещаются в буфер. Потребитель ждет, пока объект не будет помещен в буфер, удаляет его оттуда и использует. Такая ситуация может быть промоделирована сетью Петри так, как показано на рис. 29. Позиция B представляет собой буфер, каждая фишка соответствует элементу данных, который произведен, но еще не использован.

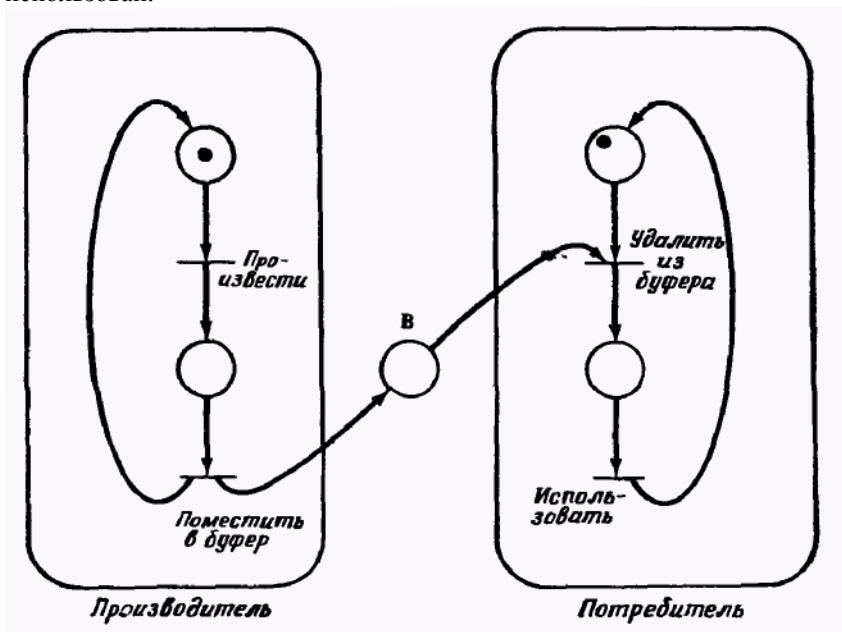


Рис. 29. Задача о производителе/потребителе, моделируемая сетью Петри.

Одни из вариантов этой задачи — это задача о нескольких производителях/нескольких потребителях. В этом случае несколько

Кононюк А.Е. Графы

производителей порождают элементы данных, помещаемые в общий буфер, для нескольких потребителей. На рис. 30 представлено решение этой задачи в виде сети Петри. Эта сеть совпадает с сетью на рис. 29, за исключением того, что для представления s производителей и t потребителей мы начали выполнение сети с s фишками в начальной позиции процесса-производителя и t фишками в начальной позиции процесса-потребителя.

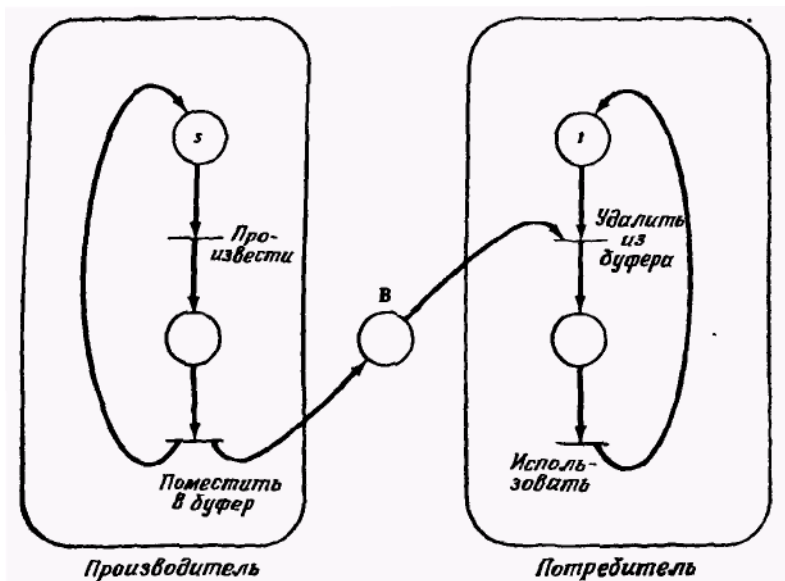


Рис. 30. Задача о нескольких производителях/нескольких потребителях (s производителей и t потребителей для фиксированных s и t).

Таким образом мы представляем s производителей и t потребителей, реализуемых реинтерабельными совместно используемыми программами. Альтернативой было бы дублирование программного кода для процессов производителя и потребителя, однако результатом этого при том же самом поведении была бы гораздо большая сеть.

В другом варианте задачи о производителе/потребителе используется буфер ограниченного размера. При такой постановке задачи буфер между производителем и потребителем ограничен, т. е. имеет только n ячеек для элементов данных. Следовательно, производитель не может постоянно работать с той скоростью, которая ему нужна, а вынужден

ждать, если потребитель работает медленно и буфер заполнен. На рис. 31 показано решение этой проблемы.

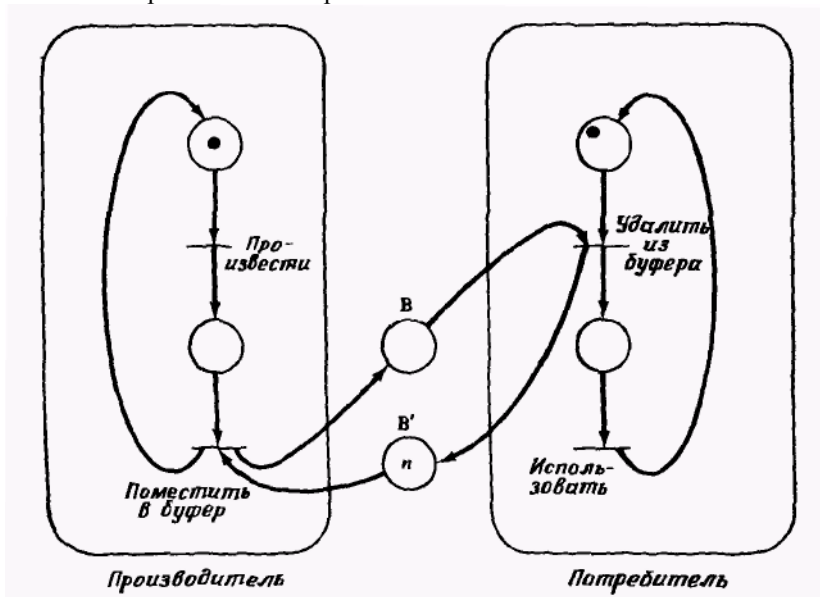


Рис. 31. Задача о производителе/потребителе с ограниченным буфером. Буфер, представленный позициями V и V' , ограничен n ячейками.

Ограниченному буферу сопоставляются две позиции: V представляет количество элементов данных, которые произведены, но еще не использованы (число заполненных ячеек), V' — количество пустых ячеек в буфере. Первоначально V' имеет n фишек, а V фишек не имеет. Если буфер заполнен, то V' фишек не имеет, а V имеет n фишек. Если теперь производитель попытается поместить еще один элемент данных в буфер, то он будет остановлен, так как в V' нет фишки, делающей этот переход разрешенным.

2.2.4.6. Задача об обедающих мудрецах

Задача об обедающих мудрецах была предложена Дейкстрой и связана с пятью мудрецами, которые попеременно то думали, то ели. Мудрецы сидят за большим круглым столом, на котором много блюд китайской кухни. Между соседями лежит одна палочка для еды. Однако для

приема китайской пищи необходимо две палочки, следовательно, каждый мудрец должен взять палочку слева и палочку справа: проблема заключается в том, что если все мудрецы возьмут палочки слева и затем будут ждать, когда освободятся палочки с правой стороны, то они будут ждать вечно и умрут от голода (*состояние тупика*).

На рис. 32 показано решение этой задачи с помощью сети Петри.

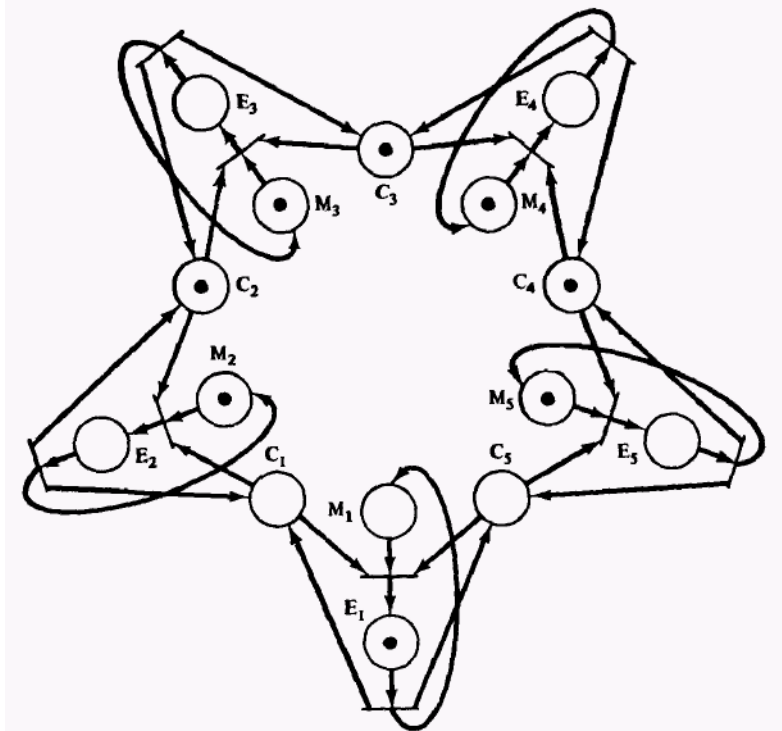


Рис. 32. Задача об обедающих мудрецах. Каждый мудрец представлен двумя позициями: размышление (M_i), принятие пищи (E_i).

Позиции C_1, \dots, C_5 представляют палочки для еды, и так как каждая из них первоначально свободна, то в начальной маркировке в каждой из этих позиций имеется фишка. Каждый мудрец представлен двумя позициями M_i и E_i для состояний размышления и принятия пищи соответственно. Для того чтобы мудрец перешел из состояния размышления в состояние принятия пищи, обе палочки (слева и справа) должны быть свободны. Это легко моделируется сетью Петри.

2.2.4.7. Задача о чтении/записи

Существует несколько вариантов задачи о чтении/записи, однако основная структура их остается неизменной. Имеются процессы двух типов: процессы чтения и процессы записи. Все процессы совместно используют общий файл или переменную или элемент данных. Процессы чтения не изменяют объект в отличие от процессов записи. Таким образом, процессы записи должны взаимно исключать все другие процессы чтения и записи, в то время как несколько процессов чтения могут иметь доступ к разделяемым данным одновременно. Задача состоит в определении структуры управления, которая не приведет к тупику и не допустит нарушения критерия взаимного исключения.

На рис. 33 иллюстрируется решение задачи в том случае, когда количество процессов чтения ограничено величиной n . Если в системе количество процессов чтения не ограничено, то только n процессов могут выполняться в одно и то же время.

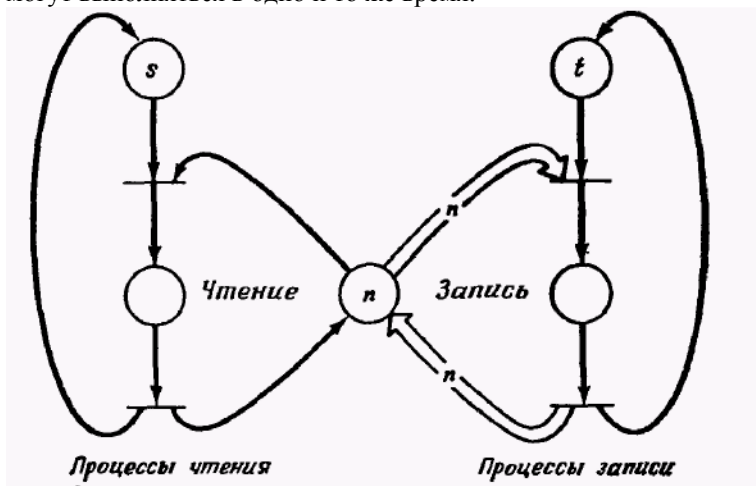


Рис. 33. Задача о чтении/записи в случае, когда число процессов чтения ограничено величиной n . Первоначально имеются s процессов чтения и t процессов записи.

Проблема возникает в том случае, если количество процессов чтения не ограничено и мы хотим предоставить возможность не-

Кононюк А.Е. Графы

ограниченному количеству процессов читать одновременно. В этом случае можно утверждать, что возникает необходимость хранения количества читающих процессов. При инициализации каждого процесса чтения в счетчик добавляется единица, а по окончании процесса единица вычитается. Это легко моделируется позицией, в которой количество фишек равно количеству процессов чтения. Однако, для того, чтобы предоставить процессу записи возможность приступить к записи, необходимо, чтобы счетчик был нулевым, т. е. соответствующая позиция была бы пустой. В сетях Петри нет механизма, который бы осуществлял проверку на нуль неограниченной позиции. Таким образом, оказывается, что задача о чтении/записи с неограниченным числом процессов чтения не может быть решена с помощью сетей Петри. Это первый случай, когда мы столкнулись с тем, что сети Петри не способны моделировать все системы. Этот вопрос заслуживает дальнейшего изучения.

2.2.4.8. P- и V-системы

Большинство задач синхронизации не могут быть решены непосредственно сетями Петри, но разрешимы скорее на основе известных механизмов синхронизации. В частности, одним из самых популярных механизмов синхронизации являются P- и V-операции над семафорами, впервые определенные Дейкстрой. Семафор — это элемент данных, который может принимать только неотрицательные целые значения. V-операция увеличивает значение на 1, а P-операция уменьшает его на 1. P-операцию можно применять только в том случае, когда значение семафора останется в результате неотрицательным; если же значение семафора равно 0, то P-операция должна ждать, пока какой-нибудь другой процесс не выполнит V-операцию. P- и V-операции определены как примитивные, т. е. никакая другая операция не может изменять значение семафора одновременно с ними. Такие операции легко моделируются сетью Петри, как показано на рис. 34.

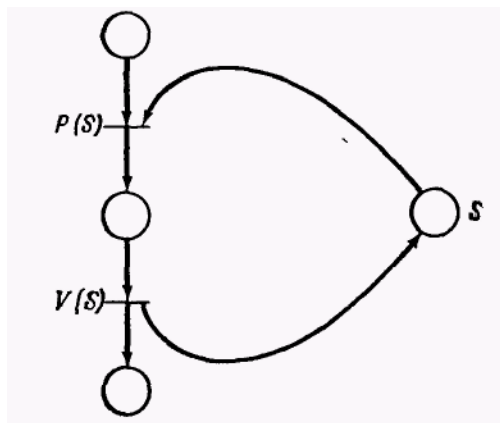


Рис. 34. Моделирование P- и V-операций над семафором S.

Каждый семафор моделируется позицией, количество фишек в позиции показывает значение семафора. P-операция использует позицию семафора в качестве входа, V-операция — в качестве выхода. Преимущество этого заключается в том, что многие системы проектируются и описываются с помощью P- и V-операций.

2.2.5. Другие системы

Описанные до сих пор системы относятся к самому очевидному типу систем, моделируемых сетями Петри: аппаратному и программному обеспечению ЭВМ. Но в большей части эта «очевидность» является результатом того, что сети Петри были определены и разработаны главным образом для этой цели. Но они также применимы для моделирования большого числа других систем, совершенно отличных от вычислительных систем. В этом разделе мы бегло ознакомимся с некоторыми из них.

PERT-диаграмма давно используется для планирования больших проектов. PERT-диаграмма является графическим представлением взаимосвязей между различными этапами, составляющими проект. Проект представляет собой совокупность большого числа этапов, при этом некоторые этапы должны завершиться прежде, чем начнут выполняться другие. Кроме того, на выполнение каждого этапа требуется определенное количество времени (иногда каждому этапу соответствует три времени, соответствующие худшему, среднему и лучшему случаям). Этапы графически представлены узлами; дуги

Кононюк А.Е. Графы

используются для отображения причинно-следственных связей между ними.

PERT-диаграмма и сеть Петри взаимосвязаны: PERT-диаграмма легко преобразуется в сеть Петри. Каждый этап PERT-диаграммы представляется позицией, причинно-следственные связи — переходами. Диаграмма на рис. 35 может быть преобразована в эквивалентную сеть Петри, изображенную на рис. 36.

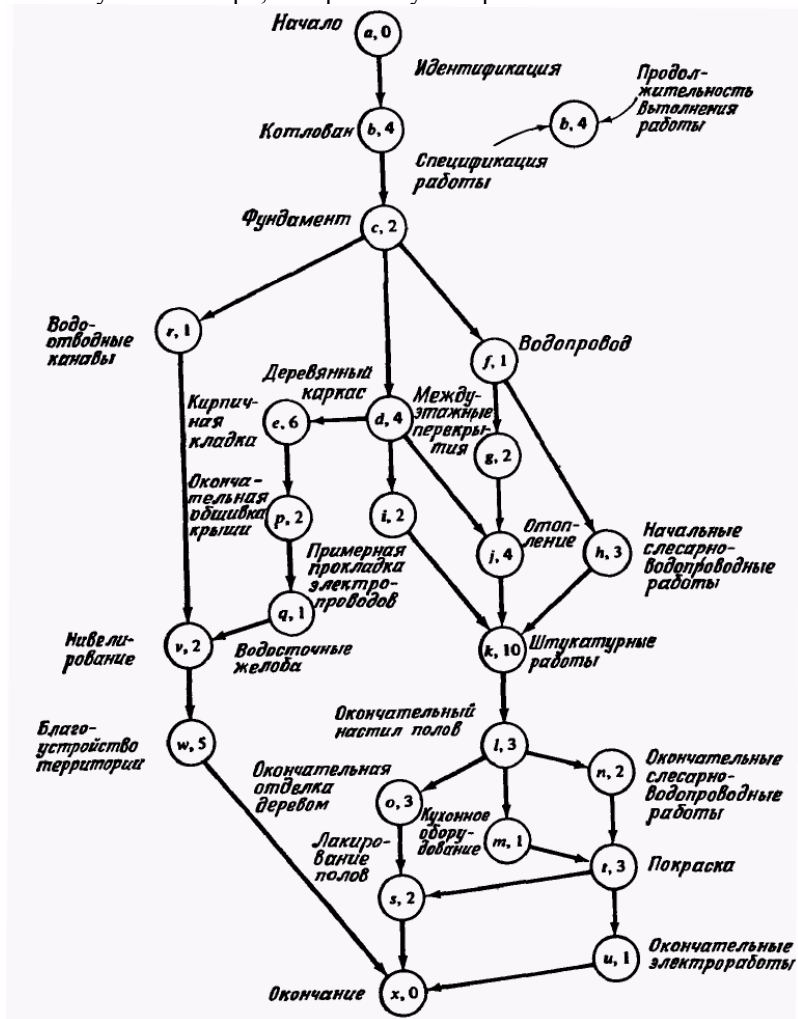


Рис. 35. PERT-диаграмма строительства дома.

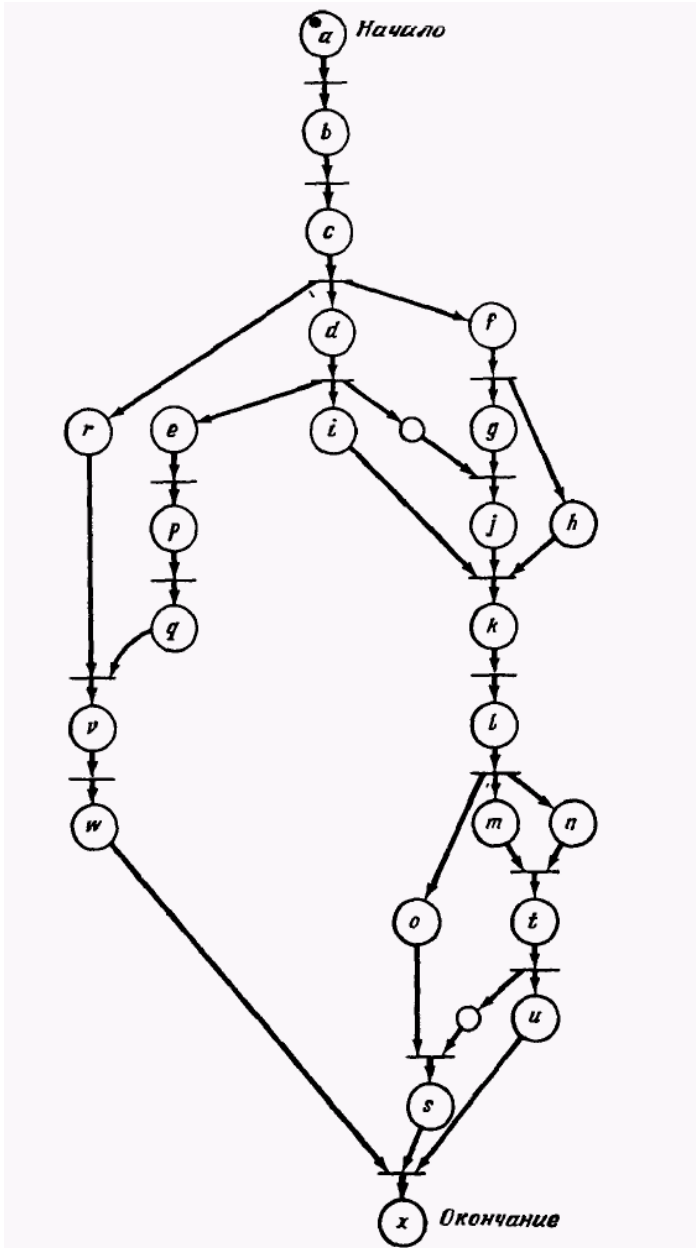


Рис. 36. Представление PERT-диаграммы на рис. 35 сетью Петри.

Кононюк А.Е. Графы

Обратите внимание на то, что введены дополнительные узлы. Они необходимы для адекватного отображения ограничения предшествования и точного представления ситуаций с ожиданием.

Сеть Петри является превосходным средством для представления параллелизма и причинно-следственных связей PERT-диаграммы, но PERT-диаграмма предоставляет также информацию о времени, используемую для определения минимального времени, необходимого для выполнения проекта, — «время позднего начала» и т. д. Сеть Петри не содержит такой информации. Добавление информации о времени может придать сети новое мощное свойство, но это несовместимо с основными концепциями сетей Петри. Для такого расширения сетей Петри требуются дополнительные исследования.

Конвейерная обработка, описанная в разд. 2.2.3.2, является частным случаем производственной системы. Сборочная линия — другой пример производственной системы. Производственные системы и сборочные линии могут быть промоделированы сетями Петри. Одно из первых применений сетей Петри — применение в качестве средства генерации оптимального кода для компилятора с Фортрана CDC 6600. Подход к решению этой задачи заключался в моделировании программы на Фортране сетью Петри способом, подобным моделированию блок-схемы. Затем отдельные операторы программы рассматривались с целью определения минимального набора причинно-следственных связей между операторами, позволяя исключать из сети Петри некоторые из искусственно введенных ограничений на последовательность операторов программы. Эти искусственные ограничения введены из-за необходимости для программиста представлять программу на Фортране в виде последовательности, задающей полный порядок на множестве операторов, даже если требуется только частичное упорядочение. В качестве примера рассмотрим следующие три оператора:

$$10x = x + 1,$$

$$20y = y + 1,$$

$$30z = x + y.$$

Они написаны таким образом, что оператор 10 предшествует оператору 20, но такая последовательность вовсе необязательна. Порядок, в котором будут выполнены операторы 10 и 20 (или они будут выполнены одновременно), не имеет значения для программы. Однако оператор 30 должен следовать только после операторов 10 и 20. После изменения упорядочения операторов необходимо рассмотреть и поток управления. Этот анализ заключается в применении условий Бернштейна для обеспечения детерминированности. Результатом такого анализа является сеть Петри, представляющая программу с

Кононюк А.Е. Графы

минимальными ограничениями на последовательность операторов, т. е. допускающую максимальный параллелизм. Теперь задача состоит в компиляции такой программы. Это требует отображения переменных в регистры и упорядочения инструкций для получения полностью упорядоченной последовательности инструкций машинного языка, некоторые из которых являются инструкциями, реализующими параллельную обработку. CDC 6600 — это ЭВМ с несколькими регистрами и кратными функциональными блоками. Так как функциональные блоки могут выполнять различные инструкции параллельно, то очень важно порождать инструкции в порядке, максимально увеличивающем параллелизм при работе функциональных блоков. На это также влияет распределение переменных по регистрам. Сеть Петри, моделирующая программу и представляющая ограничения, налагаемые программой, объединяется с сетью Петри, моделирующей устройство управления CDC 6600 и представляющей ограничения, налагаемые аппаратным обеспечением. Такая объединенная сеть представляет все возможные последовательности инструкций, которые могут выполняться аппаратурой и реализовать алгоритм данной программы. Эта сеть затем выполняется для получения всех таких последовательностей инструкций. Две (или более) последовательности образуются всякий раз, когда два (или более) перехода одновременно разрешены. Запуск одного перехода будет порождать одну последовательность; запуск другого перехода — другую. Например, сеть Петри на рис. 37 представляет последовательности *abcdef*, *bacdef*, *abcdcf*, *bacedf*.

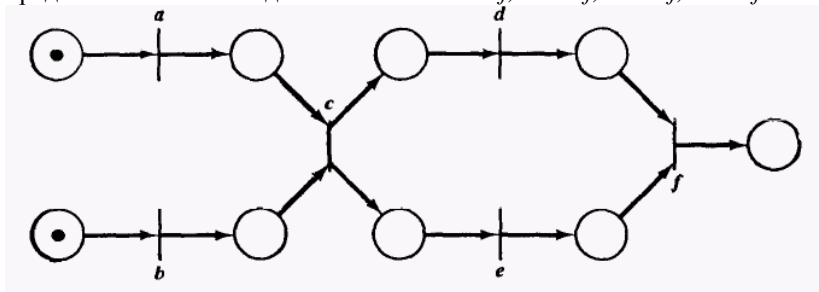


Рис. 37. Сеть Петри, представляющая несколько последовательностей выполнения инструкции.

После того как последовательности получены, вычисляются необходимые затраты времени на их выполнение, и наиболее быстрая

Кононюк А.Е. Графы

последовательность генерируется компилятором для действительного выполнения.

Химические системы — другой пример систем, которые могут быть промоделированы сетями Петри. Химические уравнения моделируются переходами; вещества, участвующие в реакции, — позициями. Количество фишек в позиции указывает на количество данного вещества в системе. Например, сеть Петри на рис. 38 представляет два химических уравнения:

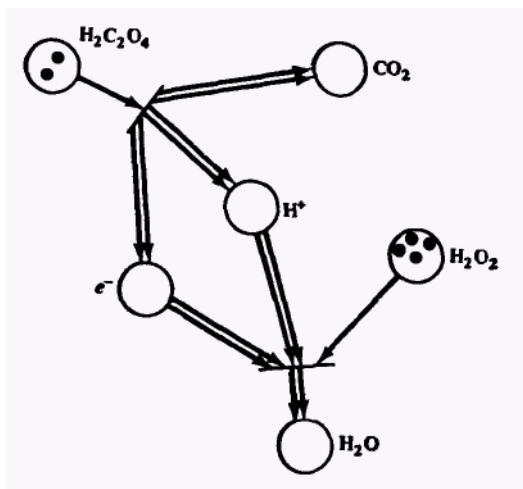
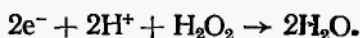


Рис. 38. Сеть Петри, представляющая окислительно-восстановительную реакцию между щавелевой кислотой и пероксидом водорода, в результате которой получаются вода и диоксид углерода.

Можно представить и реакции катализа. Соединение водорода и этилена образует этан ($\text{H}_2 + \text{C}_2\text{H}_4 \rightarrow \text{C}_2\text{H}_6$) только в присутствии платины. Это отражено на рис. 39.

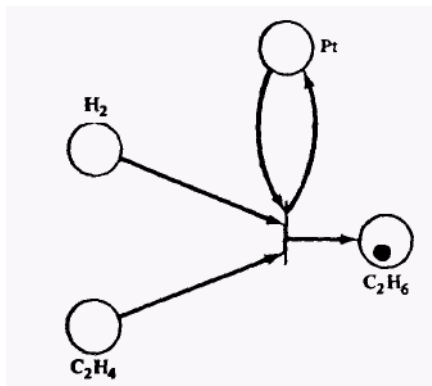


Рис. 39. Процесс получения этиана из водорода и этилена при катализации платиной.

Мельдман и Хольт выдвинули предположение, что и юридические системы могут быть промоделированы сетями Петри. В этих системах несколько действующих лиц (судьи, адвокаты, обвиняемые, клерки и др.) могут одновременно выполнять действия, относящиеся к конкретному делу. Действия и их взаимосвязи могут быть представлены сетью Петри. Например, сеть Петри на рис. 40 является моделью начальных стадий гражданского процесса.

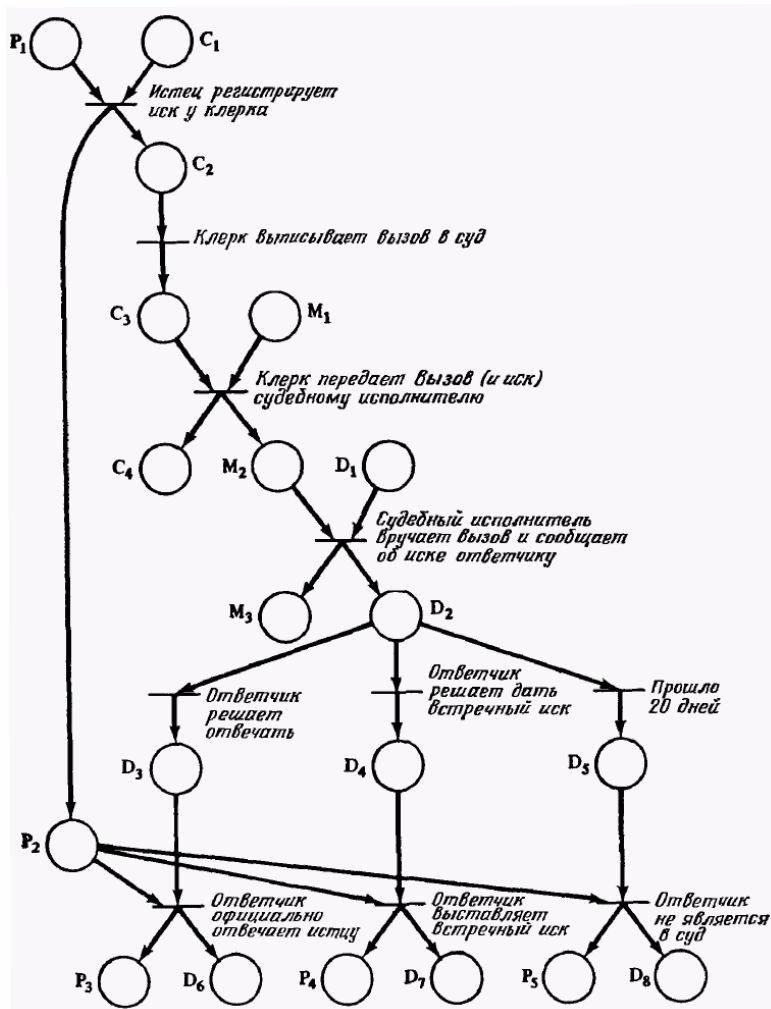


Рис. 40. Сеть Петри, представляющая начальные стадии гражданского Процесса.

Другое предложение заключается в использовании сетей Петри для моделирования и анализа протоколов связи. В сетях ЭВМ и системах распределенных процессов необходима возможность передачи информации между ЭВМ. В этой задаче существенно присутствует параллелизм, и поэтому она попадает в класс задач, для которых

Кононюк А.Е. Графы

определены сети Петри. Фарбер и его ученики разработали методологию для спецификации, проектирования и анализа простых протоколов связи, используя сети Петри и подобные модели.

Другие системы, которые могли быть промоделированы сетями Петри, включают в себя системы очередей (где очереди были бы представлены позициями, а работы — фишками); модели мозга (запуски нейронов могли бы моделироваться запусками переходов); исчисление высказываний (позиции представляют буквы, а переходы объединяют их для определения предложений в конъюнктивной нормальной форме) и многие другие системы. Этот список ограничен в основном временем и воображением человека, но не свойствами сетей Петри.

Однако мы видели по крайней мере один пример (задача о чтении/записи), который нельзя промоделировать с помощью сети Петри. И хотя моделирование с помощью сетей Петри может быть полезным при описании системы, необходимо разработать средства анализа, которые бы позволили исследовать сеть Петри и определить ее свойства. Обратимся к этому в главе 2.4, в которой представлены методы анализа сетей Петри.

2.3. Свойства сетей Петри

Для постановок задач анализа сетей Петри прежде всего необходимо указать и формально определить те свойства сетей, которые целесообразно анализировать. Естественно, что при выборе таких свойств главную роль играет ориентация на практические задачи, возникающие при исследовании моделируемых сетями дискретных систем. Когда такие свойства выделены, ставится вопрос о возможности их автоматического распознавания в произвольных сетях Петри или в некоторых частных случаях сетей. Наконец, третий этап исследований состоит в нахождении оптимальных алгоритмов распознавания тех свойств сетей, которые оказываются принципиально распознаваемыми.

2.3.1. Основные свойства сетей Петри

Первое из рассматриваемых ниже свойств сетей Петри связано с ограниченной емкостью реальных условий реализации событий. Действительно, из определения правил срабатывания переходов сети следует, что для реализации события, моделируемого некоторым

Кононюк А.Е. Графы

переходом, достаточно, чтобы каждое его входное место-условие содержало некоторое конечное число фишек, равное кратности дуги, соединяющей его с переходом. Однако при работе сети Петри общего вида некоторые ее места могут накапливать неограниченное число фишек (например, место p_2 в сети на рис. 3 раздел 2.1). Если интерпретировать места как накопители (буферы) данных, сигналов или деталей в моделируемых системах, то естественно потребовать, что при любом варианте функционирования этих систем не происходило бы переполнение накопителей, которые в реальных ситуациях имеют конечную, фиксированную емкость. Следующие понятия формализуют такие требования.

Место p в сети Петри $N = (P, T, F, W, M_0)$ называется *ограниченным*, если существует число n такое, что для любой достижимой в сети разметки M справедливо неравенство $M(p) \leq n$. Сеть N называется *ограниченной сетью*, если любое ее место ограничено. Ясно, что множество достижимых разметок $R(N)$ конечно, если и только если N — ограниченная сеть. В сети на рис. 3 раздел 2.1 места p_1 и p_3 ограничены, так как каждое из них может содержать не более одной фишки. В то же время место p_2 не ограничено, и поэтому эта сеть не является ограниченной. Сеть на рис. 2 раздел 2.1 ограничена, любое место в ней не может содержать более 2 фишек. Место p называется *безопасным*, если $\forall M \in R(N): M(p) \leq I$; соответственно *сеть безопасна*, если все ее места безопасны. Любая достижимая в безопасной сети разметка представляет собой вектор из 0 и 1. Сети на рис. 2 и 3 раздел 2.1 не являются безопасными (в отличие от сети на рис. 4, см. ниже). Ограниченность и безопасность характеризуют емкость условий: в дискретной информационной системе, моделируемой соответствующими сетями, можно ограничить емкость накопителей, необходимых для хранения условий наступления событий. Родственным этим понятиям является понятие *консервативной сети*, в которой сумма фишек во всех ее местах остается постоянной при работе сети, т.е.

$$\forall M_1, M_2 \in R(N): \sum_{p \in P} M_1(p) = \sum_{p \in P} M_2(p).$$

В консервативной сети каждый переход консервативен в том смысле, что его срабатывание не меняет число фишек в сети, т.е. $|t| = |t'|$.

Переход в сети может сработать при определенных условиях, связанных с разметкой его входных мест. В общем случае может оказаться, что для некоторого перехода условие его срабатывания никогда не выполняется, как бы ни функционировала сеть. Такой переход - лишний в сети, его можно исключить без ущерба для работы

Кононюк А.Е. Графы

сети. Может случиться также, что после некоторой последовательности срабатываний переходов сети и соответствующих изменений ее разметки некоторые переходы, в том числе те, которые уже срабатывали, больше никогда не сработают, какие бы варианты достижимых в сети разметок не возникали. Это означает, что в моделируемых системах могут появляться ситуации, тупиковые для некоторых событий, "исключающие их из дальнейшей игры". Например, в операционных системах подобные случаи имеют место при взаимных блокировках процессов (deadlocks). Следующие определения связаны с выявлением таких ситуаций в сетях Петри.

Переход t в сети Петри $N = (P, T, F, W, M_0)$ называется *потенциально живым при разметке* $M \in R(N)$, если

$$\exists M' \in R(N, M): M' \geq F(t),$$

т.е. существует достижимая от M разметка M' , при которой переход t может сработать. Если $M=M_0$, то t называется *потенциально живым в сети* N . Переход t — *мертвый при* M , если он не является потенциально живым при M , Переход t — *мертвый*, если он мертв при любой достижимой в сети разметке.

Переход t в сети Петри N называется *живым*, если

$$\forall M \in R(N), \exists M' \in R(N, M): M' \geq F(t),$$

т.е. переход t является потенциально живым при любой достижимой в сети N разметке. Переход t — *потенциально мертвый*, если существует $M \in R(N)$ такая, что при любой разметке $M' \in R(N, M)$ переход t не может сработать. Разметка M в этом случае называется *t-тупиковой*; если она *t-тупиковая* для всех $t \in T$, то она является *тупиковой*. Если M — тупиковая разметка, то $R(N, M) = \{M\}$. Сеть называется *живой*, если все ее переходы живы.

В сети на рис. 2 раздел 2.1 все переходы потенциально живы и все переходы потенциально мертвы, так как в ней достижима тупиковая разметка (рис. 2, г раздел 2.1). Эта сеть не может быть живой, так как содержит потенциально мертвые переходы. Сеть на рис. 3 раздел 2.1 также не является живой, так как в ней достижимы тупиковые разметки вида $(0, n, 0)$, $n \geq 0$.

Переход t называется *устойчивым* в сети N , если

$$\forall t' \in T \setminus \{t\}, \forall M \in R(N): \\ (M \geq F(t)) \wedge (M \geq F(t')) \Rightarrow (M \geq F(t) + F(t')),$$

т.е. если переход t может сработать, то никакой другой переход не может, сработав, лишить его этой возможности. Сеть N устойчива, если все ее переходы устойчивы.

В сети на рис. 2 раздел 2.1 устойчивы переходы t_1, t_2, t_3, t_4 , в сети на рис. 3 раздел 2.1 устойчив только переход t_2 . Однако обе сети не

устойчивы, так как в первой сети не устойчивы переходы t_5, t_6 , во второй — t_1, t_3, t_4 .

Конечная цель специальной теории сетей Петри — автоматический анализ свойств сетей, их автоматический синтез и преобразования, на основании чего могут быть построены практические алгоритмы анализа, синтеза и преобразований дискретных систем, моделируемых сетями. В частности, полезно найти алгоритмы, с помощью которых для любой предъявленной сети можно установить, обладает ли она интересующим нас свойством — является ли она ограниченной, живой, устойчивой и т.п. В первую очередь нужно выяснить существование таких алгоритмов. Эти вопросы формулируются как массовые алгоритмические проблемы для сетей: проблема ограниченности (существует ли алгоритм, который позволяет узнать, является ли данная сеть ограниченной), проблема потенциальной живости переходов, проблема живости сетей, проблема устойчивости, проблема безопасности. Говорят, что проблема разрешима, если соответствующий алгоритм распознавания свойств существует, в противном случае проблема неразрешима.

Большинство из перечисленных выше проблем связано с определением возможности достижения некоторых специальных разметок в сети (например, с достижением t -тупиковых разметок для данного перехода t), т.е. с проблемой достижимости заданной разметки. В этой проблеме ставится вопрос о существовании алгоритма, с помощью которого можно узнать для произвольной сети N и произвольной разметки M , принадлежит ли M множеству $R(N)$. Ниже будет показано, например, что проблемы живости и достижимости эквивалентны в том смысле, что решение одной из них автоматически решает другую.

Несколько особняком стоят проблемы R -включения и

R -эквивалентности сетей Петри. Пусть задан класс сетей, которые имеют одно и то же множество мест (или их множества мест изоморфны). В первом случае необходимо выяснить существование алгоритма, устанавливающего для любых двух сетей из этого класса, имеет ли место соотношение $R(N_1) \subseteq R(N_2)$, во втором - $R(N_1) = R(N_2)$.

2.3.2. Проблемы ограниченности и безопасности

Некоторое место p в сети N может оказаться неограниченным по двум причинам. Первая заключается в следующем. Пусть свободный язык сети $L(N)$ содержит последовательность срабатываний $\tau = \tau_1\tau_2$ такую, что

$$(M_0 [\tau_1 \rangle M_1] \wedge (M_1 [\tau_2 \rangle M_2] \wedge (M_2 \geq M_1) \wedge (M_2(p) > M_1(p))).$$

Поскольку $M_2 \geq M_1$, то любая подпоследовательность срабатываний, начинающаяся при разметке M_1 , может повториться и начиная с разметки M_2 . Это означает, что любая последовательность вида $\tau_1 \tau_2^n$ также принадлежит $L(T)$, каким бы большим не было n . Следовательно, разметка места p может безгранично расти. Например, для места p_2 в сети на рис. 1 существует $\tau = \lambda t_1$, где λ — пустое слово, такое, что

$$(M_0 = (1, 0, 0, 0) [t_1 \rangle M_1 = (1, 1, 0, 0)] \wedge (M_1 \geq M_0) \wedge (M_1(p_2) = 1 > M_0(p_2) = 0).$$

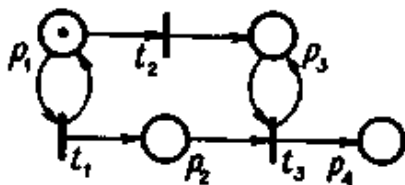


Рис.1

Однако место p_4 не ограничено по другой причине, так как нельзя найти пару достижимых в этой сети разметок M_1 и M_2 таких, что $(M_2 \geq M_1) \wedge (M_2(p_4) > M_1(p_4))$, хотя и ясно, что последовательности срабатываний вида $t^n t_2^n$ ведут к накоплению любого числа n фишек в p_4 . Можно отметить следующую разницу в неограниченности мест p_2 и p_4 : в p_2 может быть "бесконечное" число фишек, а в p_4 — сколь угодно большое, но конечное число фишек, так как переход t_3 может сработать только конечное число раз, не большее, чем число срабатываний перехода t_1 . Таким образом, неограниченность места p_4 "вторична" по отношению к неограниченности места p_2 .

Выше было дано определение графа разметок сети Петри, характеризующего динамику ее функционирования. Исследование проблемы ограниченности сводится к анализу графа другого типа, а именно — *покрывающего дерева* сети. Для любой сети такой граф конечен и может быть построен с помощью следующей процедуры:

- 1) Первоначально предполагается, что дерево содержит единственную вершину-корень M_0 и не имеет дуг.
- 2) Пусть M - вершина дерева, которая еще не объявлена листом (т.е. вершиной, из которой не исходит ни одна дуга), но в дереве нет исходящих из нее дуг. Возможны четыре случая.

Кононюк А.Е. Графы

а) Ни один из переходов сети не может сработать при разметке M , т.е.

$\forall t \in T: M \not\geq F(t)$. В этом случае вершина M объявляется листом.

б) На пути из корня дерева в вершину M существует вершина M' такая, что $M = M'$. Вершина M объявляется листом.

в) На пути из корня дерева в вершину M существует вершина M' такая, что $M' < M$. Для любого места p такого, что $M'(p) < M(p)$, значение, соответствующей координаты M заменяется на ω и вершина M объявляется

ω -листом.

г) Если ни один из вышеперечисленных случаев не имеет места, то M — внутренняя вершина дерева. Для каждого перехода $t \in T$ такого, что

$M \geq F(t)$, в дерево добавляется новая вершина $M' = M - F(t) + F'(t)$ и дуга, ведущая из M в M' , помеченная символом t .

На рис. 2,а показано покрывающее дерево для сети Петри, изображенной на рис. 3 раздел 2.1, а на рис. 2,б - для сети на рис. 1.

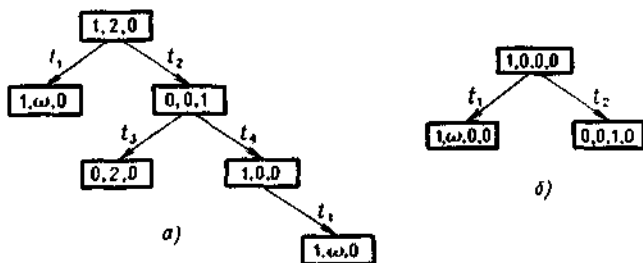


Рис. 2.

Из правил построения покрывающего дерева для сети Петри следует, что вершины дерева представляют собой векторы из множества $\mathbf{N}_{\omega}^n = \underbrace{\mathbf{N}_{\omega} \times \dots \times \mathbf{N}_{\omega}}_n$.

где n — число мест в сети.

Введем несколько определений и отметим некоторые свойства множеств и последовательностей векторов из \mathbf{N}^n и \mathbf{N}_{ω}^n . Для отношения частичного порядка \leq множество \mathbf{N}^n является решеткой, а множество \mathbf{N}_{ω}^n является полной решеткой и любое подмножество $X \subseteq \mathbf{N}_{\omega}^n$ имеет единственную наименьшую верхнюю грань $\vee \in \mathbf{N}_{\omega}^n$

$$(\forall \lambda \in X: (\lambda \leq v) \wedge \forall v' \in \mathbf{N}_{\omega}^n: (\forall \lambda \in X: (\lambda \leq v') \rightarrow (v \leq v'))).$$

Кононюк А.Е. Графы

Цепью C в \mathbf{N}_{ω}^n называется последовательность (L_1, L_2, \dots) такая, что $\forall i \geq 1, \forall j \geq 1: L_i \leq L_{i+1} \wedge L_i \neq L_j$. Множество $X \in \mathbf{N}_{\omega}^n$ замкнуто, если любая цепь в X содержит свою наименьшую верхнюю грань. Поскольку \mathbf{N}_{ω}^n - полная решетка, любая цепь в \mathbf{N}_{ω}^n замкнута, в то время как бесконечные цепи из \mathbf{N}^n не содержат своих наименьших верхних граней.

Множество $X \subseteq \mathbf{N}_{\omega}^n$ монотонно, если

$$\forall L \in X, \forall L' \in \mathbf{N}_{\omega}^n: L' \leq L \Rightarrow L' \in X.$$

Для множества $X \in \mathbf{N}_{\omega}^n$, его множество *максимальных элементов* определяется как

$$\{L \in X \mid \neg(\exists L' \in X: L' \geq L \wedge L' \neq L)\}.$$

Для множества $X \subseteq \mathbf{N}_{\omega}^n$ его *замыканием* называется наименьшее замкнутое множество из \mathbf{N}_{ω}^n , содержащее X .

Лемма 1. Любое бесконечное подмножество множества \mathbf{N}_{ω}^n , содержит бесконечную цепь.

Доказательство. Любая бесконечная цепь из \mathbf{N}_{ω}^n содержит бесконечную неубывающую последовательность. Любое бесконечное подмножество множества \mathbf{N}_{ω}^n можно упорядочить так, чтобы оно не содержало одинаковых векторов. Из полученной бесконечной последовательности выбирается бесконечная подпоследовательность так, чтобы все первые координаты векторов следовали в неубывающем порядке. Из этой подпоследовательности выбирается такая, что все вторые координаты следуют в неубывающем порядке, и т.д. для всех n координат. Результатом является бесконечная последовательность, не убывающая в каждой координате и не содержащая одинаковых векторов.

Лемма 2. Любое множество взаимно несравнимых (по отношениям $\leq, \geq, =$) векторов из \mathbf{N}_{ω}^n конечно.

Доказательство. Из леммы 1 следует, что любая бесконечная цепь в множестве \mathbf{N}_{ω}^n состоит из разных сравнимых по упомянутым отношениям векторов. Следовательно, множество несравнимых векторов должно быть конечным.

Теорема 1. Для любой сети N ее покрывающее дерево конечно.

Доказательство. Предположим, что дерево бесконечно. По построению, из каждой его вершины может исходить дуг не больше, чем число переходов в сети, которое конечно. По лемме Кёнига в дереве должен существовать бесконечный путь, начинающийся в корне. По лемме 1 должна существовать бесконечная цепь векторов-разметок,

Кононюк А.Е. Графы

представляющая собой подпоследовательность этой цепи, а в ней существуют две разметки M и M' такие, что M' предшествует M по дереву и $M \geq M'$. Но, по построению дерева, вершина M должна быть листом, что противоречит предположению о бесконечности выбранного пути.

Теорема 2. *Проблема ограниченности сети Петри разрешима.*

Доказательство. Поскольку для любой сети N покрывающее дерево конечно, описанная выше процедура его построения эффективна, т.е. может быть выполнена за конечное число шагов. Если дерево содержит лист с разметкой M , содержащей ω , то сеть не ограничена. Действительно, в этом случае существуют, по построению, внутренняя вершина M' и подпоследовательность срабатываний τ такие, что $(M' \{ \tau \} M) \wedge (M > M')$.

Подпоследовательность τ может повторяться сколь угодно большое число раз, порождая последовательность возрастающих разметок.

Если же дерево не содержит вершин-разметок с символом ω , то для каждой его вершины M , отличной от туиковой, существует переход $t \in T$ и другая вершина M' такая, что $M \neq M'$ и $M \{ t \} M'$. Учитывая, что $M = M_0$ принадлежит $R(N)$, убеждаемся, что разметка из $R(N)$ представлена в дереве. Так как дерево конечно, то число достижимых в сети разметок конечно и соответственно каждая из разметок представляет собой вектор из "конечных" чисел.

Таким образом, алгоритм распознавания ограниченности сети состоит в построении ее покрывающего дерева и последующего просмотра конечного числа вершин-разметок. Если будет найдена хотя бы одна разметка с символом ω , то сеть не ограничена, в противном случае — ограничена.

С помощью покрывающего графа можно установить лишь глобальный факт, является ли заданная сеть ограниченной или нет, так как в этом дереве ω появляется только в тех позициях разметок-векторов, которые соответствуют местам с потенциально неограниченной разметкой, возникающей по первой из двух указанных в начале параграфа причин неограниченности мест. Например, в покрывающем дереве на рис. 2, б символ ω появляется только в позиции, соответствующей месту p_2 сети на рис. 1, но нет символа ω , соответствующего месту p_4 . Поэтому с помощью алгоритма распознавания ограниченности из теоремы 1 нельзя установить, какие места сети могут иметь "вторичную" неограниченную разметку. Чтобы иметь такую возможность, следует построить *полное покрывающее дерево* сети по следующей процедуре.

1. Строится покрывающее дерево сети N . Если это дерево не имеет ω -листов, то построение полного дерева закончено и оно совпадает с покрывающим деревом.

2. Если покрывающее дерево содержит ω -лист M , то для него строится покрывающее дерево для сети N' , полученной из N заменой начальной разметки M_0 на разметку M . При этом правила срабатывания переходов и изменение разметки сети обобщаются на случай векторов с ω с учетом арифметических свойств расширенного множества натуральных чисел. Построенное дерево присоединяется к основному дереву совмещением корня M в новом дереве с листом M в основном дереве. Процесс повторяется до тех пор, пока не исчезнет последний ω -лист.

На рис. 3,а показано полное покрывающее дерево для сети, изображенной на рис. 3 раздел 2.1, на рис. 3,б — для сети на рис. 1.

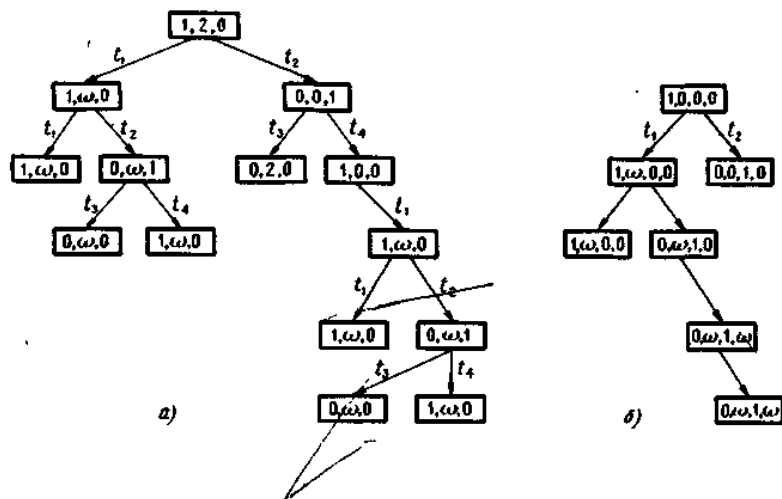


Рис. 3

В первом случае полное дерево не дает новой информации по сравнению с покрывающим деревом, так как сеть на рис.3 раздел 2.1 имеет единственное неограниченное место "первого рода". Во втором случае в полном дереве символ ω появляется в позиции места p_4 неограниченного места "второго рода".

Теорема 3. Проблема ограниченности некоторого места в произвольной сети Петри разрешима.

Доказательство. Из построения полного покрывающего дерева для сети N легко устанавливается, что это дерево конечно. Произвольное место p не ограничено, если и только если среди вершин полного дерева существует вершина M такая, что в позиции места p стоит символ ω . Действительно, как следует из доказательства теоремы 1, в

Кононюк А.Е. Графы

покрывающем дереве сети N для любой его вершины M существует в замыкании множества $R(N)$ разметка M' такая, что $M' \geq M$. По построению полного дерева, это свойство сохраняется для всех его вершин.

С другой стороны, если зафиксирована произвольная разметка $M' \in R(N)$, то в полном покрывающем дереве есть вершина с разметкой $M \geq M'$. Из определения неограниченности места p следует, что оба неравенства имеют место только в том случае, если хотя бы одна из вершин полного дерева содержит ω в позиции места p .

Теорема 4. *Существует алгоритм, с помощью которого можно установить для произвольного вектора $M \in \mathbf{N}^n$ и для произвольного перехода t в сети Петри с n местами, является ли вектор M t -тупиковым.*

Доказательство. Заменяем в сети N начальную разметку M_0 на разметку M , после чего построим полное покрывающее дерево для полученной сети с начальной разметкой M , являющейся корнем этого дерева. Ясно, что M — t -тупиковый вектор тогда и только тогда, когда построенное полное покрывающее дерево не содержит дуги, помеченной символом t .

Следующие теоремы легко доказать, исходя из основных свойств (полных) покрывающих деревьев.

Теорема 5. *Проблема безопасности сетей Петри разрешима.*

(Сеть с n местами безопасна, если и только если все вершины ее покрывающего дерева представляют собой векторы из множества $\{0,1\}^n$, т.е. векторы составленные из 0 и 1). Пример безопасной сети дан на рис. 4.

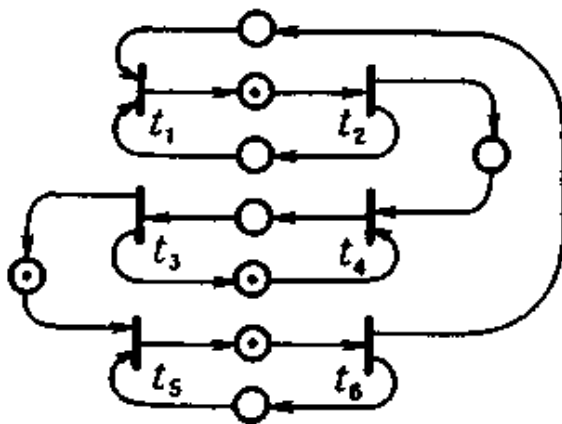


Рис. 4.

- Теорема 6.** *Проблема потенциальной живости переходов разрешима.* (Переход t потенциально живой, если и только если он метит некоторую дугу в полном покрывающем дереве сети хотя бы один раз.)
- Теорема 7.** *Существует алгоритм, с помощью которого можно узнать, получит ли данное место в сети хотя бы одну фишку.* (Если хотя бы в одной вершине покрывающего дерева позиция, соответствующая месту p , содержит число $n > 0$, то место p может получить фишку в процессе функционирования сети.)
- Теорема 8.** *Существует алгоритм, с помощью которого можно узнать, может ли данный переход сработать сколь угодно большое число раз.* (Достаточно присоединить к данному переходу новое "висячее" выходное место и затем выяснить, является ли оно неограниченным.)

2.3.3. Проблемы R -включения и R -эквивалентности

Обе проблемы не являются разрешимыми. Доказательства были проведены Рабином и Хаком сведением 10-й проблемы Гильберта к рассматриваемым проблемам. Неразрешимость 10-й проблемы была установлена Матиясевичем. Проблема Гильберта формулируется следующим образом: существует ли алгоритм, с помощью которого можно выяснить, имеет ли полином $f(x_1, \dots, x_n)$ с целыми рациональными коэффициентами целое рациональное решение, т.е. существует ли вектор чисел (y_1, \dots, y_n) такой, что $f(y_1, \dots, y_n) = 0$?

Метод сведения состоит в том, что устанавливается такая связь между решенной и решаемой проблемой, при которой допущение о разрешимости решаемой проблемы приводит к разрешимости проблемы, о которой известно, что она неразрешима. В сведении Рабина и Хака используется промежуточная проблема включения графов полиномов, в которой рассматриваются полиномы с целыми неотрицательными коэффициентами. Эта проблема состоит в следующем. Пусть $g(f)$ обозначает множество

$\{(x_1, \dots, x_n, z) \in \mathbb{N}^{n+1} \mid z \leq f(x_1, \dots, x_n)\}$. Можно ли для двух произвольных полиномов f_1 и f_2 с одинаковым числом переменных выяснить, имеет ли место включение $g(f_1) \subseteq g(f_2)$? Сведением проблемы Гильберта к данной проблеме показано, что последняя неразрешима.

Для сведения проблемы включения графов полиномов к проблемам

Кононюк А.Е. Графы

R -включения и R -эквивалентности Рабин предложил использовать сети в качестве абстрактных машин, "слабо вычисляющих" полиномы. Машина слабо вычисляет полином $f(x_1, \dots, x_n)$, если для произвольного "входного" вектора чисел (y_1, \dots, y_n) результатом на "выходе" машины может быть любое число z такое, что $0 \leq z \leq f(y_1, \dots, y_n)$. Это означает, что машина недетерминирована в том смысле, что вместо единственного результата, однозначно определяемого входным вектором (y_1, \dots, y_n) , она может выдать любой из векторов, не превышающих $f(y_1, \dots, y_n)$. Недетерминированность результата работы машины отражает недетерминированность функционирования сети Петри.

Конечная сеть (R, T, F, W) . "слабо вычисляющая" полином $f(x_1, \dots, x_n)$, должна удовлетворять следующим условиям:

- 1) в ней выделены n "входных" мест in_1, \dots, in_n , каждое из которых сопоставлено соответствующему аргументу полинома, и одно "выходное" место out ;
- 2) для управления работой сети выделены специальные места: "стартовое" on и "выключающее" off ;
- 3) произвольному вектору $y \in \mathbb{N}^n$ можно сопоставить начальную разметку M_y , такую, что

$$M_y(on) = 1, \quad \forall i, 1 \leq i \leq n: M_y(in_i) = y_i,$$

все остальные места имеют нулевую разметку;

- 4) для любой разметки M , достижимой в сети (P, T, F, W, M_y) ,

$$M(out) \leq f(y_1, \dots, y_n);$$

- 5) для любого целого k , $0 \leq k \leq f(y_1, \dots, y_n)$, существует достижимая в сети с начальной разметкой M_y тупиковая разметка M такая, что $M(out) = k$, $M(off) = 1$.

В ряде работ показано, что для любого полинома с неотрицательными коэффициентами можно конструктивно построить слабо вычисляющую его сеть. Опуская технические детали доказательства, представим только его идею.

Полином $f(x_1, \dots, x_n)$ является конечной суммой одночленов вида

$$\sum_{i=1}^k h_i(x_1, \dots, x_n), \quad \text{где } h_j(x_1, \dots, x_n) \text{ представляет собой}$$

произведение $c_j x_1^{m_{j1}} \dots x_n^{m_{jn}}$ с неотрицательным коэффициентом c_j .

Таким образом, чтобы построить сеть, слабо вычисляющую полином, нужно уметь моделировать сетями умножение и сложение неотрицательных чисел, умножение на константу, сравнение на равенство и суперпозицию сетей. На рис. 5, а и б и 6, а показаны сети, слабо вычисляющие сумму $x_1 + x_2$, произведения $x_1 \cdot x_2$ и $c \cdot x$.

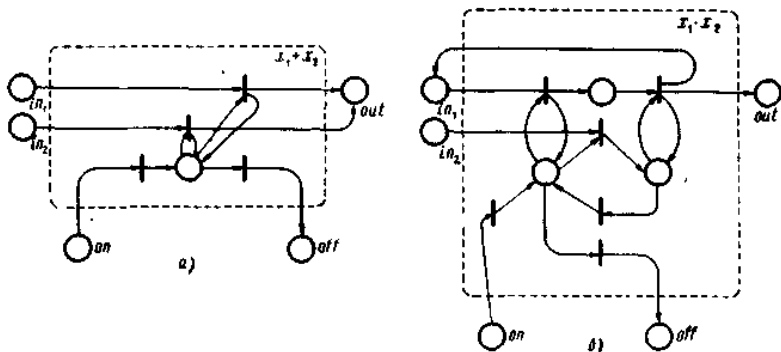


Рис. 5

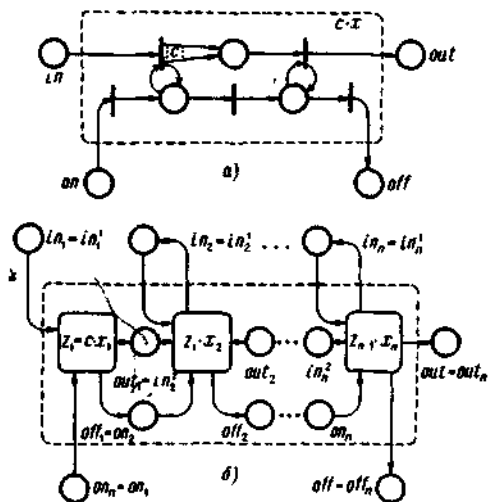


Рис.6

На рис. 6, б показана сеть, слабо вычисляющая одночлен $c \cdot x_1 \cdot \dots \cdot x_n$. Эта сеть представляет собой комбинацию сетей, представленных на рис. 5 и 6, а. Сеть, слабо вычисляющая полином с k одночленами, строится как суммирующая комбинация сетей, типа построенной на рис. 6, б (суммирование k слагаемых обобщает пример на рис. 5,а).

Лемма 3. Для любого полинома $f(x_1, \dots, x_n)$ с неотрицательными целыми коэффициентами можно построить сеть Петри N такую, что $g(f) = R_n(N)$, где $g(f)$ - граф полинома f , а $R_n(N)$ — множество проекций векторов из $R(N)$ на их первые $n + 1$ координату.

Кононюк А.Е. Графы

(Множество проекций $R(N) = \{M(P_{n+1}) | M \in R(N)\}$, где P_{n+1} - вектор, составленный из первых $n + 1$ координат вектора мест P , и $n \leq k = |P|$)

Доказательство. Основой для построения сети Петри N служит сеть N_f слабо вычисляющая полином f . К этой сети добавляются переходы t_0, t_1, \dots, t_n и места p_1, p_2, \dots, p_n . Месту out сети N_f присваивается новый символ p_{n+1} . Дуги связывают добавленные места и переходы друг с другом, а также с входными местами in_1, \dots, in_n и местом on сети N_f так, как показано на рис. 7.

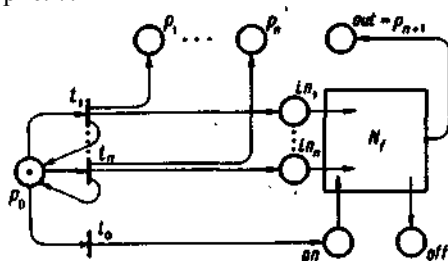


Рис. 7.

Место p_0 имеет разметку $M_0(p_0) = 1$. Стартовое место on сети N_f получает фишку после срабатывания перехода f_0 . Однако до этого момента могут сработать любые из переходов t_1, \dots, t_n произвольное число раз каждый, переслав во входные места in_1, \dots, in_n и дополнительные места p_1, \dots, p_n соответствующее число фишек. Таким образом, к моменту, когда стартовое место on получает возможность запустить в работу сеть N_f , каждое входное место $in_i, 1 \leq i \leq n$, имеет разметку $M(in_i)$, каждое место $p_i, 1 \leq i \leq n$, имеет разметку $M(p_i)$ и $M(in_i) = M(p_i)$. В процессе дальнейшей работы сети $\forall M' \in R(N): 0 \leq M'(p_{n+1}) \leq f(M(p_1), \dots, M(p_n))$. Следовательно, проекции достижимых разметок на места p_1, \dots, p_n, p_{n+1} образуют множество $R_n(N) = g(f)$.

Теорема 9. *Проблема R-включения неразрешима.*

Доказательство. Для двух произвольных полиномов f и h с одинаковым числом переменных n можно построить сети Петри N и N' такие, что $g(f) \subseteq g(h) \Leftrightarrow R(N) \subseteq R(N')$. Пусть N_f и N_h - сети, слабо вычисляющие полиномы f и соответственно h , причем, $g(f) = R_n(N_f)$ и $g(h) = R_n(N_h)$ (см. лемму 3). Если первая сеть имеет k мест, а вторая l , то число мест в обеих сетях можно уравнивать, добавив $|k - l|$ мест в сеть с меньшим числом мест. Чтобы эти места не изменили множество достижимых разметок, начальная разметка каждого из них полагается равной 0, они не связываются дугами ни с одним из переходов сети и

Кононюк А.Е. Графы

добавляется новый (мертвый) переход, для которого они являются входными местами.

Сети N и N' строятся по сетям N_f и N_h (одна из них имеет дополнительные места) так, как показано на рис. 8.

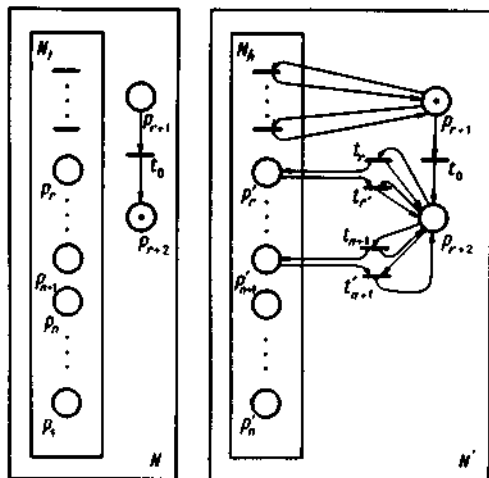


Рис.8

Сети N_f и N_h представлены прямоугольниками, в которых показаны все их переходы и места, но не указаны связывающие их дуги. В каждую из сетей добавляются места p_{r+1} и p_{r+2} , где $r = \max(k, l)$. К сети N_f добавлен также переход t_0 , который никогда не сработает, поэтому разметка мест p_{r+1} и p_{r+2} постоянна в сети N . К сети N_h добавлены новые переходы, место p_{r+1} является входным и выходным для всех переходов сети N_h , переходы t_i и t'_i , где $n + 1 \leq i \leq r$, связаны с местами p'_{n+1}, \dots, p'_r так, как показано на рис. 8.

Пока переход t_0 не сработал и не перевел фишку из места p_{r+1} в p_{r+2} , сеть N' функционирует точно так же, как и сеть N_h . После срабатывания t_0 ни один из переходов сети N_h не может сработать и разметка мест p'_{n+1}, \dots, p'_n не может измениться. В то же время "новые" переходы t_i и t'_i , $n + 1 \leq i \leq r$, могут произвольным образом изменять разметку мест p'_{n+1}, \dots, p'_n . Если вектор $X \in \mathbb{N}^n$ и вектор $Y \in \mathbb{N}^{r-n}$, где $n \leq r$, то через Z будем обозначать "конкатенацию" векторов X и Y , т.е. вектор $Z \in \mathbb{N}^r$, первые n компонент которого совпадают с X , а компоненты с $(n + 1)$ -й по r -й совпадают с компонентами вектора Y . Если U — множество векторов из \mathbb{N}^n , а V — множество векторов из \mathbb{N}^{r-n} , то через $U \times V$ будем обозначать множество попарных конкатенации векторов из U и V .

Кононюк А.Е. Графы

Из анализа функционирования сети N' следует, что

$$R(N') = R(N_h) \times \{(1, 0)\} \cup R_n(N_h) \times N^{r-n} \times \{(0, 1)\}.$$

Из анализа функционирования сети N следует, что

$$R(N) = R(N_f) \times \{(0, 1)\}$$

Таким образом, $R(N) \subseteq R(N') \Leftrightarrow R(N_f) \subseteq R_n(N_h) \times N^{r-n}$.

Правая часть эквивалентна отношению $R_n(N_f) \subseteq R_n(N_h)$. Следовательно,

$$R(N) \subseteq R(N') \Leftrightarrow R_n(N_f) \subseteq R_n(N_h).$$

Это означает, что для решения проблемы включения графов двух произвольных полиномов f и h достаточно построить сети N_f , N_h , N , N' , убедиться, имеет ли место включение $R(N) \subseteq R(N')$, на основании этого сделать заключение о том, имеет ли место $R_n(N_f) \subseteq R_n(N_h)$, и на основании леммы 3 установить, содержится ли $g(f)$ в $g(h)$. Таким образом, если предположить разрешимость проблемы включения для множеств достижимых разметок сетей Петри, получаем разрешимость проблемы включения графов полиномов с неотрицательными коэффициентами, что не верно.

Теорема 10. *Проблема R -эквивалентности неразрешима.*

Доказательство. Для доказательства разрешимости методом сведения достаточно показать, что проблема R -включения сводится к проблеме R -эквивалентности. Это означает, что нужно указать способ, с помощью которого для любых двух сетей A и B можно построить сети A' и B' такие, что

$$R(A) \subseteq R(B) \Leftrightarrow R(A') = R(B').$$

Обе сети A' и B' строятся добавлением разных переходов к одной и той же сети C , которая, в свою очередь, строится из сетей A и B . Она как бы кодирует объединение $R(A) \cup R(B)$, и ниже будет использован тот факт, что $R(A) \subseteq R(B) \Leftrightarrow R(B) = R(A) \cup R(B)$. Способ конструирования сетей C , A' , B' показан на рис. 9.

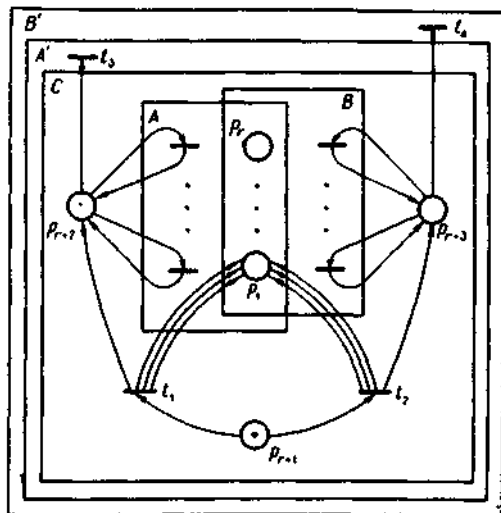


Рис.9

Предполагается, что сети A и B имеют одно и то же множество мест (в противном случае добавляются новые места, как в доказательстве теоремы 9, и сопоставленные друг другу места помечаются одинаковыми символами). Из сетей A и B убрана начальная разметка. Она устанавливается переходами t_1 и t_2 способом, указанным на рис. 10.

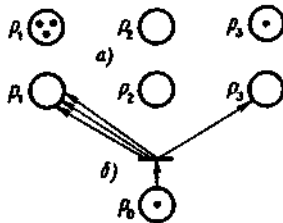


Рис. 10

На нем показан пример, как начальная разметка мест p_1, p_2, p_3 может быть задана с помощью дополнительного перехода и одного места p_0 с единичной начальной разметкой.

Переход t_1 "запускает" сеть A , передав фишку из места p_{r+1} в место p_{r+2} . Последнее является входным и выходным для всех мест сети A и служит стартовым и одновременно выключающим местом этой сети. Переход t_1 устанавливает также начальную разметку для сети A . Аналогичную роль играет переход t_2 для сети B . Таким образом, в

Кононюк А.Е. Графы

зависимости от того, какой из переходов t_1, t_2 сработает первым, сеть C функционирует далее как сеть A или как сеть B .

Сеть A' строится по сети C добавлением перехода, который "выключает" сеть A , изымая фишку из места p_{r+2} . Это может случиться только в том случае, если вначале первым сработал переход t , и сеть C моделирует работу сети A . Аналогично сеть B' строится по сети C добавлением еще одного перехода t_4 , который выключает сеть B .

Легко видеть, что

$$R(C) = R(A) \times \{(0, 1, 0)\} \cup R(B) \times \{(0, 0, 1)\} \cup \{(0, \dots, 0, 1, 0, 0)\}.$$

где $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ — достижимые в C разметки мест

$p_{r+1}, p_{r+2}, p_{r+3}$;

$$R(A') = R(C) \cup R(A) \times \{(0, 0, 0)\};$$

$$R(B') = R(A') \cup R(B) \times \{(0, 0, 0)\} = R(C) \cup (R(A) \cup R(B)) \times \{(0, 0, 0)\}.$$

Так как разметка $M(p_{r+1}, p_{r+2}, p_{r+3}) = (0, 0, 0)$ не достижима в сети C , то

$$R(A') = R(B') \iff R(A) \subseteq R(B). \square$$

Поскольку множество достижимых разметок сети Петри характеризует, в некотором смысле, множество возможных ситуаций (состояний) в сети и в моделируемой ею дискретной системе, проблемы Я-включения и Я-эквивалентности привлекли внимание на первых же этапах исследований сетей Петри. С практической точки зрения проблема Я-эквивалентности, например, интересна потому, что разработка различных систем оптимизирующих преобразований сетей может потребовать сохранения достижимых разметок. Неразрешимость указанных проблем свидетельствует о том, что сети Петри являются математически более сложными и мощными объектами, чем, например, конечные автоматы. Более детально эти вопросы будут обсуждены ниже, в главах 3 и 5.

2.3.4. Проблемы достижимости и живости

Проблема достижимости является центральной в специальной теории сетей Петри, так как многие другие проблемы эквивалентны ей в том смысле, что их разрешимость или неразрешимость непосредственно следует из разрешимости или неразрешимости проблемы достижимости. Неоднократные попытки доказать общепринятую гипотезу о разрешимости последней, в том числе опубликованные, страдали одним общим недостатком — в доказательствах были обнаружены ошибки. Одной из работ в этом ряду является статья Майра, в которой приведено очередное доказательство разрешимости

Кононюк А.Е. Графы

проблемы достижимости. Мы ограничимся рассмотрением вопроса об эквивалентности проблемы достижимости и живости. Доказательства соответствующих теорем содержат ряд приемов, которые могут представлять интерес в других задачах анализа свойств сетей Петри. Проблема достижимости состоит в обнаружении алгоритма, с помощью которого для любой сети N и для любой разметки $M \in \mathbb{N}^{|P|}$ можно выяснить, $M \in R(N)$? Более частной представляется проблема, в которой выясняется достижимость нулевой разметки

$$\vec{0} = (\underbrace{0, \dots, 0}_{|P|}).$$

Теорема 11. *Проблема достижимости произвольной разметки сводится к проблеме достижимости нулевой разметки.*

Доказательство. Пусть задана сеть N и разметка $M \in \mathbb{N}^{|P|}$. Покажем, что N можно преобразовать в сеть N' такую, что $M \in R(N)$ тогда и только тогда, когда $0 \in R(N')$. Соответствующее преобразование показано на рис. 11.

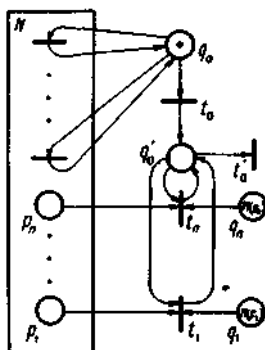


Рис. 11.

Пока место q_0 содержит фишку, сеть N' функционирует точно так же, как и сеть N . После срабатывания перехода t_0 место q'_0 получает фишку. Каждое из мест q_i , $1 \leq i \leq n$, $n = |P|$, имеет в сети N' начальную разметку $M_0(q_i) = M(p_i)$. Из этого следует, что сеть N может достигнуть нулевой разметки только в том случае, если все места q_1, \dots, q_n и все места p_1, \dots, p_n лишатся фишек. В свою очередь это возможно только при достижении сетью N разметки M . Переход t'_0 завершает "очистку" сети N' от фишек, убирая фишку из места q'_0 . Поэтому решение проблемы достижимости произвольной разметки в сети N можно свести к решению проблемы достижимости нулевой разметки в сети N' .

Кононюк А.Е. Графы

Теорема 12. *Проблема достижимости разметки сводится к проблеме живости сети.*

Доказательство. В силу теоремы 11 достаточно показать сводимость проблемы достижимости нулевой разметки к проблеме живости. В свою очередь для этого достаточно показать, что для произвольной сети N можно построить сеть N' такую, что N' жива тогда и только тогда, когда в сети N не достижима нулевая разметка. Построение N' показано на рис. 12.

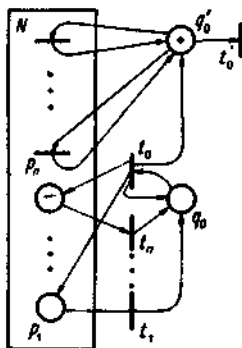


Рис. 12.

Сеть N' функционирует следующим образом. До срабатывания переходов t_0 или любого перехода t_i , $1 \leq i < n$, сеть N' работает точно так же как и сеть N . Если t_0 сработает раньше, чем любой из переходов t_i , $1 \leq i \leq n$, то возможны два случая. В первом случае, если текущая разметка подсети N является нулевой, вся сеть N останавливается (с нулевой разметкой). Во втором случае, если хотя бы одно из мест p_i , $1 \leq i \leq n$, содержит хотя бы одну фишку, может сработать переход t_i и место q_0 получает фишку, которая не может далее исчезнуть из него. Тогда переход t_0 постоянно готов срабатывать и снабжать места сети N каким угодно числом фишек, т.е. подсеть N останавливается и может породить произвольную последовательность срабатываний. Отсюда следует, что для любого перехода t сети N' t -тупиковая разметка должна иметь нулевую проекцию на места p_1, \dots, p_n сети N . Наоборот, если в N достижима нулевая разметка, то в сети N' переходы t_0, t_i , $1 \leq i \leq n$, являются потенциально мертвыми. Таким образом, сеть N' жива тогда и только тогда, когда в N не достижима нулевая разметка.

Для доказательства сводимости проблемы живости к проблеме достижимости рассмотрим более детально свойства t -тупиковых разметок.

Кононюк А.Е. Графы

Пусть D_t — множество всех t -тупиковых разметок сети Петри с n местами.

Лемма 4. *Множество D_t монотонно.*

Доказательство. Пусть $M_1 \in D_t$ и $M_2 \leq M_1$. Предположим, что $M_2 \notin D_t$, т.е. существуют разметка $M_3 \in R(N, M_2)$ и последовательность срабатываний τ такие, что $M_2 \xrightarrow{\tau} M_3$, и переход t может сработать при M_3 . Но, по теореме 1 раздел 2.1, переход t может также сработать при разметке M_4 такой, что

$$M_1 \xrightarrow{\tau} M_4 \text{ и } M_4 = M_3 + |M_2 - M_1|, \text{ т.е. } M_1 \notin D_t,$$

что противоречит начальному предположению.

Лемма 5. *Множество \tilde{D}_t максимальных элементов множества D_t конечно для любой сети Петри и может быть эффективно построено.*

Доказательство. Из определений множества D_t и его множества максимальных элементов следует, что \tilde{D}_t является множеством несравнимых по отношению \leq векторов из \mathbb{N}^n . В силу леммы 2 множество \tilde{D}_t , конечно. Его можно построить следующим способом. Зафиксируем множество $\{0, \omega\}^n$ всех векторов длины n , составленных из 0 и ω . В этом множестве выделим подмножество X_t максимальных t -тупиковых векторов:

$$X_t = \{M \in \{0, \omega\}^n \mid M \text{ — } t\text{-тупиковый и } \forall M' : M' > M \Rightarrow M' \text{ — } t\text{-тупиковый}\}.$$

Ясно, что множество X_t является конечным множеством несравнимых по отношению \leq t -тупиковых векторов, и оно может быть эффективно построено, так как по теореме 4 можно для любого вектора из $\{0, \omega\}^n$ выяснить, является ли он t -тупиковым. Если любая нулевая компонента вектора из X_t заменить на ω , то получится вектор, не принадлежащий X_t , так как он не будет t -тупиковым. Таким образом, для любого вектора $M \in X_t$ существует вектор $M' = M + K(M)$, где $K(M) \in \mathbb{N}^n$, причем M' является t -тупиковым, но любой вектор $M'' > M'$ уже не является t -тупиковым. Для каждого $M \in X_t$ вектор $K(M)$ может быть найден последовательной проверкой на t -тупиковость векторов

$$M, M + K_1, M + K_2, \dots, M + K(M),$$

где $0 \leq K_1 \leq K_2 \leq \dots \leq K(M)$ — конечная цепь векторов из \mathbb{N}^n .

Множество \tilde{D}_t можно представить как $\{M + K(M) \mid M \in X_t\}$, откуда следует, что его можно эффективно построить с помощью описанной процедуры поиска вектора $K(M)$ для каждого $M \in X_t$.

Теорема 13. *Проблема живости сети Петри сводится к проблеме достижимости некоторой разметки.*

Кононюк А.Е. Графы

Доказательство. Пусть известно, что разметка M достижима в сети N . Тогда, если для произвольного перехода t в множестве \tilde{D}_t существует вектор M' такой, что $M \leq M'$, то переход t не является живым. Наоборот, если известно, что t не является живым переходом, то существует t -тупиковая разметка $M \in R(N)$ такая, что $\exists M' \in \tilde{D}_t: M \leq M'$. Таким образом, t является потенциально мертвым, если и только если $\exists M' \in \tilde{D}_t, \exists M \in R(N): M \leq M'$. Поскольку множество \tilde{D}_t , конечно и может быть эффективно построено, то для установления того факта, что переход t является потенциально мертвым, достаточно установить достижимость разметки $M \leq M' \in D_t$. Так как число переходов в сети N конечно, то живость сети можно установить, проверяя для каждого перехода, является ли он живым или потенциально мертвым.

Теорема 14. *Проблема живости сети Петри и проблема достижимости в ней произвольной разметки эквивалентны.*

Доказательство. Прямое следствие теорем 12 и 13.

2.4 Анализ сетей Петри

Ранее была продемонстрирована моделирующая мощьность сетей Петри. С помощью сетей Петри можно моделировать широкий класс систем, представляя должным образом взаимодействие различных процессов, которые могут возникнуть. Наиболее сильны сети Петри при моделировании систем, включающих параллельные действия, причем параллельность моделируется естественным и удобным образом. Сети Петри можно использовать для представления и сообщения о проекте параллельной системы.

Однако само моделирование малополезно. Необходимо провести анализ моделируемой системы. Этот анализ, можно надеяться, приведет к глубокому проникновению в поведение моделируемой системы. Таким образом, мы подошли к необходимости рассмотрения методов анализа сетей Петри. Много методов анализа уже разработано, однако в области анализа сетей Петри существует еще много проблем. Для того чтобы лучше оценить некоторые разработанные методы анализа, рассмотрим прежде всего типы задач, которые требуют решения. Цель анализа сети Петри — получение ответа на вопрос о конкретной сети Петри. Какие же вопросы о сети Петри можно задать?

2.4.1. Задачи анализа сетей Петри

В ряде публикаций по сетям Петри рассмотрены следующие свойства и вопросы. (Здесь мы определим и проиллюстрируем эти свойства, а во второй части главы приведем соответствующие методы анализа.)

2.4.1.1. Безопасность

Одно из важнейших свойств сети Петри, которая должна моделировать реальное устройство, — *безопасность*. Позиция сети Петри является безопасной, если число фишек в ней никогда не превышает 1. Сеть Петри *безопасна*, если безопасны все позиции сети.

Определение 1. Позиция $p_i \in P$ сети Петри $C = (P, T, I, O)$ с начальной маркировкой μ является безопасной, если $\mu'(p_i) \leq 1$ Для любой $\mu' \in R(C, \mu)$. Сеть Петри безопасна, если безопасна каждая ее позиция.

Безопасность — очень важное свойство для устройств аппаратного обеспечения. Если позиция безопасна, то число фишек в ней равно 0 или 1. Следовательно, позицию можно реализовать одним триггером. В первоначальном определении сети Петри были безопасны, поскольку переход не мог быть запущен, если не все из выходных позиций были пусты (а кратные дуги не были разрешены). Это объяснялось интерпретацией позиции как условия. Условие, будучи логическим высказыванием, либо истинно (представляется фишкой в позиции), либо ложно (представляется отсутствием фишки); кратные фишки не имеют никакой интерпретации. Таким образом, если интерпретировать сети как условия и события, маркировка каждой позиции должна быть безопасной.

Если позиция не является кратной входной или кратной выходной для перехода, ее можно сделать безопасной. К позиции p_i , которую необходимо сделать безопасной, добавляется новая позиция p'_i . Переходы, в которых p_i используется в качестве входной или выходной, модифицируются следующим образом:

Если $p_i \in I(t_j)$ и $p_i \notin O(t_j)$, тогда добавить p'_i к $O(t_j)$.

Если $p_i \in O(t_j)$ и $p_i \notin I(t_j)$, тогда добавить p'_i к $I(t_j)$.

Цель введения этой новой позиции p'_i — представить условие « p_i пуста». Следовательно, p_i и p'_i дополнительные; p_i имеет фишку, только если p'_i - не имеет фишки и наоборот. Любой переход, удаляющий фишку из p_i , должен помещать фишку в p'_i , а всякий переход, удаляющий фишку из p'_i , должен помещать фишку в p_i . Начальная

Кононюк А.Е. Графы

маркировка также должна быть модифицирована для обеспечения того, чтобы точно одна фишка была либо в p_i , либо в p'_i . (Мы допускаем, что начальная маркировка безопасна.) Заметим, что такая принудительная безопасность возможна только для позиций, которые в начальной маркировке являются безопасными и входная и выходная кратность которых равна 0 или 1 для всех переходов. Позиция, имеющая для некоторого перехода выходную кратность 2, будет получать при его запуске две фишки и, следовательно, не может быть безопасной. Простая сеть Петри на рис. 1 преобразована в безопасную, как показано на рис. 2.

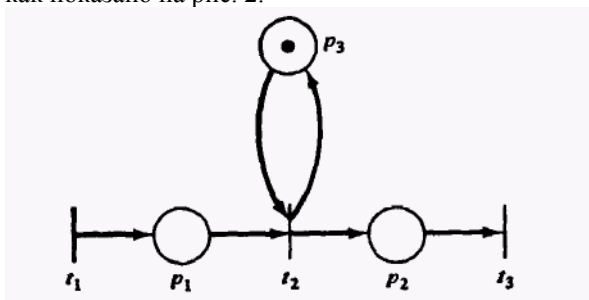


Рис. 1. Сеть Петри, не являющаяся безопасной.

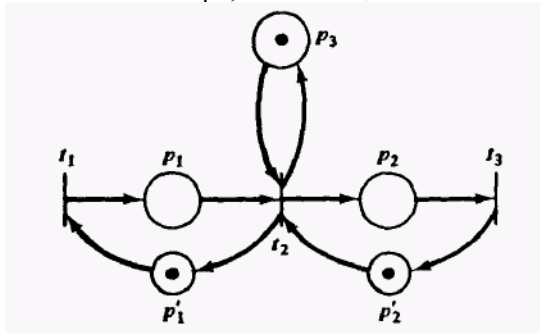


Рис. 2. Безопасная сеть Петри, эквивалентная сети, приведенной на рис. 1.

2.4.1.2. Ограниченность

Безопасность — это частный случай более общего свойства *ограниченности*. Некоторые соображения относительно реального ограничения на аппаратную реализацию позиций позволяют прийти к заключению, что безопасность — необязательное требование. Безопасность позволяет реализовать позицию триггером, но в более

общем случае можно использовать счетчик. Однако любой аппаратно-реализованный счетчик ограничен по максимальному числу, которое он может представить. Позиция является *k-безопасной* или *k-ограниченной*, если количество фишек в ней не может превышать целое число k .

Определение 2. Позиция $\mu \in R$ сети Петри $C = (P, T, I, O)$ с начальной маркировкой μ является *k-безопасной*, если $\mu'(p_i) \leq k$ для всех $\mu' \in R(C, \mu)$.

1-безопасная позиция называется просто безопасной. Заметим, что граница k' по числу фишек, которые могут находиться в позиции, может быть функцией от позиции (например, позиция p_1 может быть 3-безопасной, тогда как позиция p_2 — 8-безопасной). Однако, если позиция p_i *k-безопасна*, то она также и k' -безопасна для всех $k' \geq k$. Поскольку число позиций конечно, можно выбрать k , равное максимуму из границ каждой позиции, и определить сеть Петри *k-безопасной*, если каждая позиция сети *k-безопасна*.

Иногда нас будет интересовать только то, является число фишек в позиции ограниченным или нет, а не конкретное значение границы. Позиция называется *ограниченной*, если она *k-безопасна* для некоторого k ; сеть Петри ограничена, если все ее позиции ограничены. Ограниченную сеть Петри можно реализовать аппаратно, тогда как сеть Петри с неограниченными позициями в общем случае реализовать аппаратно нельзя. (Вспомним, что эти определения не зависят от интерпретации. В реализации позиция может представлять некоторый объект, являющийся ограниченным, хотя сама структура сети не отражает этот факт.)

2.4.1.3. Сохранение

Сети Петри можно использовать для моделирования систем распределения ресурсов. Например, сеть Петри может моделировать запросы, распределения и освобождения устройств ввода/вывода в вычислительной системе. В этих системах некоторые фишки могут представлять ресурсы. Группа из трех построчно печатающих устройств представляется позицией, имеющей в начальной маркировке три фишки. Запрос построчно-печатающего устройства — это переход, для которого данная позиция является входной; затем устройство освобождается переходом, для которого позиция построчно печатающих устройств является выходной.

Для сетей Петри такого типа помимо прочих важным свойством является сохранение. Нам бы хотелось показать, что фишки, представляющие ресурсы, никогда не создаются и не уничтожаются.

Кононюк А.Е. Графы

Простейший способ сделать это — потребовать, чтобы общее число фишек в сети оставалось постоянным.

Определение 3. Сеть Петри $C = (P, T, I, O)$ с начальной маркировкой μ называется *строго сохраняющей*, если для всех $\mu' \in R(C, \mu)$

$$\sum_{p_i \in P} \mu'(p_i) = \sum_{p_i \in P} \mu(p_i).$$

Строгое сохранение — это очень сильное ограничение. Например, из него немедленно следует, что число входов в каждый переход должно равняться числу выходов, $|I(t_j)| = |O(t_j)|$. Если бы это было не так, запуск перехода изменил бы число фишек в сети.

Однако для более общего представления о свойстве сохранения рассмотрим рис. 3.

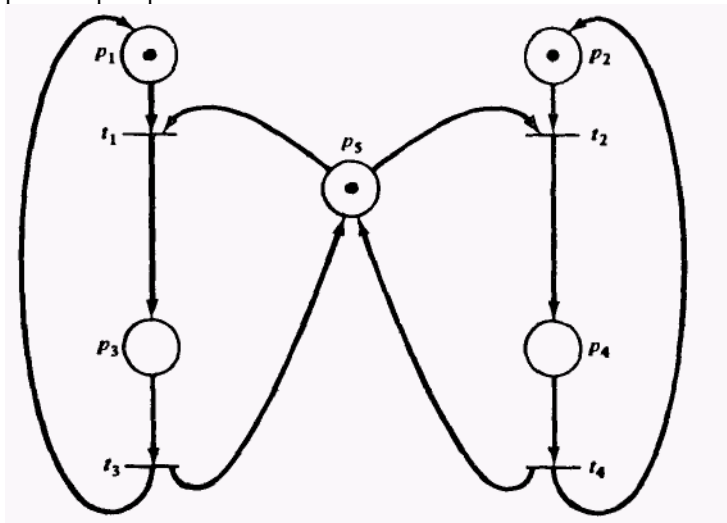


Рис. 3. Сеть Петри, не являющаяся строго сохраняющей.

Изображенная на нем сеть Петри не является строго сохраняющей, поскольку запуск перехода t_1 или t_2 уменьшит число фишек на 1, а запуск перехода t_3 или t_4 добавит фишку к маркировке. Мы можем тем не менее преобразовать эту сеть Петри в сеть на рис.4, являющуюся строго сохраняющей.

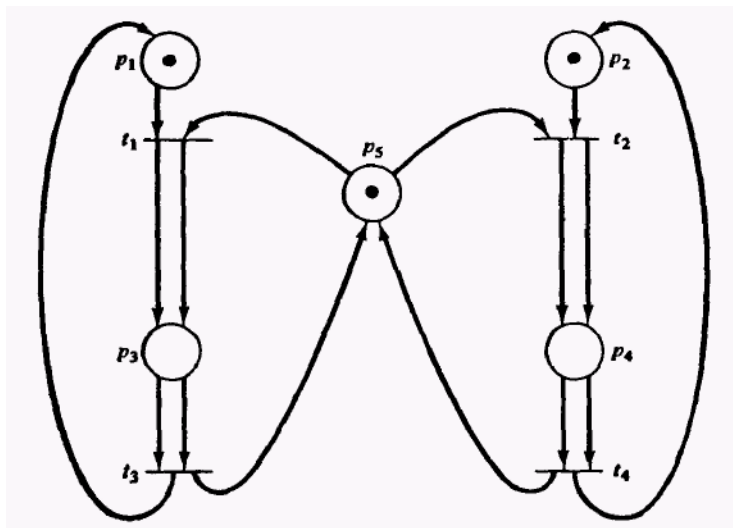


Рис. 4. Строго сохраняющая сеть Петри, эквивалентная сети, изображенной на рис. 3.

Сеть Петри должна сохранять ресурсы, которые она моделирует. Однако взаимно однозначного соответствия между фишками и ресурсами нет. Фишка может представлять и один программный счетчик или какой-нибудь другой элемент и может представлять несколько ресурсов сразу. Во втором случае фишка позже используется для создания кратных фишек (по одной на ресурс) путем запуска перехода с большим числом выходов, чем входов. В общем следует определить *взвешивание* фишек. *Взвешенная* сумма для всех достижимых маркировок должна быть постоянной. Фишкам, не являющимся важными, можно присвоить вес 0; другим фишкам можно присвоить веса 1,2,3 или любое другое целое. (Допустимы рациональные веса, поскольку для определения целого взвешивания такое взвешивание можно умножить на общее кратное. Иррациональные веса не представляются необходимыми.)

Фишка определяется ее позицией в сети, все фишки в позиции неразличимы. Следовательно, веса связываются с каждой позицией сети. Вектор взвешивания $w = (w_1, w_2, \dots, w_n)$ определяет вес w_i для каждой позиции $p_i \in P$.

Кононюк А.Е. Графы

Определение 4. Сеть Петри $C = (P, T, I, O)$ с начальной маркировкой μ называется сохраняющей по отношению к вектору взвешивания w , $w = (w_1, w_2, \dots, w_n)$, $n = |P|$, $w_i \geq 0$, если для всех $\mu' \in R(C, \mu)$

$$\sum_i w_i \cdot \mu'(p_i) = \sum_i w_i \cdot \mu(p_i).$$

Строго сохраняющая сеть Петри является сохраняющей по отношению к вектору взвешивания $(1, 1, \dots, 1)$. Все сети Петри являются сохраняющими по отношению к вектору взвешивания $(0, 0, \dots, 0)$. Этот факт вносит в теорию некоторую нестройность, поскольку нам бы хотелось определить сеть Петри как сохраняющую, если она является сохраняющей по отношению к некоторому вектору взвешивания. Однако, так как всякая сеть Петри является сохраняющей по отношению к нулевому вектору, такое определение не является удовлетворительным. Таким образом, сеть Петри называется *сохраняющей*, если она сохраняющая по отношению к некоторому положительному ненулевому вектору взвешивания $w > 0$ (с положительными ненулевыми весами, $w_i > 0$).

Сеть Петри с рис. 3 является поэтому сохраняющей, поскольку она сохраняющая по отношению к $(1, 1, 2, 2, 1)$. Сеть Петри с рис.5 не является сохраняющей.

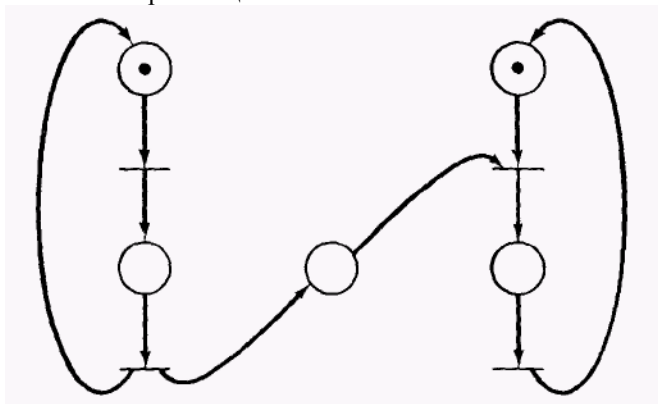


Рис. 5. Несохраниющая сеть Петри.

2.4.1.4. АКТИВНОСТЬ

Причиной рассмотрения сохранения в сети Петри было распределение ресурсов в операционной системе ЭВМ. Другая задача, которая может возникнуть при распределении ресурсов вычислительной системы — тупики. Тупики служат предметом многих исследований в области

Кононюк А.Е. Графы

вычислительной техники. Лучше всего иллюстрирует задачу простой пример. Рассмотрим систему, включающую два различных ресурса q и r и два процесса a и b . Если оба процесса нуждаются в обоих ресурсах, им необходимо будет совместно использовать ресурсы. Для выполнения этого потребуем, чтобы каждый процесс запрашивал ресурс, а затем освобождал его. Теперь предположим, что процесс a сначала запрашивает ресурс q , затем ресурс r и, наконец, освобождает и q , и r . Процесс b работает аналогично, но сначала запрашивает r , а затем q . Сеть Петри на рис. 6 иллюстрирует два процесса и распределение ресурсов между ними.

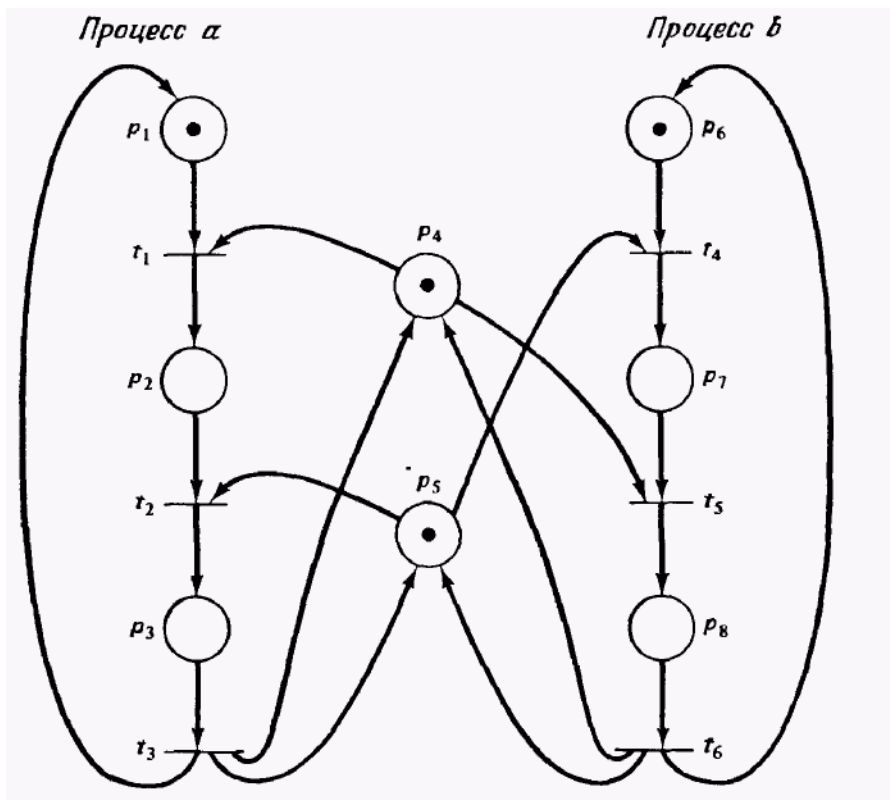


Рис. 6. Распределение ресурсов для случая двух процессов (a и b) и двух ресурсов (q (моделируется p_4) и r (моделируется p_5)).

Начальная маркировка помечает ресурсы $q(p_4)$ и $r(p_4)$ доступными и указывает на готовность процессов a и b . Одним выполнением этой

Кононюк А.Е. Графы

сети является $t_1t_2t_3t_4t_5t_6$; другим — $t_4t_5t_6t_1t_2t_3$. Ни одно из этих выполнений не приводит к тупику. Однако рассмотрим последовательность, которая начинается переходами t_1, t_4 : процесс a обладает ресурсом q и хочет получить r , процесс b обладает r и хочет получить q . Система заблокирована; никакой процесс продолжаться не может.

Тупик в сети Петри — это переход (или множество переходов), которые не могут быть запущены. В сети Петри на рис. 6. тупик возникает, если нельзя запустить переходы t_2 и t_5 . Переход называется *активным*, если он не заблокирован (нетупиковый). Это не означает, что переход разрешен, скорее он может быть разрешенным. Переход t_j сети Петри S называется *потенциально запусимым* в маркировке μ , если существует маркировка $\mu' \in R(C, \mu)$, в которой t_j разрешен. Переход активен в маркировке μ , если потенциально запустим во всякой маркировке из $R(C, \mu)$. Следовательно, если переход активен, то всегда возможно перевести сеть Петри из ее текущей маркировки в маркировку, в которой запуск перехода станет разрешенным.

Существуют другие, связанные с активностью понятия, которые рассматривались при изучении тупиков. Их можно разбить на категории по *уровню* активности и определить для сети Петри S с маркировкой μ следующим образом:

Уровень 0: Переход t_j обладает *активностью уровня 0*, если он никогда не может быть запущен.

Уровень 1: Переход t_j обладает *активностью уровня 1*, если он потенциально запусим, т. е. если существует такая $\mu' \in R(C, \mu)$, что t_j разрешен в μ' .

Уровень 2: Переход t_j обладает *активностью уровня 2*, если для всякого целого n существует последовательность запусков, в которой t_j присутствует по крайней мере n раз.

Уровень 3: Переход t_j обладает *активностью уровня 3*, если существует бесконечная последовательность запусков, в которой t_j присутствует неограниченно часто.

Уровень 4: Переход t_j обладает *активностью уровня 4*, если для всякой $\mu' \in R(C, \mu)$ существует такая последовательность запусков σ , что t_j разрешен в $\delta(\mu', \sigma)$.

Переход, обладающий активностью уровня 0, называется *пассивным*. Переход, обладающий активностью уровня 4, называется *активным*. Сеть Петри обладает активностью уровня i , если каждый ее переход обладает активностью уровня i .

В качестве примера, иллюстрирующего уровни активности, рассмотрим сеть Петри на рис. 7. Переход t_0 не может быть запущен никогда; он пассивен.

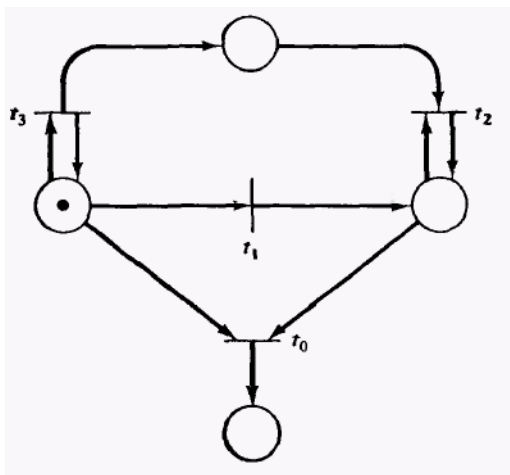


Рис. 7. Сеть Петри, иллюстрирующая различные уровни активности.

Переход t_1 можно запустить точно один раз; он обладает активностью уровня 1. Переход t_2 может быть запущен произвольное число раз, но это число зависит от числа запусков перехода t_3 . Если мы хотим запустить t_2 пять раз, мы запускаем пять раз t_3 , затем t_1 и после этого пять раз t_2 . Однако, как только запустится t_1 (t_1 должен быть запущен до того, как будет запущен t_2), число возможных запусков t_2 станет фиксированным. Следовательно, t_2 обладает активностью уровня 2, но не уровня 3. С другой стороны, переход t_3 можно запускать бесконечное число раз, и поэтому он обладает активностью уровня 3, но не уровня 4, поскольку, как только запустится t_1 , t_3 больше запустить будет нельзя.

2.4.1.5. Достижимость и покрываемость

Большинство задач, к которым мы до сих пор обращались, касаются достижимых маркировок. Задача *достижимости* является, по-видимому, наиболее простой (для формулировки).

Определение 5. *Задача достижимости.* Для данной сети Петри C с маркировкой μ и маркировки μ' определить: $\mu' \in R(C, \mu)$?

Задача достижимости, быть может, основная задача анализа сетей Петри; многие другие задачи анализа можно сформулировать в терминах задачи достижимости. Например, для сети Петри с рис.6

тупик может возникнуть, если достижимым является состояние $(0, 1, 0, 0, 0, 0, 1, 0)$.

На рис. 8 показана сеть Петри, цель которой заключается в решении задачи о взаимном исключении, — предполагается, что позиции p_1 и p_9 будут взаимно исключающими.

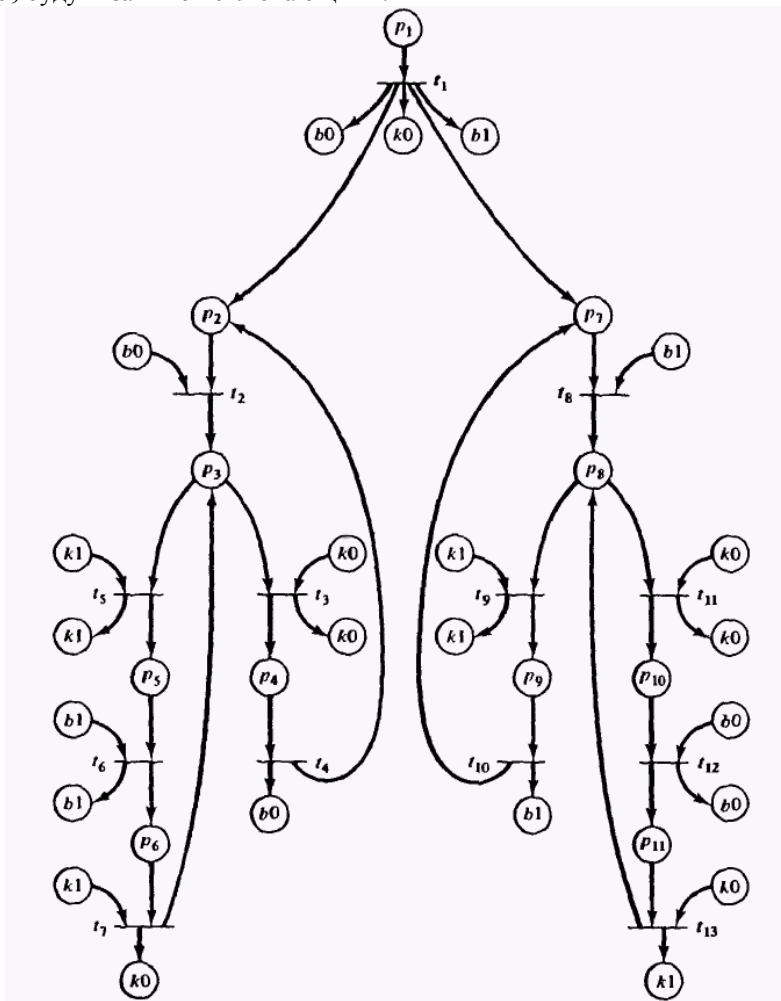


Рис. 8. Представление сетью Петри «решения» Химана задачи о взаимном исключении. Вследствие постоянного использования позиций k_0, k_1, b_1 и b_0 они для удобства продублированы в графе,

представляющей сеть. Все позиции, помеченные k_0 , являются одной позицией.

Мы хотим знать, является ли какое-либо состояние с $\mu(p_4) \geq 1$ и $\mu(p_9) \geq 1$ достижимым. Эта задача аналогична достижимости, но несколько отличается; она называется задачей покрываемости. Маркировка μ'' покрывает маркировку μ' , если $\mu'' \geq \mu'$.

Определение 6. *Задача покрываемости.* Для данной сети Петри S с начальной маркировкой μ и маркировки μ' определить, существует ли такая достижимая маркировка $\mu'' \in R(S, \mu)$, что $\mu'' \geq \mu'$.

В других возможных задачах типа достижимости могло бы игнорироваться содержимое некоторых позиций и приниматься во внимание сравнение или покрытие содержимого нескольких важных позиций. Например, в сети Петри на рис. 8 наш интерес ограничен позициями p_4 и p_9 — маркировка остальных позиций не важна. Таким образом, мы можем рассматривать достижимость и покрываемость «по модулю» множества позиций. Эти задачи называются задачами *достижимости подмаркировки* и *покрываемости подмаркировки*.

Их можно еще более усложнить требованием, определить достижимость и покрываемость для множества маркировок, придя к задачам *достижимости множества* и *покрываемости множества*. Однако, если множество конечно, задачи можно решить обычно многократным решением задач достижимости и покрываемости для одной маркировки.

2.4.1.6. Последовательности запусков

Другой, предложенный к анализу подход основан не на позициях, а на последовательностях запусков переходов, т. е. связан с активностью, поскольку уместен вопрос: может ли переход быть запущен (иначе, не является ли он пассивным)? В более общем случае можно потребовать определить, возможна ли заданная последовательность запусков переходов или возможна ли какая-либо последовательность из множества последовательностей запусков. В сети Петри на рис. 8, например, взаимное исключение было бы нарушенным, если бы могла осуществиться последовательность t_3t_9 или t_4t_{10} , или, более общо, если бы могла возникнуть последовательность $t_3\sigma t_9$, где σ — произвольная последовательность запусков, не включающая t_4 . Эти вопросы анализа приводят к понятию языков сетей Петри и будут исследованы более детально дальше.

2.4.1.7. Задачи эквивалентности и подмножества

Последний класс задач порожден соображениями оптимизации. Пусть сети Петри присуще некоторое поведение, определяемое ее множеством последовательностей запусков переходов или ее множеством достижимости. Можно ли изменить (оптимизировать) сеть Петри, не изменяя ее поведения? Изменение означает удаление пассивных переходов (которые никогда нельзя запустить), или пассивных позиций (которые никогда не могут быть маркированы), или переопределение некоторых переходов. Можно ли показать, что две различные маркированные сети Петри с одинаковым числом переходов (но с различным числом позиций) будут порождать одинаковые последовательности запусков переходов или что две различные маркированные сети Петри с одинаковым числом позиций (но с различным числом переходов) будут порождать одно множество достижимости? Это позволило бы модифицировать сети Петри для увеличения параллелизма, уменьшения стоимости реализации или улучшения других оптимизируемых функционалов.

В этих случаях мы затрагиваем проблему определения того, являются ли две сети Петри эквивалентными или является ли одна из них подмножеством другой. Для точного определения понятия эквивалентности или включения необходимо быть особенно внимательным. Если мы определим эквивалентность как равенство множеств достижимости, тогда нельзя будет изменять число позиций, если потребуем равенства множеств последовательностей — нельзя будет изменять переходы. Поэтому определение задачи, которое мы дадим, исключительно важно.

2.4.2. Методы анализа

Задачи, которые были представлены здесь, являются наиболее общими, тем не менее существует множество других задач, решение которых может потребоваться для сетей Петри. Можно ли разработать методы анализа сетей Петри для решения этих задач? Причем нас интересуют особенно те методы, которые легко реализовались бы на ЭВМ, что важно для осуществления автоматического анализа моделируемых систем.

В этом разделе мы представим два основных метода анализа сетей Петри, которые описывают механизмы решения некоторых из уже перечисленных задач. Первый метод анализа, используемый для сетей

Кононюк А.Е. Графы

Петри, — это *дерево достижимости*, второй метод связан с *матричными уравнениями*. Обсудим каждый из них.

2.4.2.1. Дерево достижимости

Дерево достижимости представляет множество достижимости сети Петри. Рассмотрим, например, маркированную сеть Петри на рис. 9.

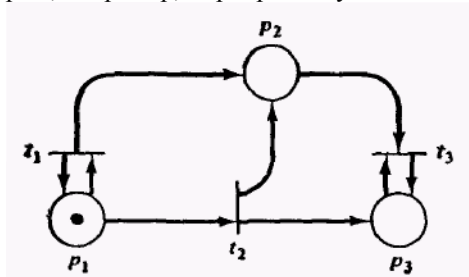


Рис. 9. Маркированная сеть Петри, для которой строится дерево достижимости.

Начальная маркировка ее — $(1, 0, 0)$. В этой начальной маркировке разрешены два перехода: t_1 и t_2 . Поскольку мы хотим рассмотреть все множества достижимости, определим новые вершины в дереве достижимости (корневой (начальной) является вершина, соответствующая начальной маркировке) для (достижимых) маркировок, получающихся в результате запуска каждого из этих двух переходов. Дуга, помеченная запускаемым переходом, приводит из начальной маркировки к каждой из новых маркировок (рис. 10).

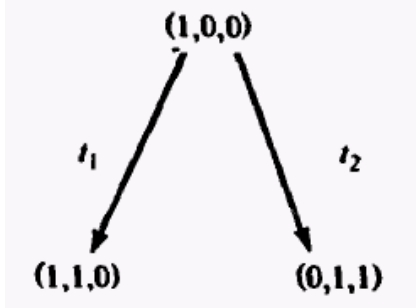


Рис. 10. Первый шаг построения дерева достижимости.

Это (частичное) дерево показывает все маркировки, непосредственно достижимые из начальной маркировки.

Теперь необходимо рассмотреть все маркировки, достижимые из новых маркировок. Из маркировки $(1, 1, 0)$ можно снова запустить t_1 (получая $1, 2, 0$) и t_2 (получая $0, 2, 1$); из $(0, 1, 1)$ можно запустить t_3 (получая $0, 0, 1$). Это порождает дерево, изображенное на рис. 11.

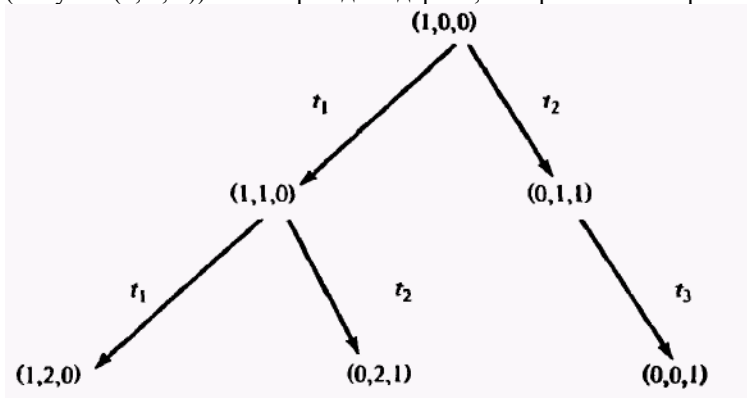


Рис. 11. Второй шаг построения дерева достижимости.

Начиная с трех новых маркировок, необходимо повторить этот процесс, порождая новые маркировки, которые нужно вводить в дерево, как показано на рис. 12.

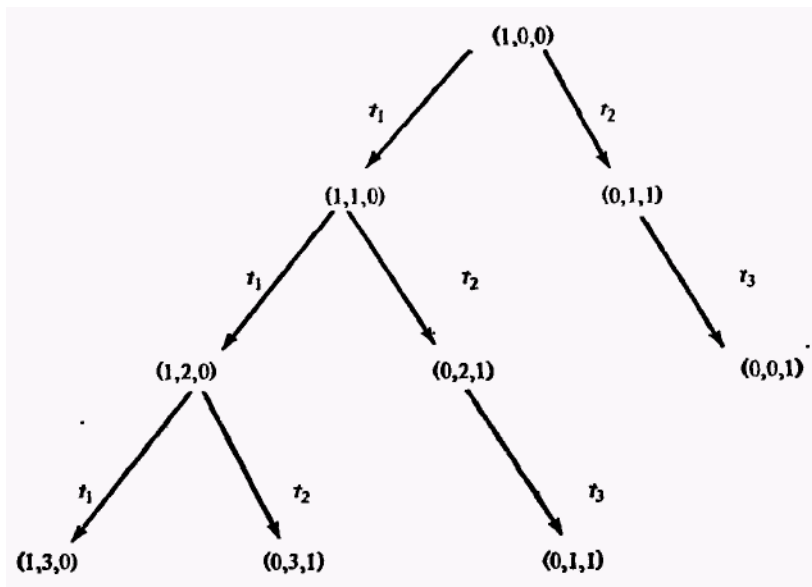


Рис. 12. Третий шаг построения дерева достижимости.

Заметим, что маркировка $(0, 0, 1)$ пассивная; никакой переход в ней не является разрешенным, поэтому никакие новые маркировки из этой пассивной маркировки в дереве порождаться не будут. Кроме того, необходимо отметить, что маркировка $(0, 1, 1)$, порождаемая запуском t_3 в $(0, 2, 1)$, была уже порождена непосредственно из начальной маркировки запуском t_2 .

Если эту процедуру повторять снова и снова, каждая достижимая маркировка окажется порожденной. Однако получившееся дерево достижимости может оказаться бесконечным. Будет порождена каждая маркировка из множества достижимости, поэтому для любой сети Петри с бесконечным множеством достижимости соответствующее дерево также должно быть бесконечным. Даже сеть Петри с конечным множеством достижимости может иметь бесконечное дерево (рис. 13).

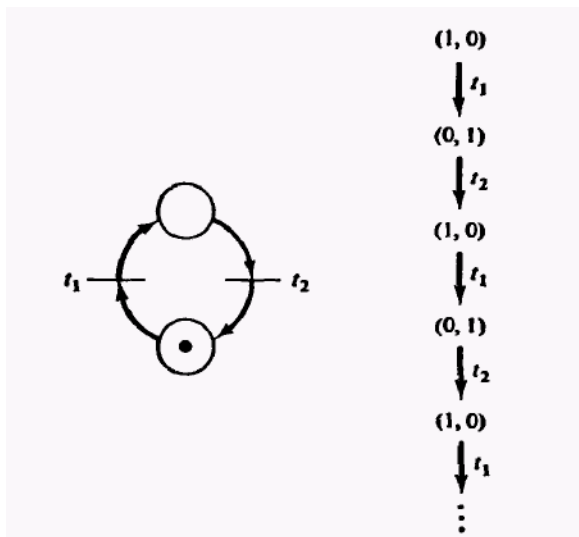


Рис. 13. Простая сеть Петри с бесконечным деревом достижимости.

Дерево представляет все возможные последовательности запусков переходов. Всякий путь в дереве, начинающийся в корне, соответствует допустимой последовательности переходов. Для превращения дерева в полезный инструмент анализа необходимо найти средства ограничения его до конечного размера. (Заметим, что если какое-то представление бесконечного множества конечно, то бесконечное множество маркировок должно отображаться в такое представление. В общем случае это приведет к потере информации и, возможно, к тому, что некоторые свойства сетей Петри определить будет нельзя, но все зависит от того, как представление было получено.)

Приведение к конечному представлению осуществляется несколькими способами. Нам необходимо найти те средства, которые ограничивают введение новых маркировок (называемых *граничными* вершинами) на каждом шаге. Здесь могут помочь пассивные маркировки — маркировки, в которых нет разрешенных переходов. Эти пассивные маркировки называются *терминальными вершинами*. Другой класс маркировок — это маркировки, ранее встречавшиеся в дереве. Такие дублирующие маркировки называются *дублирующими вершинами*; никакие последующие маркировки рассматривать нет нужды — все они будут порождены из места первого появления дублирующей маркировки в дереве. Таким образом, в дереве на рис. 12 маркировка

$(0, 1, 1)$, получившаяся в результате выполнения последовательности $t_1 t_2 t_3$, не будет порождать какие-либо новые вершины в дереве, поскольку она ранее встречалась в дереве в результате выполнения последовательности t_2 из начальной маркировки.

Для сведения дерева достижимости к конечному представлению используется еще одно средство. Рассмотрим последовательность запусков переходов σ , начинающуюся в начальной маркировке μ и кончающуюся в маркировке μ' , $\mu' > \mu$. Маркировка μ' совпадает с маркировкой μ , за исключением того, что имеет некоторые «дополнительные» фишки в некоторых позициях, т. е. $\mu' = \mu + (\mu' - \mu)$ и $(\mu' - \mu) > 0$. Теперь, поскольку на запуски переходов лишние фишки не влияют, последовательность σ можно запустить снова, начиная в μ' , приходя к маркировке μ'' . Так как действие последовательности переходов σ добавило $\mu' - \mu$ фишек к маркировке μ , она добавит также $\mu' - \mu$ фишек и к маркировке μ' , поэтому $\mu'' = \mu' + (\mu' - \mu)$ или $\mu'' = \mu + 2(\mu' - \mu)$. В общем можно запустить последовательность σ n раз, получив в результате маркировку $\mu + n(\mu' - \mu)$. Следовательно, для тех позиций, которые увеличивают число фишек последовательностью σ , можно создать произвольно большое число фишек, просто повторяя последовательность σ столько, сколько это нужно. В сети Петри на рис. 9, например, можно запустить переход t_1 столько раз, сколько необходимо для того, чтобы получить произвольное число фишек в p_2 .

Представим бесконечное число маркировок, получающихся из циклов такого типа, с помощью специального символа ω , который обозначает «бесконечность». Для любого постоянного a определим

$$\begin{array}{ll} \omega + a = \omega, & a < \omega, \\ \omega - a = \omega, & \omega \leq \omega. \end{array}$$

Для построения дерева достижимости необходимы только эти операции над ω .

Теперь можно точно сформулировать действительный алгоритм построения дерева достижимости. Каждая вершина i дерева связывается с расширенной маркировкой $\mu[i]$; в расширенной маркировке число фишек в позиции может быть либо неотрицательным целым, либо ω . Каждая вершина классифицируется или как граничная вершина, терминальная вершина, дублирующая вершина, или как внутренняя вершина. Граничными являются вершины, которые еще не обработаны алгоритмом; алгоритм превратит их в терминальные, дублирующие или внутренние вершины.

Кононюк А.Е. Графы

Алгоритм начинает с определения начальной маркировки корнем дерева, т. е. граничной вершиной. До тех пор пока имеются граничные вершины, они обрабатываются алгоритмом.

Пусть x — граничная вершина, которую необходимо обработать.

1. Если в дереве имеется другая вершина y , не являющаяся граничной, и с ней связана та же маркировка, $\mu[x] = \mu[y]$, то вершина x — *дублирующая*.

2. Если для маркировки $\mu[x]$ ни один из переходов не разрешен (т. е. $\delta(\mu[x], t_j)$ не определено для всех $t_j \in T$), то x — *терминальная вершина*.

3. Для всякого перехода $t_j \in T$, разрешенного в $\mu[x]$ (т. е. $\delta(\mu[x], t_j)$ определено), создать новую вершину z дерева достижимости. Маркировка $\mu[z]$, связанная с этой вершиной, определяется для каждой позиции p_i следующим образом:

а) Если $\mu[x]_{p_i} = \omega$, то $\mu[z]_{p_i} = \omega$.

б) Если на пути от корневой вершины к x существует вершина y с $\mu[y]_{p_i} < \delta(\mu[x], t_j)_{p_i}$ и $\mu[y]_{p_i} < \delta(\mu[x], t_j)_{p_i}$, то $\mu[z]_{p_i} = \omega$.

в) В противном случае $\mu[z]_{p_i} = \delta(\mu[x], t_j)_{p_i}$.

Дуга, помеченная t_j , направлена от вершины x к вершине z . Вершина x переопределяется как внутренняя, вершина z становится граничной.

Когда все вершины дерева — терминальные, дублирующие или внутренние, алгоритм останавливается.

На рис. 14 представлено дерево достижимости для сети Петри на рис. 9.

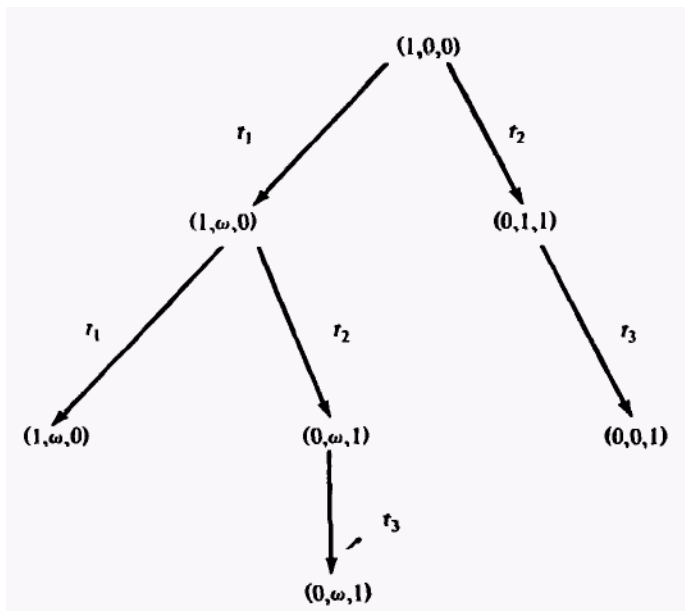


Рис. 14. Дерево достижимости сети Петри, приведенной из рис. 9.

Дерево достижимости сети Петри с рис. 15 изображено на рис. 16.

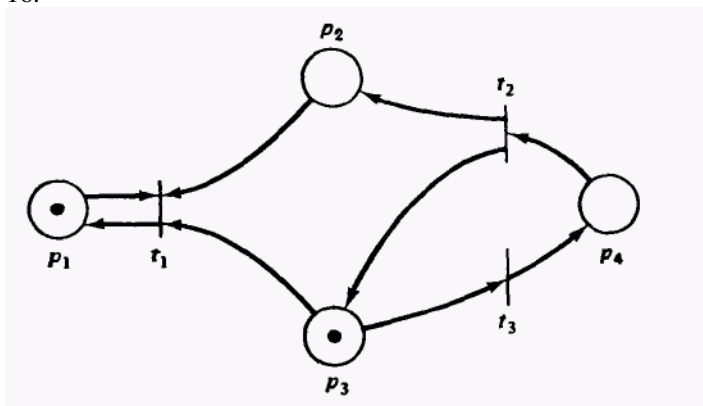


Рис. 15. Сеть Петри, для которой строится дерево достижимости.

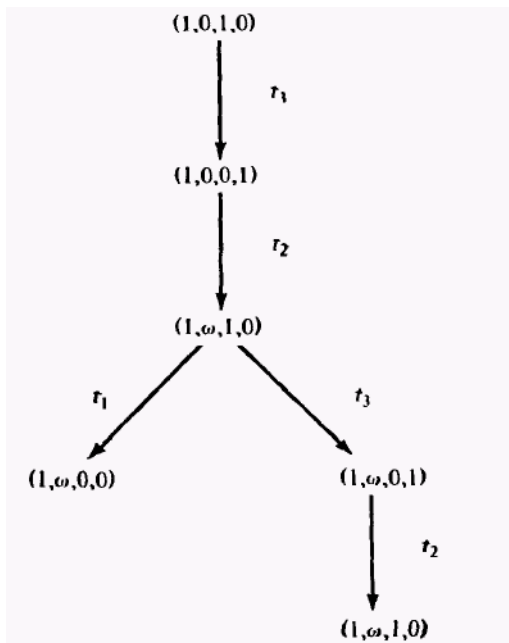


Рис. 16. Дерево достижимости сети Петри, изображенной на рис. 15.

Очень важным свойством алгоритма построения дерева достижимости является то, что он заканчивает работу. Для доказательства этого мы должны показать, что алгоритм не может создавать новые граничные вершины бесконечно. Доказательство основано на трех леммах.

Лемма 1. В любом бесконечном направленном дереве, в котором всякая вершина имеет только конечное число непосредственно последующих вершин, существует бесконечный путь, исходящий из корня.

Доказательство. Начнем с корня в вершине x_0 . Поскольку имеется только конечное число непосредственно следующих за x_0 вершин, но общее число вершин в дереве бесконечно, по крайней мере одна из непосредственно следующих за x_0 вершин должна быть корнем бесконечного поддерева. (Если бы все поддеревья с корнями, непосредственно следующими за x_0 , были конечными, то и дерево с корнем x_0 было бы конечным.) Выберем вершину x_1 , непосредственно следующую за x_0 и являющуюся корнем бесконечного поддерева. Теперь одна из непосредственно следующих за ней вершин также является корнем бесконечного поддерева, выберем в качестве такой

Кононюк А.Е. Графы

вершины x_2 . Продолжая таким же образом, получим бесконечный путь в дереве x_0, x_1, x_2, \dots .

Лемма 2. Всякая бесконечная последовательность неотрицательных целых содержит бесконечную неубывающую подпоследовательность.

Доказательство. Возможны два случая:

1. Если какой-либо элемент последовательности встречается бесконечно часто, то пусть x_0 является таким элементом. Бесконечная подпоследовательность x_0, x_0, x_0, \dots является бесконечной неубывающей подпоследовательностью.

2. Если никакой элемент не встречается бесконечно часто, тогда всякий элемент встречается только конечное число раз. Пусть x_0 — произвольный элемент последовательности. Существует самое большее x_0 целых, неотрицательных и меньших, чем x_0 ($0, \dots, x_0 - 1$), причем каждый из них присутствует в последовательности только конечное число раз. Следовательно, продвигаясь достаточно долго по последовательности, мы должны встретить элемент $x_1, x_1 \geq x_0$. Аналогично должен существовать в последовательности $x_2, x_2 \geq x_1$ и т. д. Это определяет бесконечную неубывающую подпоследовательность x_0, x_1, x_2, \dots .

В обоих случаях бесконечная неубывающая последовательность существует.

Лемма 3. Всякая бесконечная последовательность n -векторов над расширенными неотрицательными целыми (неотрицательные целые плюс символ ω) содержит бесконечную неубывающую подпоследовательность.

Доказательство. Доказываем индукцией по n , где n — размерность векторного пространства.

1. Базовый случай ($n = 1$). Если в последовательности имеется бесконечное число векторов (ω), то они образуют бесконечную неубывающую последовательность. В противном случае бесконечная последовательность, образованная удалением конечного числа векторов (ω), имеет по лемме 2 бесконечную неубывающую подпоследовательность.

2. Индуктивное предположение. (Допустим, что лемма верна для n , докажем ее справедливость для $n + 1$.) Рассмотрим первую координату. Если существует бесконечно много векторов, имеющих в качестве первой координаты ω , тогда выберем эту бесконечную подпоследовательность, которая не убывает (постоянна) по первой координате. Если только конечное число векторов имеют ω в качестве первой координаты, то рассмотрим бесконечную последовательность целых, являющихся значениями первых координат. По лемме 2 эта последовательность имеет бесконечную неубывающую

Кононюк А.Е. Графы

подпоследовательность. Она определяет бесконечную подпоследовательность векторов, которые не убывают по своей первой координате.

В любом случае мы имеем последовательность векторов, неубывающих по первой координате. Применим индуктивное предположение к последовательности n -векторов, которая получается в результате отбрасывания первой компоненты $n+1$ -векторов. Полученная таким образом бесконечная подпоследовательность является неубывающей по каждой координате.

Докажем следующую теорему.

Теорема 1. Дерево достижимости сети Петри конечно.

Доказательство. Доказательство проводим от противного. Допустим, что существует бесконечное дерево достижимости. Тогда по лемме 1 в нем имеется бесконечный путь x_0, x_1, x_2, \dots , исходящий из корня x_0 . (Число вершин, следующих за каждой вершиной в дереве, ограничено числом переходов m .) Тогда $\mu[x_0], \mu[x_1], \mu[x_2], \dots$ — бесконечная последовательность n -векторов над $\mathbb{N} \cup \{\omega\}$, а по лемме 3 она имеет бесконечную неубывающую подпоследовательность $\mu[x_{i_0}] \leq \mu[x_{i_1}] \leq \mu[x_{i_2}] \dots$. Но по построению мы не можем иметь $\mu[x_i] = \mu[x_j]$, поскольку тогда одна из вершин была бы дублирующей и не имела следующих за собой вершин. Следовательно, мы имеем бесконечную строго убывающую последовательность $\mu[x_{i_0}] < \mu[x_{i_1}] < \dots$. Но по построению, так как $\mu[x_i] < \mu[x_j]$, следовало бы заменить по крайней мере одну компоненту $\mu[x_i]$, не являющуюся ω , на ω в $\mu[x_j]$. Таким образом, $\mu[x_{i_1}]$ имеет по крайней мере одну компоненту, являющуюся ω , $\mu[x_{i_2}]$ имеет по крайней мере две ω -компоненты, а $\mu[x_{i_n}]$ имеет по крайней мере n ω -компонент.

Поскольку маркировки n -мерные, $\mu[x_{i_n}]$ имеет во всех компонентах ω .

Но тогда $\mu[x_{i_{n+1}}]$ не может быть больше $\mu[x_{i_n}]$. Пришли к противоречию, что доказывает, что наше допущение относительно существования бесконечного дерева достижимости неверно.

Построение дерева достижимости было впервые описано Карпом и Миллером. Используемый здесь вариант алгоритма был предложен Келлером. Доказательство конечности, данное здесь, взято у Хэка, который принял за основу доказательство Карпа и Миллера.

Дерево достижимости — очень полезный инструмент анализа сетей Петри. В последующих разделах мы покажем, как его можно

использовать для решения некоторых задач, представленных в разд. 2.4.1.

2.4.2.2. Безопасность и ограниченность

Сеть Петри безопасна, если число фишек в каждой позиции не может превысить 1; сеть Петри ограничена, если существует такое целое k , что число фишек в любой позиции не может превысить k . Оба этих свойства можно проверить с помощью дерева достижимости. *Сеть Петри ограничена тогда и только тогда, когда символ ω отсутствует в ее дереве достижимости.* Присутствие символа ω в дереве достижимости означает, что число фишек потенциально не ограничено; существует последовательность запусков переходов, которую можно повторить произвольное число раз, увеличивая количество фишек до произвольно большого числа. Таким образом, если имеется символ ω , сеть неограничена. Кроме того, положение символа ω показывает, какие позиции неограниченны.

Обратное утверждение также является верным, если сеть Петри неограничена, то число достижимых маркировок бесконечно. Поскольку дерево достижимости конечно, бесконечное число достижимых маркировок отражает присутствие символа ω .

Если сеть Петри ограничена и символ ω отсутствует в дереве достижимости, то сеть Петри представляет систему конечных состояний. Дерево достижимости, по существу, является графом состояний и будет содержать вершину, соответствующую всякой достижимой маркировке. Это позволяет решить вопросы анализа простым перебором и проверкой конечного множества всех достижимых маркировок. Например, чтобы определить границу для заданной позиции, нужно построить дерево достижимости и найти наибольшее значение компоненты маркировки, соответствующей этой позиции. Найденное значение является границей числа фишек для заданной позиции. Если граница для всех позиций равна 1 (точнее, не превышает 1), сеть безопасна.

На рис. 17 демонстрируется использование дерева достижимости для определения ограниченности.

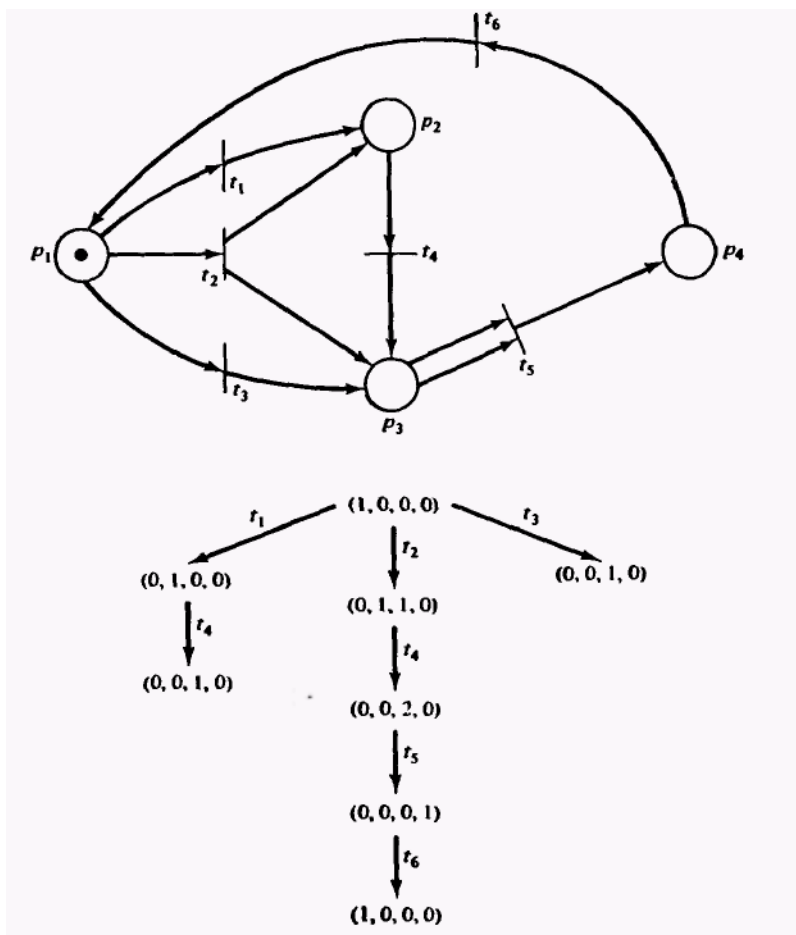


Рис. 17. Определение ограниченности для сети Петри с помощью дерева достижимости.

Отметим, что по дереву достижимости даже для сетей Петри, не являющихся ограниченными (вследствие неограниченности некоторой позиции), можно определить границы для тех позиций, которые являются ограниченными. Таким образом, дерево достижимости позволяет элективно решить задачи анализа сетей Петри по определению ограниченности и безопасности для отдельных позиций и целых сетей.

2.4.2.3. Сохранение

Сеть Петри является *сохраняющей*, если она не теряет и не порождает фишки, а просто передвигает их. Поскольку две фишки можно закодировать как одну фишку, которая позже вызовет запуск перехода, создающего две фишки, значение фишки в каждой позиции определяет вектор взвешивания, веса неотрицательны. Сеть Петри является сохраняющей по отношению к вектору взвешивания, если взвешенная сумма фишек постоянна для всех достижимых маркировок.

Свойство сохранения эффективно проверяется с помощью дерева достижимости. Так как дерево достижимости конечно, для каждой маркировки можно вычислить взвешенную сумму. Если сумма одинакова для каждой достижимой маркировки, сеть — сохраняющая по отношению к данному весу. Если суммы не равны, сеть — несохраняющая.

При оценке сохранения необходимо быть внимательным с символом ω . Если маркировка имеет ω в качестве маркировки позиции p_i , тогда для того, чтобы сеть была сохраняющей, вес этой позиции должен быть равным 0. Напомним, что символ ω представляет бесконечное множество значений. Так как все веса неотрицательны, вес должен равняться либо нулю (тем самым означая, что число фишек в данной позиции не важно), либо быть положительным. Если вес положителен, то сумма будет разной для двух маркировок, различающихся в своей ω -компоненте. Следовательно, если какая-либо маркировка с ненулевым весом равна ω , сеть — несохраняющая.

Эти рассуждения относятся к сохранению по отношению к определенному взвешиванию. Сеть Петри является сохраняющей, если она сохраняющая по отношению к некоторому вектору w , $w_i > 0$. Дерево достижимости можно использовать для определения того, является сеть сохраняющей или нет, путем нахождения вектора весов w (если он существует). Заметим прежде всего, что для того, чтобы сеть Петри была сохраняющей по отношению к положительному вектору весов, она должна быть ограниченной. Как было указано выше, неограниченная позиция должна иметь нулевой вес, что недопустимо в сети с положительным вектором весов. (Если мы хотим допустить нулевые компоненты, нужно просто установить веса всех неограниченных позиций равными нулю и рассматривать после этого только оставшиеся компоненты.) Теперь, если сеть сохраняющая, существуют взвешенная сумма, обозначим ее s , и вектор весов

Кононюк А.Е. Графы

$w = (w_1, w_2, \dots, w_n)$. Для каждой маркировки $\mu[x]$ дерева достижимости имеем

$$w_1 \cdot \mu[x]_1 + w_2 \cdot \mu[x]_2 + \dots + w_n \cdot \mu[x]_n = s.$$

Это равенство определяет для k вершин дерева достижимости совокупность из k линейных уравнений с $n + 1$ неизвестными. Добавим к ним ограничения: $w_i > 0, i = 1, \dots, n$, в результате чего определим ограничения для вектора весов.

Решение этой системы линейных уравнений — хорошо известная задача, имеющая множество алгоритмов решения. Можно рассматривать ее как задачу линейного программирования или просто как систему линейных уравнений. В любом случае, если решение существует, оно будет вычислено. (Решения, получаемые этими методами, будут, как правило, рациональными, не целыми, но веса можно умножить на общее кратное для получения целого решения.)

Если ограничения, накладываемые на веса, являются чрезмерно жесткими и, следовательно, вектора взвешиваний не существует, это будет определено. В любом случае можно определить, является ли сеть Петри сохраняющей, и если это так, получить вектор весов.

2.4.2.4. Покрываемость

Последняя задача, которую можно решить с помощью дерева достижимости, — задача покрываемости. В задаче покрываемости мы хотим для данной маркировки μ' определить, достижима ли маркировка $\mu'' \geq \mu'$. Данная задача решается проверкой дерева достижимости. Строим для начальной маркировки μ дерево достижимости. Затем ищем любую вершину x с $\mu[x] \geq \mu'$. Если такой вершины не существует, маркировка μ' не покрывается никакой достижимой маркировкой; если она найдена, $\mu[x]$ дает достижимую маркировку, покрывающую μ' .

Путь от корня к покрывающей маркировке определяет последовательность переходов, которые приводят из начальной маркировки к покрывающей маркировке, а маркировка, связанная с этой вершиной, определяет покрывающую маркировку. Символ ω вновь должен рассматриваться как обозначение бесконечного множества значений. Если компонента покрывающей маркировки — ω , то в пути от корня к покрывающей маркировке имеется «цикл». Для увеличения соответствующей компоненты с тем, чтобы она была не меньше, чем в данной маркировке, необходимо достаточное число раз повторить этот цикл.

Кононюк А.Е. Графы

Заметим, что, если несколько компонент покрывающей маркировки равны ω , между изменениями маркировки, получающимися в результате прохождения циклов, возможна взаимосвязь. Рассмотрим сеть Петри на рис. 18 и ее дерево достижимости, показанное на рис. 19.

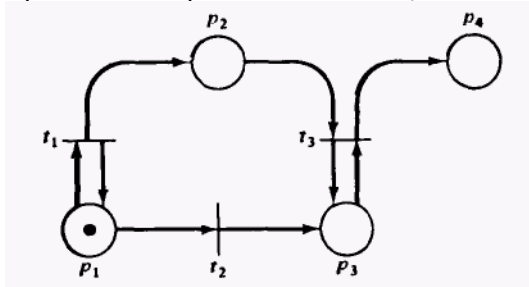


Рис. 18. Сеть Петри.

Согласно проведенному анализу, маркировка $(0, 14, 1, 7)$ покрывается в множестве достижимости. Путь, порождающий покрывающую маркировку, состоит из некоторого числа переходов t_1 , за которыми следует переход t_2 , после которого уже следует некоторое число переходов t_3 . Задача заключается в определении того, сколько раз нужно запустить переходы t_1 и t_3 . Так как мы хотим иметь в позиции p_2 14 фишек, а t_1 помещает в p_2 одну фишку, попытаемся выполнить 14 t_1 .

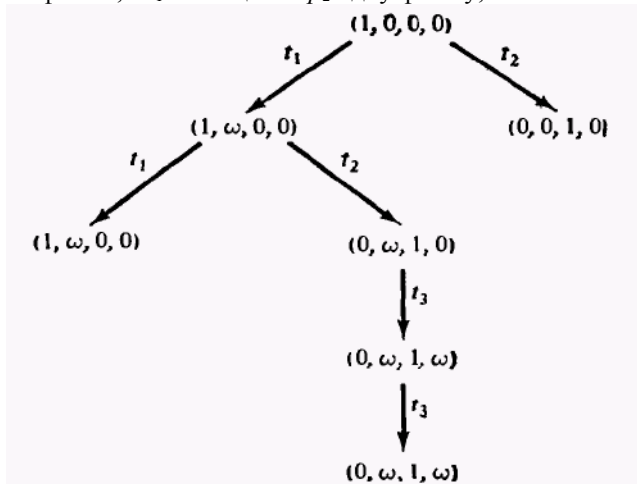


Рис. 19. Дерево достижимости для сети Петри, приведенной на рис. 18.

Однако нам необходимо выполнить $7t_3$, а каждый запуск t_3 удаляет из p_2 фишку, поэтому в действительности необходимо выполнить не

менее $21t_1$, затем t_2 и после этого не менее $7t_3$ (выполнить t_3 такое число раз, чтобы в позиции p_2 осталось не менее 14 фишек). Карп и Миллер предложили алгоритм, определяющий минимальное число запусков переходов, необходимых для покрытия дайной маркировки.

2.4.2.5. Ограниченность дерева достижимости

Как видим, дерево достижимости можно использовать для решения задач безопасности, ограниченности, сохранения и покрытия. К сожалению, в общем случае его нельзя использовать для решения задач достижимости и активности, а также для определения возможной последовательности запусков. Решение этих задач ограничено существованием символа ω . Символ ω означает потерю информации; конкретные количества фишек отбрасываются, учитывается только существование их большого числа.

Рассмотрим, например, сети Петри на рис. 20 и 4.21, дерево достижимости которых изображено на рис. 22.

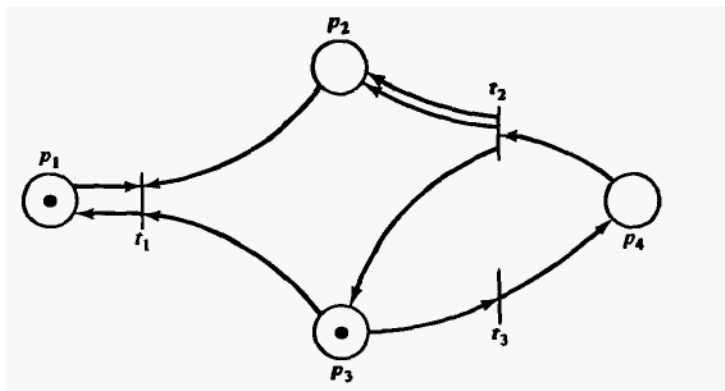


Рис. 20. Сеть Петри, дерево достижимости которой представлено на рис. 22.

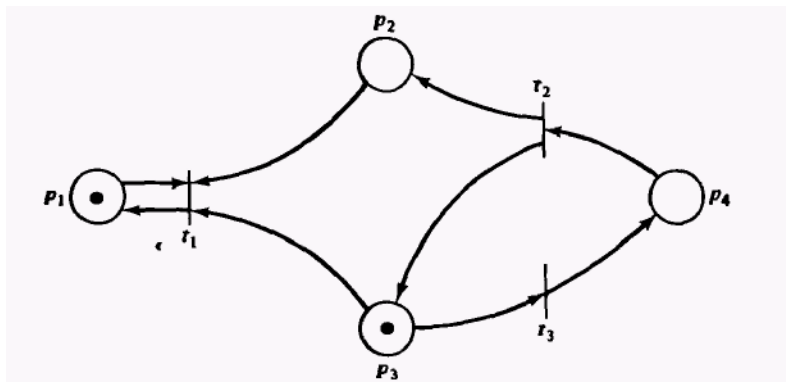


Рис. 21. Вторая сеть Петри, дерево достижимости которой представлено на рис. 22. Сеть Петри, изображенная на рис. 20, имеет в позиции p_2 только четное число фишек, тогда как эта сеть Петри допускает для p_2 произвольную маркировку.

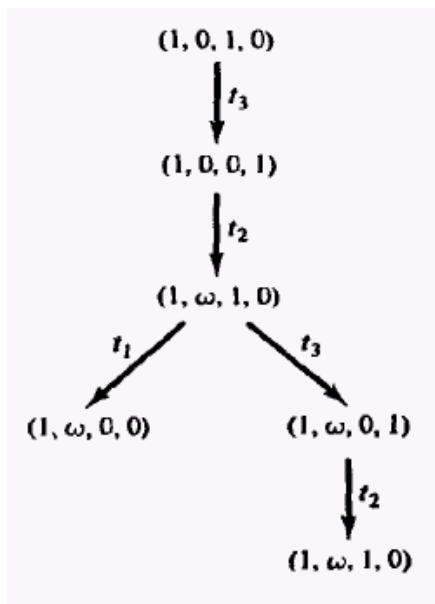


Рис. 22. Дерево достижимости для сетей Петри, приведенных на рис. 20 и 21.

Кононюк А.Е. Графы

Одно дерево достижимости представляет эти две схожие (но различные) сети Петри. Множества же достижимости их не совпадают. В сети Петри на рис. 20 число фишек в позиции p_2 всегда четно (пока не будет запущен переход t_1), тогда как в сети Петри на рис. 21 оно может быть произвольным целым. Символ ω не позволяет обнаруживать информацию такого рода, препятствуя использованию дерева достижимости для решения задачи достижимости. Аналогичная трудность существует и для задачи активности. На рис. 23 и 24 приведены две сети Петри, дерево достижимости которых изображено на рис. 25.

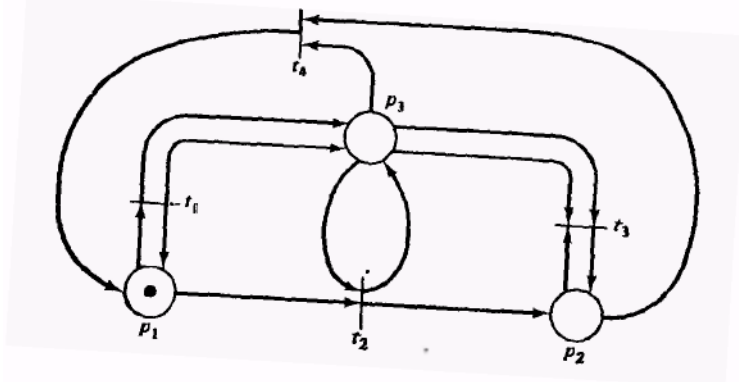


Рис. 23. Сеть Петри с возможным тупиком.

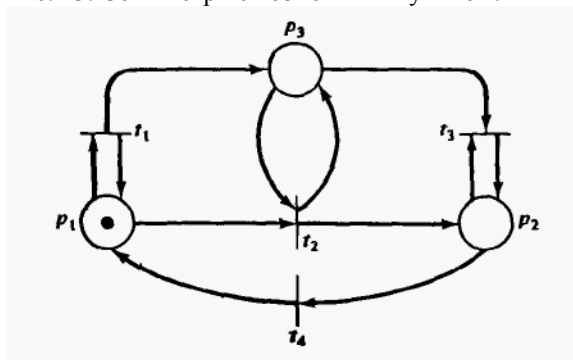


Рис. 24. Сеть Петри, в которой тупик невозможен. Эта сеть активна, хотя ее дерево достижимости (рис. 25) идентично дереву достижимости неактивной сети Петри, приведенной на рис. 23.

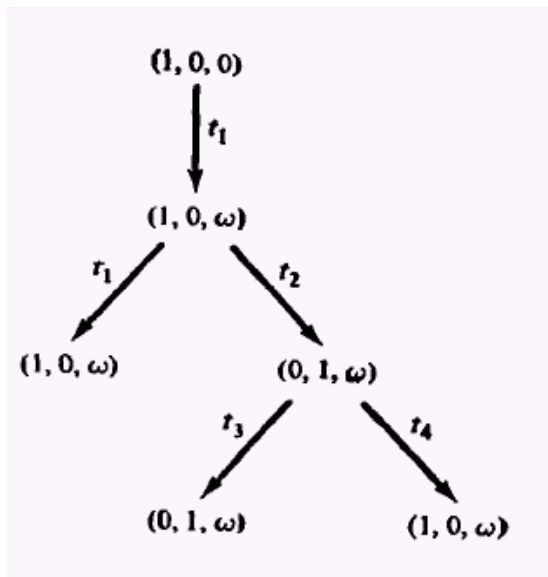


Рис. 25. Дерево достижимости сетей Петри, изображенных на рис. 23 и 24

Однако сеть на рис. 23 может иметь тупик (например, в результате последовательности $t_1 t_2 t_3$), а сеть Петри с рис. 24 — нет. Дерево достижимости же вновь не может передать различие этих двух случаев.

Заметим, что хотя дерево достижимости не обязательно содержит достаточную информацию для решения задач достижимости и активности, тем не менее в некоторых случаях это бывает возможно. Сеть, дерево достижимости которой содержит терминальную вершину (вершину, не имеющую исходящих дуг), не активна (поскольку некоторая достижимая маркировка не имеет последующих маркировок). Аналогично маркировка μ' в задаче достижимости может встретиться в дереве достижимости, и если это так, то она достижима. Кроме того, если маркировка не покрывается некоторой вершиной дерева достижимости, то она недостижима.

Эти условия достаточны для решения некоторых задач достижимости и активности, но они не решают эти задачи в общем. Следовательно, для решения этих двух задач необходимы другие подходы.

Большинство формулировок алгоритма построения дерева достижимости позволяет классифицировать вершину как

Кононюк А.Е. Графы

дублирующую только в том случае, если идентичная вершина встретилась на пути от корня к этой вершине. Следовательно, подобные вершины, принадлежащие разным поддеревьям и представляющие одну маркировку, будут продолжать обрабатываться алгоритмом. Эта модификация (по отношению к описанному в этом разделе алгоритму) увеличивает размер поддерева, но алгоритм закончит работу, даже если дублирующие маркировки находятся в том же поддереве, что и первая маркировка. Доказано, что разрешение дублирующим маркировкам быть в разных поддеревьях не вызовет в дереве потерю какой-либо маркировки.

2.4.2.6. Матричные уравнения

Второй подход к анализу сетей Петри основан на матричном представлении сетей Петри. Альтернативным по отношению к определению сети Петри в виде (P, T, I, O) является определение двух матриц D^- и D^+ , представляющих входную и выходную функции. (Они эквивалентны функциям F и B определения Хэка сетей Петри.) Каждая матрица имеет m строк (по одной на переход) и n столбцов (по одному на позицию). Определим $D^-[j, i] = \#(p_i, I(t_j))$, а $D^+[j, i] = \#(p_i, O(t_j))$. D^- определяет входы в переходы, D^+ — выходы.

Матричная форма определения сети Петри (P, T, D^-, D^+) эквивалентна стандартной форме, используемой нами, но позволяет дать определения в терминах векторов и матриц. Пусть $e[j]$ — m -вектор-строка, содержащий нули везде, за исключением j -й компоненты. Переход t_j представляется m -вектором $e[j]$.

Теперь переход t_j в маркировке μ разрешен, если $\mu \geq e[j] \cdot D^-$, а результат запуска перехода t_j в маркировке μ записывается как

$$\begin{aligned} \delta(\mu, t_j) &= \mu - e[j] \cdot D^- + e[j] \cdot D^+ = \\ &= \mu + e[j] \cdot (-D^- + D^+) = \mu + e[j] \cdot D, \end{aligned}$$

где $D = D^+ - D^-$ — составная матрица изменений.

Тогда для последовательности запусков переходов $\sigma = t_{j_1} t_{j_2} \dots t_{j_k}$

имеем

$$\begin{aligned} \delta(\mu, \sigma) &= \delta(\mu, t_{j_1}, t_{j_2}, \dots, t_{j_k}) = \\ &= \mu + e[j_1] \cdot D + e[j_2] \cdot D + \dots + e[j_k] \cdot D = \\ &= \mu + (e[j_1] + e[j_2] + \dots + e[j_k]) \cdot D = \mu + f(\sigma) \cdot D. \end{aligned}$$

Вектор $f(\sigma) = e[j_1] + e[j_2] + \dots + e[j_k]$ называется вектором за-

Кононюк А.Е. Графы

пусков последовательности $t_{j_1} t_{j_2} \dots t_{j_k}$. i -й элемент вектора $f(\sigma)$, $f(\sigma)_i$ — это число *запусков перехода* t_i в последовательности $t_{j_1} t_{j_2} \dots t_{j_k}$. Вектор запусков, следовательно, является вектором с неотрицательными целыми компонентами. (Вектор $f(\sigma)$ — это *отображение Париха* последовательности.)

Для того чтобы показать полезность такого матричного подхода к сетям Петри, рассмотрим, например, задачу сохранения: является ли данная маркированная сеть Петри сохраняющей? Для того чтобы показать сохранение, необходимо найти (ненулевой) вектор взвешивания, для которого взвешенная сумма по всем достижимым маркировкам постоянна. Пусть $w = n \times 1$ — вектор-столбец. Тогда, если μ — начальная маркировка, а μ' — произвольная достижимая маркировка, необходимо, чтобы $\mu \cdot w = \mu' \cdot w$. Теперь, поскольку μ' достижима, существует последовательность запусков переходов σ , которая переводит сеть из μ в μ' . Поэтому

$$\mu' = \delta(\mu, \sigma) = \mu + f(\sigma) \cdot D.$$

Следовательно,

$$\mu \cdot w = \mu' \cdot w = (\mu + f(\sigma) \cdot D) \cdot w = \mu \cdot w + f(\sigma) \cdot D \cdot w, \text{ поэтому } f(\sigma) \cdot D \cdot w = 0. \text{ Поскольку это должно быть верно для всех } f(\sigma), \text{ имеем}$$

$$D \cdot w = 0.$$

Таким образом, сеть Петри является сохраняющей тогда и только тогда, когда существует такой положительный вектор w , что $D \cdot w = 0$. Это обеспечивает простой алгоритм проверки сохранения, а также позволяет получать вектор взвешивания w .

Развитая матричная теория сетей Петри является инструментом для решения проблемы достижимости. Предположим, что маркировка μ' достижима из маркировки μ . Тогда существует последовательность (возможно, пустая) запусков переходов σ , которая приводит из μ к μ' . Это означает, что $f(\sigma)$ является неотрицательным целым решением следующего матричного уравнения для x :

$$\mu' = \mu + x \cdot D. \tag{1}$$

Следовательно, если μ' достижима из μ , тогда уравнение (1) имеет решение в неотрицательных целых; если уравнение (1) не имеет решения, тогда μ' недостижима из μ . Рассмотрим, например, маркированную сеть Петри на рис. 26.

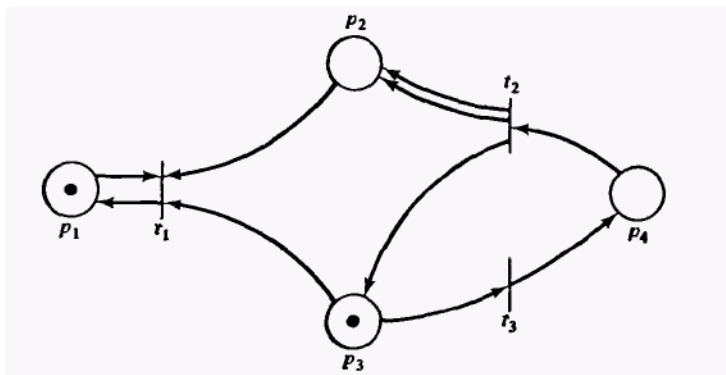


Рис. 26. Сеть Петри, иллюстрирующая метод анализа, основанный на матричных уравнениях.

Матрицы D^- и D^+ имеют вид:

$$D^- = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$D^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

а матрица D :

$$D = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix}.$$

В начальной маркировке $\mu = (1, 0, 1, 0)$ переход t_3 разрешен и приводит к маркировке μ' , где

$$\begin{aligned} \mu' &= (1, 0, 1, 0) + (0, 0, 1) \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix} = \\ &= (1, 0, 1, 0) + (0, 0, -1, +1) = (1, 0, 0, 1). \end{aligned}$$

Последовательность $\sigma = t_3 t_2 t_3 t_2 t_1$ представляется вектором запусков $f(\sigma) = (1, 2, 2)$ и получает маркировку μ' :

Кононюк А.Е. Графы

$$\begin{aligned}
 &= (1, 0, 1, 0) + (1, 2, 2) \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix} = \\
 &= (1, 0, 1, 0) + (0, 3, -1, 0) = (1, 3, 0, 0).
 \end{aligned}$$

Для определения того, является ли маркировка $(1, 8, 0, 1)$ достижимой из маркировки $(1, 0, 1, 0)$, имеем уравнение

$$\begin{aligned}
 (1, 8, 0, 1) &= (1, 0, 1, 0) + x \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix}, \\
 (0, 8, -1, 1) &= x \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix},
 \end{aligned}$$

которое имеет решение $x = (0, 4, 5)$. Это соответствует последовательности $\sigma = t_3 t_2 t_3 t_2 t_3 t_2 t_3 t_2 t_3$.

Далее мы можем показать, что маркировка $(1, 7, 0, 1)$ недостижима из маркировки $(1, 0, 1, 0)$, поскольку матричное уравнение

$$\begin{aligned}
 (1, 7, 0, 1) &= (1, 0, 1, 0) + x \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix} \\
 (0, 7, -1, 1) &= x \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix}
 \end{aligned}$$

не имеет решения.

Матричный подход к анализу сетей Петри очень перспективен, но имеет и некоторые трудности. Заметим прежде всего, что матрица D сама по себе не полностью отражает структуру сети Петри. Переходы, имеющие как входы, так и выходы из одной позиции (петли), представляются соответствующими элементами матриц D^- и D^+ , но затем взаимно уничтожаются в матрице $D = D^+ - D^-$. Это отражено в предыдущем примере позицией p_l и переходом t_l .

Другая проблема — это отсутствие информации о последовательности в векторе запуска. Рассмотрим сеть Петри на рис. 27.

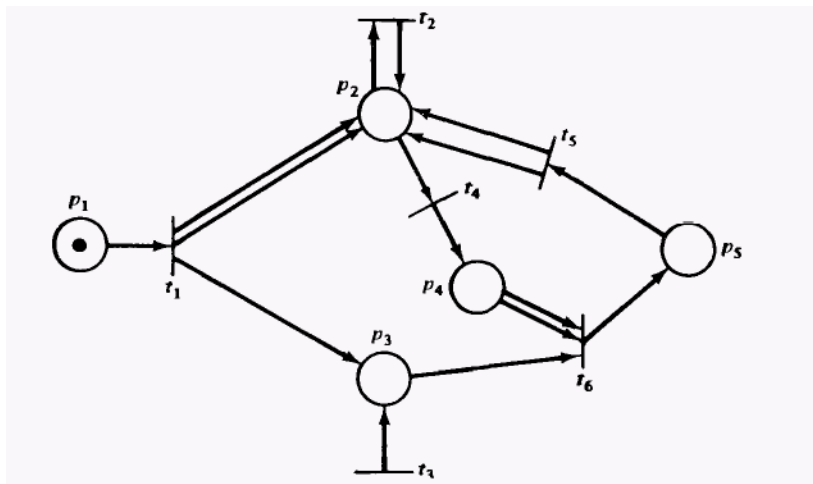


Рис. 27. Другая сеть Петри, служащая для иллюстрации матричного анализа.

Предположим, мы хотим определить, является ли маркировка $(0, 0, 0, 0, 1)$ достижимой из $(1, 0, 0, 0, 0)$. Тогда имеем уравнение

$$(0, 0, 0, 0, 1) = (1, 0, 0, 0, 0) + x \cdot \begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & -1 & -2 & 1 \end{bmatrix}.$$

Это уравнение не имеет однозначного решения, но сводится к множеству решений $\{x | f(x) = (1, x_2, x_6 - 1, 2x_6, x_6 - 1, x_6)\}$. Оно определяет взаимосвязь между запусками переходов. Если положим $x_6 = 1$ и $x_2 = 1$, то $f(x) = (1, 1, 0, 2, 0, 1)$, но этому вектору запуска соответствуют как последовательность $t_1 t_2 t_4 t_6$, так и последовательность $t_1 t_4 t_2 t_6$. Следовательно, хотя и известно число запусков переходов, порядок их запуска неизвестен.

Еще одна трудность заключается в том, что решение уравнения (1) является необходимым для достижимости, но недостаточным. Рассмотрим простую сеть Петри, приведенную на рис. 28.

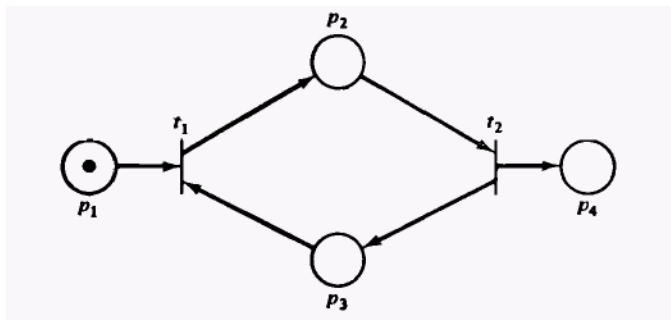


Рис. 28. Сеть Петри, показывающая, что решение матричного уравнения— необходимое, но недостаточное условие для решения задачи достижимости.

Если мы хотим определить, является ли $(0, 0, 0, 1)$ достижимым из $(1, 0, 0, 0)$, необходимо решить уравнение

$$(0, 0, 0, 1) = (1, 0, 0, 0) + f(\sigma) \cdot \begin{bmatrix} -1 & +1 & -1 & 0 \\ 0 & -1 & +1 & +1 \end{bmatrix}.$$

Это уравнение имеет решение $f(\sigma) = (1, 1)$, соответствующее двум последовательностям: $t_1 t_2$ и $t_2 t_1$. Но ни одна из этих двух последовательностей переходов невозможна, поскольку в $(1, 0, 0, 0)$ ни t_1 , ни t_2 не разрешены. Таким образом, решения уравнения (1) недостаточно для доказательства достижимости.

Возможность недействительных решений уравнения (1) (решений, которые не соответствуют возможным последовательностям переходов) стала причиной только ограниченного исследования матричного представления сетей Петри.

В публикациях Хольта и др. и Хольта и Коммонера определены некоторые из первых задач анализа сетей Петри — активность и безопасность, которые продолжают считаться основными задачами анализа. Активность изучалась Коммонером, Лаутенбахом и Льеном. Келлер вместе с другими задачами рассматривал также и активность. Льен определил задачу сохранения.

Карп и Миллер впервые описали построение дерева достижимости и доказали его конечность. Задачи покрываемости и достижимости были определены ими как если бы это были задачи эквивалентности и включения. Матричный подход рассматривался Питерсоном, но, как оказалось, возможности его ограничены. Мурата, имеющий большую

Кононюк А.Е. Графы

квалификацию в линейной алгебре, достиг чуть большего в этом подходе.

2.5. СЛОЖНОСТЬ И РАЗРЕШИМОСТЬ

В разделе 2.4 было представлено множество задач теории сетей Петри. Эти задачи касаются различных свойств структуры и поведения сетей Петри.

Были представлены и два метода решения: подход дерева достижимости и подход матричных уравнений. Эти методы позволяют определить свойства безопасности, ограниченности, сохранения и покрываемости для сетей Петри. Кроме того, установлены необходимые условия для достижимости. Однако эти методы анализа недостаточны для решения некоторых других задач, в частности активности, достижимости и эквивалентности. В этом разделе мы изучим эти задачи с целью или найти их решения, или по крайней мере узнать больше о свойствах сетей Петри.

2.5.1. Сводимость задач анализа

Будем использовать фундаментальное понятие *сводимости*. Решение задачи подразумевает сведение ее к другой задаче, способ решения которой уже известен. В предыдущем разделе, например, задача определения того, является ли сеть Петри сохраняющей, сводилась к решению системы линейных уравнений. Задача решения системы линейных уравнений в свою очередь сводится к определенной последовательности арифметических операций (сложение, вычитание, умножение, деление и сравнение). Таким образом, поскольку арифметические операции выполнить можно, сохранение может быть определено.

Другой пример связан с задачей равенства и задачей подмножества для множеств достижимости.

Определение 1. Задача равенства. Выполняется ли для двух данных маркированных сетей Петри $C_1 = (P_1, T_1, I_1, O_1)$ с маркировкой μ_1 и $C_2 = (P_2, T_2, I_2, O_2)$ с маркировкой μ_2 равенство $R(C_1, \mu_1) = R(C_2, \mu_2)$?

Определение 2. Задача подмножества. Выполняется ли для двух данных маркированных сетей Петри $C_1 = (P_1, T_1, I_1, O_1)$ с маркировкой μ_1 и $C_2 = (P_2, T_2, I_2, O_2)$ с маркировкой μ_2 соотношение $R(C_1, \mu_1) \subseteq R(C_2, \mu_2)$?

Эти задачи могут оказаться очень важными для определения, является ли сеть Петри оптимизируемой или сравнимы ли сети двух систем.

Кононюк А.Е. Графы

Заметим, что если можно найти решение задачи подмножества, то задача равенства также решается. Если необходимо определить, выполняется ли $R(C_1, \mu_1) = R(C_2, \mu_2)$, сначала применим алгоритм решения задачи подмножества для определения, выполняется ли $R(C_1, \mu_1) \subseteq R(C_2, \mu_2)$, а затем применим тот же алгоритм для определения, выполняется ли $R(C_2, \mu_2) \subseteq R(C_1, \mu_1)$, $R(C_1, \mu_1) = R(C_2, \mu_2)$ тогда и только тогда, когда $R(C_1, \mu_1) \subseteq R(C_2, \mu_2)$ и $R(C_2, \mu_2) \subseteq R(C_1, \mu_1)$. Таким образом, задачу равенства можно *свести* к задаче подмножества.

При рассмотрении задач анализа и сводимости важно принимать во внимание следующие два соображения. Во-первых, пытаясь найти решение, необходимо учитывать возможность того, что задача не имеет метода решения, т. е. *неразрешима*. Во-вторых, если метод решения существует, мы должны оценить его стоимость: как много времени и памяти требуется для решения? Для того чтобы применение сетей Петри расширялось, задачи анализа должны решаться, а алгоритмы, осуществляющие решение, должны быть не слишком дороги в смысле времени и памяти ЭВМ.

Сводимость играет большую роль для обеих из указанных проблем. Сводимость задач обычно используется для того, чтобы показать, разрешима ли задача или неразрешима. Рассматриваемый подход к *теории разрешимости* основан главным образом на работах Тьюринга и его модели вычислений — машине Тьюринга. Значение машины Тьюринга состоит в том, что она служит разумным представлением вычислительной машины, и можно показать, что не существует алгоритма, который бы решал определенные проблемы машины Тьюринга, в частности проблему остановки. На основе этого был найден ряд неразрешимых задач. И важность этой теории — в том, что невозможно создать программу для ЭВМ, которая бы решала эти задачи. Следовательно, для целей практического анализа необходимо избежать этих неразрешимых задач, иначе вопросы анализа не получат ответа.

(Здесь важно подчеркнуть, что неразрешимые задачи порождают вопросы, на которые не просто нет ответа, но его и *не может быть*. Вопросы, на которые нет ответа, еще могут его получить; это значит, что ответ еще никто не нашел, но он существует. Известным примером является большая теорема Ферма: имеет ли решение уравнение $x^n + y^n = z^n$ для $n > 2$ в классе целых чисел x, y, z ? Этот вопрос не получил ответа, но он его имеет. Ответом служит либо "да", либо "нет". Одним способом получения ответа на вопрос является поиск чисел x, y, z и n , удовлетворяющих условиям теоремы. Другим будет

Кононюк А.Е. Графы

доказательство (логический вывод) того, что таких x , y , z и n не существует. Никто еще не сделал этого.

Однако предположим, что задача неразрешима. Тогда невозможно было бы определить, существуют ли x, y, z и n , решающие уравнение. Это означало бы, что мы не могли логически вывести их несуществование из аксиом математики и не могли получить x, y, z и n , которые были бы решением уравнения. Но если мы не можем получить x, y, z и n , тогда они не должны существовать. Если же они существуют, мы можем найти их с помощью ЭВМ. Но если x, y, z и n не существуют, ответом на вопрос будет «нет», и, следовательно, мы разрешили его. Это противоречит нашему предположению о том, что вопрос неразрешим, следовательно, он разрешим.)

Теперь допустим, что задача A сводима к задаче B : конкретную задачу A можно перевести в конкретную задачу B . Если задача B разрешима, то разрешима и задача A , а алгоритм решения задачи B можно применить к решению задачи A . Конкретную задачу A можно решить, преобразуя ее в конкретную задачу B и применяя к ней алгоритм решения задачи B . Таким образом, если задача A сводится к задаче B и задача B разрешима, то разрешима и задача A .

Обратное также верно: если задача A сводится к задаче B и задача A неразрешима, то неразрешима и задача B ; если бы задача B была разрешима, то описанная выше процедура была бы методом решения задачи A , что противоречит ее неразрешимости. Эти два факта положены в основу большинства методов определения разрешимости. Для того чтобы показать, что задача разрешима, сводят ее к задаче, известной как разрешимая; а для того чтобы показать, что задача неразрешима, сводят к ней задачу, которая известна как неразрешимая. Мы также будем широко использовать этот подход для сокращения объема работы, которую нам необходимо выполнить. Например, вследствие того, что задача равенства для множеств достижимости сводится к задаче подмножества, мы хотим найти либо 1) процедуру решения задачи подмножества, либо 2) доказательство того, что задача равенства неразрешима. Если мы сможем выполнить 1), то будем иметь метод решения обеих задач, если выполним 2), будем знать, что обе задачи неразрешимы.

В некоторых случаях мы сможем достичь даже большего. Две задачи называются *эквивалентными*, если они взаимно сводимы. Другими словами, задача A эквивалентна задаче B , если задача A сводится к задаче B , а задача B сводится к задаче A . В этом случае или обе задачи разрешимы, или обе неразрешимы, и мы можем рассматривать любую из них. (Заметим, что в общем случае это неверно. Например, если бы мы показали, что задача подмножества для множеств достижимости

Кононюк А.Е. Графы

неразрешима, это не сказало бы нам ничего о разрешимости или неразрешимости задачи равенства.)

Второе соображение по исследованию задач анализа заключается в том, что если метод решения существует, то он должен быть разумно эффективным. Это означает, что количества времени и памяти, требуемые алгоритмом для решения конкретной задачи, не должны быть чрезмерными. Исследование стоимости выполнения алгоритма — это часть *теории сложности*. Теория сложности имеет дело с количествами времени и памяти, необходимыми для решения задачи. Очевидно, что количества времени и памяти не постоянны, а изменяются в зависимости от размерности решаемой задачи. Для сетей Петри временные и емкостные затраты будут, по-видимому, функцией от числа позиций и переходов. Другими факторами, влияющими на время и память, будут число фишек в начальной маркировке и число входов и выходов для каждого перехода и позиции (число дуг в графе). Требуемое время и память будут изменяться от одной задачи к другой. Поэтому оценки сложности алгоритма могут быть даны для наилучшего случая (нижняя граница) или наихудшего случая (верхняя граница). Поскольку не известно, относится данный пример задачи к наилучшему или наихудшему случаю, обычно принимается наихудший случай, и *сложность* алгоритма — это временные и емкостные затраты в худшем случае как функция от размерности задачи.

Анализ сложности касается главным образом сложности задачи, а не конкретных деталей реализации какого-либо частного алгоритма. Поэтому в теории сложности игнорируются константы. Сложность задачи размерности n определяют порядком n^2 или 2^n , или $n \log n$, допуская меньшие члены и постоянные множители. Особенно важны два общих класса алгоритмов: имеющих полиномиальную сложность (n , n^2 , $n \log n$, n^8 и т. д.) и имеющих неполиномиальную сложность (в частности, экспоненциальную, 2^n , и факториальную, $n!$).

Анализ сложности обычно применяют к конкретным алгоритмам, но его можно применить также и к общим задачам. В этом случае определяется нижняя оценка сложности из всех алгоритмов, решающих задачу. Это обеспечивает значение сложности, не зависящее от алгоритма, и, кроме того, может оказаться полезным в доказательстве того, что данный алгоритм оптимален (с точностью до константы), и в определении случаев, когда дальнейшая работа может привести к лучшему алгоритму решения задачи. Например, хорошо известно, что сортировка n чисел имеет сложность $n \log n$. Следовательно, алгоритмы сложности $n \log n$ нельзя значительно улучшить (в асимптотически худшем случае).

При определении сложности может быть полезно сведение одной задачи к другой. Если задачу A можно свести к задаче B и B имеет сложность $f_B(n)$, то сложность A — это, самое большее, сложность B плюс стоимость преобразования из A в B (помня, что в результате перевода размерность задачи может измениться.) Сложность преобразования является, как правило, константой или линейной функцией и поэтому часто игнорируется. Следовательно, сведение задачи A к задаче B дает или верхнюю границу сложности A (если известна сложность B), или нижнюю границу сложности B (если известна сложность A). Используя снова в качестве примера задачи равенства и подмножества, получаем, что объем вычислений, необходимый для решения задачи равенства, не больше чем удвоенный объем вычислений для задачи подмножества. Поскольку здесь участвует постоянный сомножитель, сложность задачи подмножества должна совпадать со сложностью задачи равенства.

Эти два свойства задач анализа сетей Петри — разрешимость и сложность — имеют первостепенную важность для использования сетей Петри. В этом разделе представляются некоторые известные результаты. Один из используемых методов — сведение одной задачи анализа Петри к другой.

2.5.2. Задачи достижимости

Задача достижимости — одна из самых важных задач анализа сетей Петри. Она до сих пор еще не решена для различных вариаций ее определения. Были поставлены следующие четыре задачи достижимости для сети Петри $C = (P, T, I, O)$ с начальной маркировкой μ .

Определение 3. *Задача достижимости.* Выполняется ли для данной μ' : $\mu' \in R(C, \mu)$?

Определение 4. *Задача достижимости подмаркировки.* Для подмножества $P' \subseteq P$ и маркировки μ' существует ли $\mu'' \in R(C, \mu)$, такая, что $\mu''(p_i) = \mu'(p_i)$ для всех $p_i \in P'$?

Определение 5. *Задача достижимости нуля.* Выполняется ли $\mu' \in R(C, \mu)$, где $\mu'(p_i) = 0$ для всех $p_i \in P$? ($0 \in R(C, \mu)$?)

Определение 6. *Задача достижимости нуля в одной позиции.* Для данной позиции $p_i \in P$ существует ли $\mu' \in R(C, \mu)$ с $\mu'(p_i) = 0$?

Задача достижимости подмаркировки ограничивает задачу достижимости до рассмотрения только подмножества позиций, не принимая во внимание маркировки других позиций. Задача достижимости нуля выясняет, является ли достижимой частная

Кононюк А.Е. Графы

маркировка с нулем фишек во всех позициях. Задача достижимости нуля в одной позиции выясняет, возможно ли удалить все фишки из данной позиции.

Хотя эти четыре задачи различны, все они эквивалентны. Определенные взаимосвязи очевидны сразу. Задача достижимости нуля сводится к задаче достижимости; просто в задаче достижимости устанавливается $\mu' = 0$. Аналогично задача достижимости сводится к задаче достижимости подмаркировки путем установки $P' = P$. Задача достижимости нуля в одной позиции сводится к задаче достижимости подмаркировки установкой $P' = \{p_i\}$ и $\mu' = 0$. Сложнее показать, что задача достижимости подмаркировки сводится к задаче достижимости нуля и что задача достижимости нуля сводится к задаче достижимости нуля в одной позиции. Все множество взаимосвязей представлено на рис. 1.

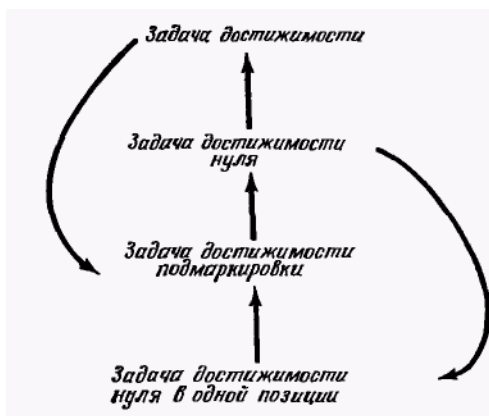


Рис. 1. Сводимость задач достижимости. Дуга от одной задачи к другой означает, что первая сводится ко второй.

Сначала покажем, что задача достижимости подмаркировки сводится к задаче достижимости нуля. Пусть даны сеть Петри $C_1 = (P_1, T_1, I_1, O_1)$ с начальной маркировкой μ_1 , подмножество $P' \subseteq P_1$ и маркировка μ' . Мы хотим узнать, существует ли $\mu'' \in R(C, \mu)$ с $\mu'(p_i) = \mu''(p_i)$ для всех $p_i \in P'$. Наш подход заключается в построении новой сети Петри $C_2 = (P_2, T_2, I_2, O_2)$ с маркировкой μ_2 , такой, что маркировка $\mu'' \in R(C_1, \mu_1)$ с $\mu'(p_i) = \mu''(p_i)$ для всех $p_i \in P'$ существует тогда и только тогда, когда $0 \in R(C_2, \mu_2)$.

Кононюк А.Е. Графы

Построение C_2 из C_1 осуществляется просто. Начнем с C_2 , совпадающей с C_1 . Для того чтобы позволить всякой позиции p_i вне P' стать пустой, введем переход t'_i с входом $\{p_i\}$ и пустым выходом. Этот переход можно запускать всякий раз, когда в p_i имеются фишки, чтобы изъять их, что позволяет достичь нулевой маркировки и тем самым игнорировать такие позиции.

Относительно p_i в P' необходимо иметь уверенность, что в p_i точно $\mu'(p_i)$ фишек. Для этого построим новые позиции p'_i для каждой $p_i \in P'$ с начальной маркировкой $\mu'(p_i)$ фишек и переход t'_i с входом $\{p_i, p'_i\}$ и пустым выходом. Если в p_i точно $\mu'(p_i)$ фишек, то этот переход можно запустить точно $\mu'(p_i)$ раз, сокращая маркировки p_i и p'_i до нуля. Если число фишек в p_i отлично от $\mu'(p_i)$, то переход t_i можно запустить только минимальное из двух маркировок число раз, и поэтому либо в p_i , либо в p'_i останутся фишки, препятствующие достижению нулевой маркировки.

Два типа вводимых переходов иллюстрируются на рис. 2.

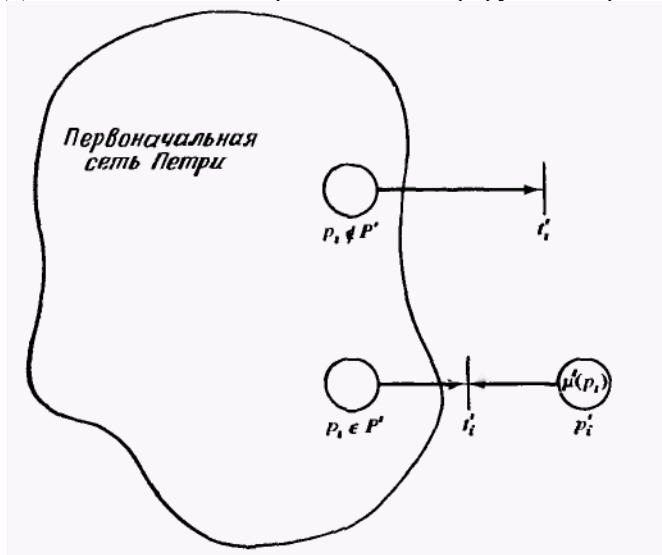


Рис. 2. Сеть Петри, служащая для доказательства сводимости задачи достижимости подмаркировки к задаче достижимости нуля. Подмножество позиций P' будет иметь маркировку μ' в первоначальной сети тогда и только тогда, когда в модифицированной показанным здесь способом сети будет достижима нулевая маркировка.

Формально определим C_2 следующим образом:

$$P_2 = P_1 \cup \{p'_i \mid p_i \in P'\},$$

$$T_2 = T_1 \cup \{t'_i \mid p_i \in P_1\},$$

$$I_2(t_j) = I_1(t_j) \text{ для } t_j \in T_1,$$

$$I_2(t'_i) = \{p_i\} \text{ для } p_i \notin P',$$

$$= \{p_i, p'_i\} \text{ для } p_i \in P',$$

$$O_2(t_j) = O_1(t_j) \text{ для } t_j \in T_1,$$

$$O_2(t'_i) = \{ \} \text{ для } p_i \in P_1,$$

с начальной маркировкой:

$$\mu_2(p_i) = \mu_1(p_i), \quad p_i \in P_1,$$

$$\mu_2(p'_i) = \mu'(p_i), \quad p_i \in P'.$$

Теорема 1. Задача достижимости подмаркировки сводится к задаче достижимости нуля.

Доказательство. Покажем, что для сети Петри C_2 , построенной из C_1 вышеописанным способом, $0 \in R(C_2, \mu_2)$ тогда и только тогда, когда $\mu'' \in R(C_1, \mu_1)$ с $\mu''(p_i) = \mu'(p_i)$ для всех $p_i \in P'$.

Для того чтобы показать, что $0 \in R(C_2, \mu_2)$ тогда и только тогда, когда существует $\mu'' \in R(C_1, \mu_1)$ с $\mu''(p_i) = \mu'(p_i)$ для $p_i \in P'$, допустим сначала, что μ'' присутствует в $R(C_1, \mu_1)$. Тогда в C_2 также можно достичь маркировки μ'' в позициях $p_i \in P_1$ запуском только переходов из T_1 . Сейчас для каждой позиции $p_i \in P'$ можно запустить t'_i точно $\mu'(p_i)$ раз, сокращая маркировку и p_i и p'_i до нуля. Затем можно запустить t'_i для каждой $p_i \notin P'$ столько раз, сколько необходимо для приведения маркировки этих позиций к нулю, поэтому $0 \in R(C_2, \mu_2)$. Теперь предположим, что $0 \in R(C_2, \mu_2)$; тогда существует последовательность запусков переходов σ , переводящая μ_2 в 0 . Эта последовательность будет содержать точно $\mu'(p_i)$ запусков t'_i для $p_i \in P'$ (для удаления фишек из p'_i) и некоторое число запусков t'_i для $p_i \notin P'$. Заметим, что запуски этих переходов только удаляют фишки из C_1 , и поскольку $\delta(\mu', t_j)$ определено там, где определено $\delta(\mu, t_j)$ для $\mu' \geq \mu$, (лишние фишки не могут повредить запуску переходов), последовательность σ после удаления из нее всех запусков t'_i останется действительной и будет приводить к маркировке μ'' с точно $\mu'(p_i)$ фишками в p_i для $p_i \in P'$. Таким образом, если $0 \in R(C_2, \mu_2)$, то $\mu'' \in R(C_1, \mu_1)$ с $\mu''(p_i) = \mu'(p_i)$ для $p_i \in P'$.

Кононюк А.Е. Графы

Наша следующая задача — показать, что задача достижимости нуля сводится к задаче достижимости нуля в одной позиции. Доказательство этого утверждения также основано на вспомогательном построении. Пусть дана сеть Петри $C_1 = (P_1, T_1, I_1, O_1)$ с начальной маркировкой μ_1 , мы хотим определить, выполняется ли $0 \in R(C_1, \mu_1)$. Построим из C_1 новую сеть Петри C_2 с дополнительной позицией $s (P_2 = P_1 \cup \{s\})$, такой, что маркировка $\mu' \in R(C_2, \mu_2)$ с $\mu'(s) = 0$ существует тогда и только тогда, когда $0 \in R(C_1, \mu_1)$.

Построение C_2 определяет s так, что всякий раз число фишек в s равно сумме числа фишек в позициях C_1 . Следовательно, если $\mu'(s) = 0$, то в позициях C_1 нет фишек, и наоборот. Определим начальную маркировку μ_2 следующим образом:

$$\mu_2(p_i) = \mu_1(p_i) \text{ для } p_i \in P_1,$$

$$\mu_2(s) = \sum_{p_i \in P_1} \mu_1(p_i).$$

Далее для каждого перехода $t_j \in T_1$ в C_2 будет тот же переход, но с добавленными дугами к позиции s . Определим

$$d_j = \sum_{p_i \in P_1} (\#(p_i, O(t_j)) - \#(p_i, I(t_j))).$$

Тогда d_j — это разность числа фишек после запуска перехода t_j и числа фишек до запуска t_j . Теперь, если $d_j > 0$, то d_j фишек должно быть добавлено в позицию s , поэтому введем d_j дуг из t_j в s , если $d_j < 0$, то удалим $-d_j$ фишек из s введением $-d_j$ дуг из s в t_j .

Если $d_j > 0$, то $\#(s, I(t_j)) = 0$, $\#(s, O(t_j)) = d_j$.

Если $d_j < 0$, то $\#(s, I(t_j)) = -d_j$, $\#(s, O(t_j)) = 0$.

Если $d_j = 0$, то $\#(s, I(t_j)) = 0$, $\#(s, O(t_j)) = 0$.

При таком построении любая последовательность запусков переходов, приводящая C_1 к маркировке 0, приведет C_2 к маркировке μ' с $\mu'(s) = 0$ (а также $\mu'(p_i) = 0$), и наоборот.

Теорема 2. Задача достижимости нуля сводится к задаче достижимости нуля в одной позиции.

Доказательство. Формальное доказательство, основанное на описанной конструкции, оставляем читателю.

На основе этих двух теорем и очевидных выводов можно заключить следующее:

Теорема 3. Следующие задачи достижимости эквивалентны:

1. Задача достижимости.
2. Задача достижимости нуля.
3. Задача достижимости подмаркировки.
4. Задача достижимости нуля в одной позиции.

Эти теоремы и их доказательства принадлежат главным образом Хэку.

2.5.3. Сети Петри с ограничениями

В первых работах по сетям Петри, а также в некоторых работах сети Петри определяются в несколько более ограниченной форме, чем в определении, данном ранее. В частности, иногда принимаются следующие два ограничения:

Ограничение 1. Кратность любой позиции равна 0 или 1. Иначе говоря, $\#(p_i, I(t_j)) \leq 1$ и $\#(p_i, O(t_j)) \leq 1$ для всех $p_i \in P$ и $t_j \in T$. Это ограничивает входные и выходные комплекты до множеств.

Ограничение 2. Никакая позиция не может быть одновременно входом и выходом одного и того же перехода. $I(t_j) \cap O(t_j) = \emptyset$. Это часто формулируется как $\#(p_i, I(t_j)) \cdot \#(p_i, O(t_j)) = 0$ для всех p_i и t_j .

Сети Петри, удовлетворяющие ограничению 1, называются *ординарными*. Сети Петри, удовлетворяющие ограничению 2, называются *сетями Петри без петель*, или *нерефлексивными сетями*. Сети Петри, удовлетворяющие обоим ограничениям, называются *простыми сетями Петри*. Эти классы сетей Петри соотносятся так, как показано на рис. 3.

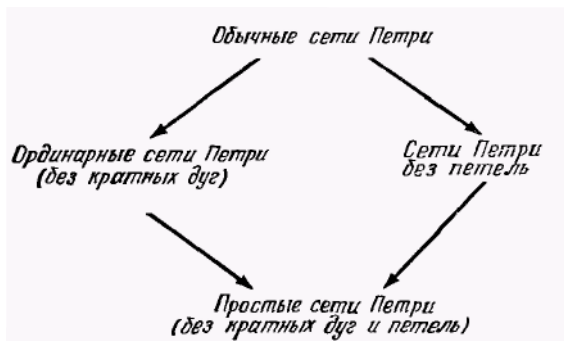


Рис. 3. Взаимосвязи между классами сетей Петри. Дуга означает включение; дуги сводимости будут ориентированы в противоположном направлении.

Подклассы модели обычных сетей Петри рассматривались по нескольким причинам. Основная причина заключается в том, что введение понятий сетей Петри в начале развития теории было не-

формальным. Необходимость в кратных дугах и петлях при моделировании тогда отсутствовала. Кроме того, казалось, что теория без этих усложнений будет проще. По мере развития теории стало ясно, что работать с более общими определениями не труднее. Использование моделей с этими ограничениями в работах объясняется стремлением исследователя к более простому и лаконичному изложению.

Однако эти ограничения не добавляют ничего к способности анализировать сети Петри. Рассмотрим для этих классов сетей задачу достижимости. Для того чтобы показать эквивалентность всех четырех классов сетей Петри, докажем следующее:

Теорема 4. Задача достижимости эквивалентна для следующих классов сетей Петри:

1. Обычные сети Петри.
2. Ординарные сети Петри.
3. Сети Петри без петель.
4. Простые сети Петри.

Доказательство. Из определений очевидны следующие сведения:

1. Задача достижимости для ординарных сетей Петри сводится к задаче достижимости для обычных сетей Петри.
2. Задача достижимости для сетей Петри без петель сводится к задаче достижимости для обычных сетей Петри.
3. Задача достижимости для простых сетей Петри сводится и к задаче достижимости для ординарных сетей Петри, и к задаче достижимости для сетей Петри без петель.

Покажем, что обычные сети Петри можно преобразовать в простые сети Петри таким образом, чтобы свести задачу достижимости для обычных сетей Петри к задаче достижимости для простых сетей Петри. Это покажет, что все четыре задачи достижимости эквивалентны.

Для преобразования обычной сети Петри в простую сеть Петри используем следующий подход. Каждая позиция в обычной сети Петри заменяется *кольцом* позиций в простой сети Петри. На рис. 4 показан общий вид кольца позиций.

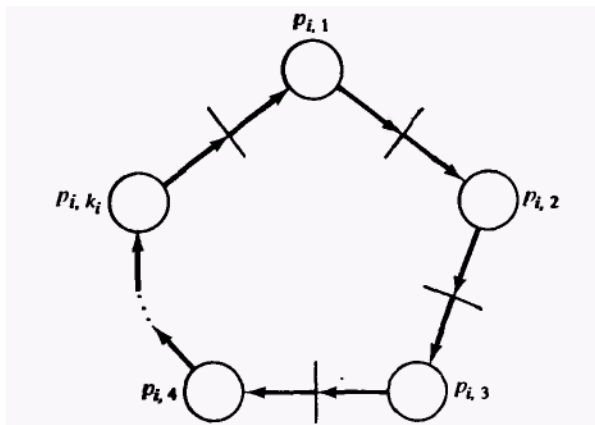


Рис. 4. Кольцо позиций, используемое в простой сети Петри для представления позиции обычной сети Петри. Число k_i позиций, представляющих позицию p_i , определяется максимумом сумм кратностей позиции.

Заметим, что набор фишек, помещенных в кольцо, может свободно двигаться по кольцу к любой позиции в любой момент времени; все они могут сгруппироваться в позиции $p_{i,1}$ или равномерно распределиться, покрывая все k_i позиций кольца. Поэтому переход, для запуска которого необходимы три фишки из позиции p_i , может выбрать вместо всех трех из p_i по одной из $p_{i,1}$, $p_{i,2}$ и $p_{i,3}$. Аналогично переход, использующий p_i в качестве входа и в качестве выхода (петля), может иметь вход в $p_{i,1}$, а выход в $p_{i,2}$, тем самым устранив петлю.

Формально для обычной сети Петри $S_1 = (P_1, T_1, I_1, O_1)$ с маркировкой μ_1 определим простую сеть Петри $S_2 = (P_2, T_2, I_2, O_2)$ с маркировкой μ_2 следующим образом. Сначала для каждой позиции $p_i \in P_1$ определим целое k_i :

$$k_i = \max_{t_j \in T} (\#(p_i, I(t_j)) + \#(p_i, O(t_j))).$$

Сеть Петри с ограничениями S_2 определяется

$$P_2 = \{p_{i,h} \mid p_i \in P_1, 1 \leq h \leq k_i\},$$

$$T_2 = T_1 \cup \{t_{i,h} \mid p_{i,h} \in P_2\}.$$

Входная и выходная функции для «нормальных» переходов определяются так, что

Кононюк А.Е. Графы

$$\#(p_{i,h}, I_2(t_j)) = \begin{cases} 1, & \text{если } 1 \leq h \leq \#(p_i, I_1(t_j)), \\ 0, & \text{в противном случае;} \end{cases}$$

$$\#(p_{i,h}, O_2(t_j)) = \begin{cases} 1, & \text{если } \#(p_i, I_1(t_j)) < h \leq \#(p_i, I_1(t_j)) + \\ & + \#(p_i, O_1(t_j)), \\ 0, & \text{в противном случае;} \end{cases}$$

а для «кольцевых» переходов

$$I_2(t_{i,h}) = \{p_{i,h}\},$$

$$O_2(t_{i,h}) = \{p_{i,n} | n = 1 + (h \pmod{k_i})\}.$$

Маркировка μ_2 определяется следующим образом:

$$\mu_2(p_{i,1}) = \mu_1(p_i) \text{ для } p_i \in P_1,$$

$$\mu_2(p_{i,h}) = 0 \text{ для } h > 1.$$

Для любой маркировки μ , достижимой в C_1 по построению существует такая маркировка μ' в C_2 , что

$$\sum_h \mu'(p_{i,h}) = \mu(p_i) \text{ для всех } p_i \in P_1,$$

в частности, в любой момент времени в C_2 можно переместить все фишки из $p_{i,h}$ в $p_{i,1}$. Следовательно, можно определить маркировку μ' следующим образом:

$$\mu'(p_{i,1}) = \mu(p_i) \text{ для } p_i \in P_1,$$

$$\mu'(p_{i,h}) = 0 \text{ для } h > 1,$$

и μ' становится достижимой в простой сети Петри C_2 тогда и только тогда, когда μ достижима в C_1 .

Таким образом, с точки зрения анализа обычные сети Петри и три ограниченных класса обычных сетей Петри — ординарные сети Петри, сети Петри без петель и простые сети Петри — эквивалентны, всякую сеть можно преобразовать в подобную сеть другого класса, сводя задачу достижимости для одной сети к задаче достижимости для другой. Используемые в этом разделе конструкции принадлежат Хэку.

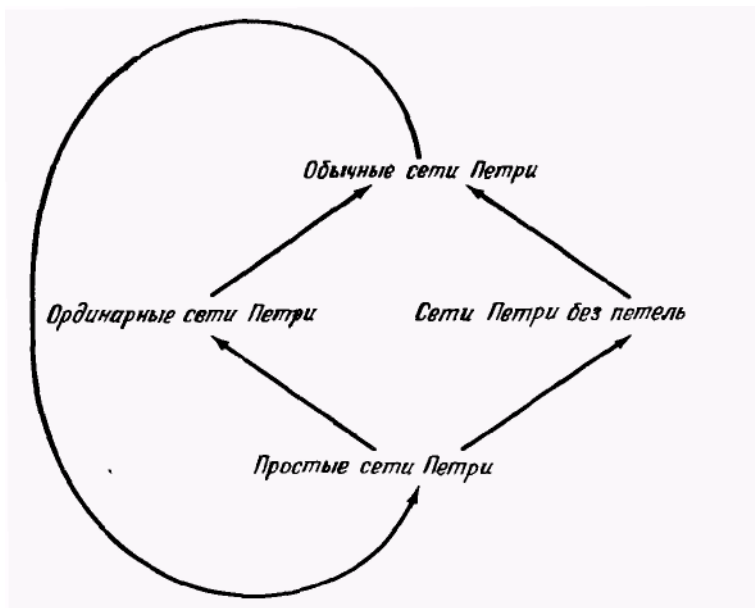


Рис. 5. Сводимость задачи достижимости среди классов сетей Петри с различными типами ограничений.

2.5.4. АКТИВНОСТЬ И ДОСТИЖИМОСТЬ

Достижимость — важная задача сетей Петри. Другой задачей, получившей много внимания в публикациях по сетям Петри, является активность. Как отмечено в разд. 2.4.1.4, активность связана с тупиками. Мы коснемся здесь двух задач, связанных с активностью сети Петри $S = (P, T, I, O)$ с начальной маркировкой μ . Сеть Петри активна, если активен всякий ее переход. Переход t_j активен в маркировке μ , если для всякой маркировки $\mu' \in R(S, \mu)$ существует последовательность σ , такая, что t_j разрешен в $\delta(\mu', \sigma)$. Переход t_j пассивен в маркировке μ , если не существует достижимой маркировки, в которой бы он мог быть запущен.

Определение 7. *Задача активности.* Активны ли все переходы $t_j \in T$?

Определение 8. *Задача активности одного перехода.* Активен ли данный переход $t_j \in T$?

Очевидно, что задача активности сводится к задаче активности одного перехода. Для нахождения решения задачи активности просто решим

Кононюк А.Е. Графы

задачу активности одного перехода для каждого $t_j \in T$; если $|T| = m$, то мы должны решить m задач активности одного перехода. Задачу достижимости можно также свести к задаче активности. Поскольку варианты задачи достижимости эквивалентны, рассмотрим задачу достижимости нуля в одной позиции. Если перед нами стоят какие-либо другие задачи достижимости, их можно свести, как показано в разд. 2.5.2, к задаче достижимости нуля в одной позиции. Теперь, если мы хотим определить, может ли быть позиция p_i нулевой в какой-либо достижимой маркировке для сети Петри $C_1 = (P_1, T_1, I_1, O_1)$ с начальной маркировкой μ_1 , то построим сеть Петри $C_2 = (P_2, T_2, I_2, O_2)$ с начальной маркировкой μ_2 , которая будет активна тогда и только тогда, когда нулевая маркировка не будет достижима из μ_1 .

Сеть Петри C_2 строится из C_1 введением двух позиций r_1 и r_2 и трех переходов s_1, s_2 и s_3 . Сначала модифицируем все переходы T_1 , включая r_1 в качестве входа и выхода. Начальная маркировка μ_2 будет включать фишку в r_1 . Позиция r_1 — это позиция «действия», пока фишка остается в r_1 , переходы T_1 могут запускаться. Следовательно, любая маркировка, достижимая в C_1 , достижима также и в позициях P_1 в C_2 . Определим переход s_1 так, что его входом будет r_1 , а выход пуст. Это позволяет удалить фишку из r_1 , запрещая запуск всех переходов в T_1 и «замораживая» маркировку P_1 . (Заметим, что все переходы T_1 находятся в конфликте и не только по определению, но и по построению могут запускаться каждый раз не более чем по одному.)

Позиция r_1 и переход s_1 позволяют сети C_1 достичь любой достижимой маркировки, затем запуском s_1 заморозить сеть в этой маркировке. Далее необходимо проверить, является ли позиция p_i нулевой. Введем новую позицию r_2 и переход s_2 , имеющий в качестве входа p_i , а в качестве выхода r_2 . Если p_i может когда-либо стать нулевой, то этот переход не является активным. В действительности вся сеть будет пассивной, если в этой маркировке сработает переход s_1 . Следовательно, если p_i может быть пустой, сеть не является активной. Если p_i не может быть пустой, тогда s_2 всегда может быть запущен, помещая фишку в r_2 . В этом случае мы должны будем вернуть фишку в r_1 и гарантировать, что все переходы в C_2 активны. Необходима уверенность в том, что C_2 активна, даже если C_1 не является активной. Это обеспечивается переходом s_3 , который «наполняет» сеть C_2 фишками, гарантируя тем самым, что, если фишка помещена в r_2 , каждый переход активен. Переход s_3 в качестве входа имеет r_2 , а в качестве выхода все позиции C_2 (все p_i в C_1, r_1 и r_2). Эта конструкция иллюстрируется рис. 6.

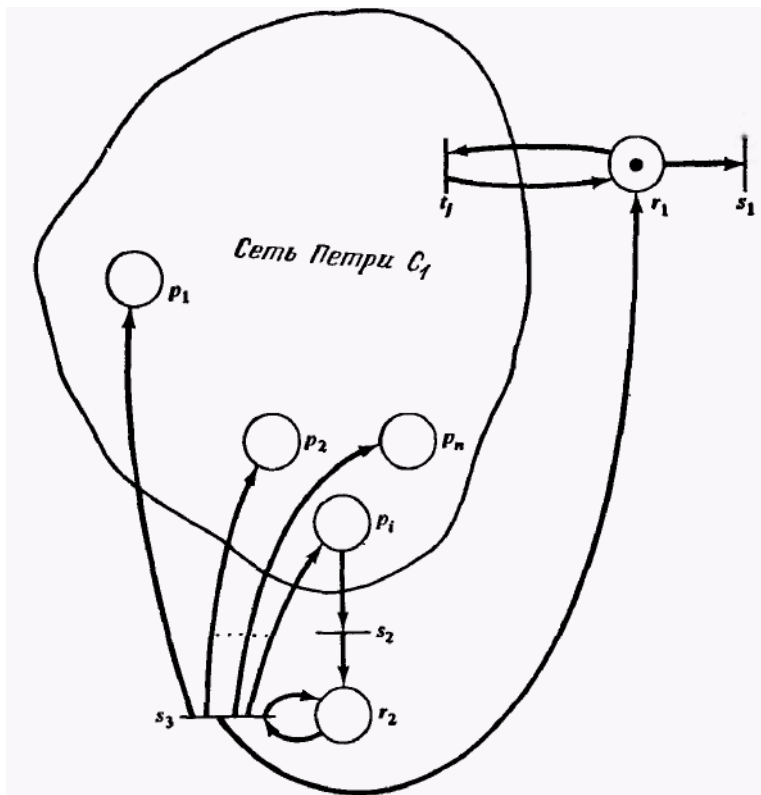


Рис. 6. Конструкция, переводящая задачу достижимости нуля в одной позиции (достижима ли маркировка с $\mu(p_i) = 0$?) в задачу активности (является ли сеть активной?).

Далее, если маркировка μ с $\mu(p_i) = 0$ достижима в $R(C_1, \mu_1)$, тогда C_2 также может достичь этой маркировки в позициях P_i путем выполнения той же самой последовательности запусков переходов. Затем можно запустить s_1 , замораживая подмножество C_1 . Поскольку $\mu(p_i) = 0$, s_2 запустить нельзя и C_2 пассивна. Таким образом, если p_i может стать нулевой — C_2 неактивна.

Справедливо обратное, если C_2 неактивна, тогда должна быть достижима маркировка μ с $\mu(p_i) = 0$, из которой недостижимо состояние с фишкой в r_2 . (Если в r_2 есть фишка, то s_3 разрешен, а повторно запуская s_3 достаточное число раз, можно разрешить любой (или все) переход, т. е. сеть активна.) Если r_2 не имеет фишек и не

может их получить, тогда маркировка p_i также должна быть нулевой. Таким образом, если S_2 неактивна, тогда достижима маркировка, в которой маркировка p_i нулевая.

На основе этой конструкции мы доказали следующую теорему.

Теорема 5. Задача достижимости сводится к задаче активности. Для доказательства основного утверждения раздела покажем следующее.

Теорема 6. Задача активности одного перехода сводится к задаче достижимости.

Доказательство того, что задача активности одного перехода сводима к задаче достижимости, опирается на проверку достижимости любой из конечного множества *максимальных пассивных для t_j подмаркировок*. Сеть Петри не активна для перехода t_j тогда и только тогда, когда достижима некоторая маркировка, в которой переход t_j не запускаем и не может стать запускаемым. Маркировка такого вида называется *пассивной для t_j* . Для любой маркировки μ можно проверить, является ли она пассивной для t_j построением дерева достижимости с корнем μ и проверкой, можно ли где-либо в дереве запустить переход t_j . Если нельзя, то μ пассивна для t_j . Проверка активности t_j в таком случае требует проверки достижимости какой-либо пассивной для t_j маркировки.

В общем случае, однако, может существовать бесконечное число пассивных для t_j маркировок и бесконечное множество маркировок, в котором находятся пассивные для t_j маркировки. Заметив два свойства, сведем множество маркировок, которые необходимо проверить для достижимости, к конечному числу. Во-первых, если маркировка μ пассивна для t_j , то и любая маркировка $\mu' \leq \mu$ пассивна для t_j . (Любая последовательность запусков, возможная из μ' , возможна также из μ , поэтому если μ' может привести к запуску t_j , то это может и μ .) Во-вторых, маркировки некоторых позиций не будут влиять на пассивность для t_j данной маркировки, поэтому маркировки этих позиций являются «несущественными», они могут быть произвольными. Заимствуя прием из построения дерева достижимости, заменим «несущественные» компоненты на ω , показывая, что в этих позициях может быть произвольно большое число фишек, не влияющих на пассивность маркировки для t_j . Теперь, поскольку любая $\mu' \leq \mu$ пассивна для t_j , если μ пассивна для t_j , нам не нужно рассматривать позиции p_i с $\mu(p_i) = \omega$. Это означает, что мы применяем задачу достижимости подмаркировки с $P' = \{p_i \mid \mu(p_i) \neq \omega\}$. Рассмотрим в качестве примера сеть Петри на рис. 7.

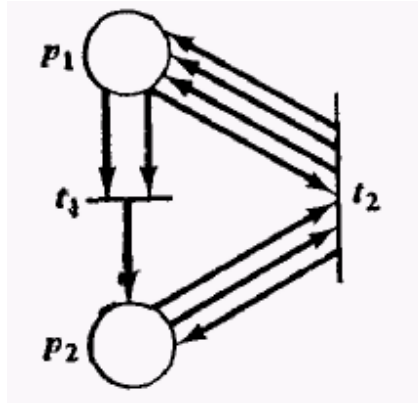


Рис. 7. Сеть Петри, иллюстрирующая пассивные маркировки для t_1 .

Маркировки $(2, 0)$, $(1, 0)$, $(0, 0)$, $(0, 1)$, $(0, 2)$, $(0, 3)$,... являются пассивными для t_2 , но их можно представить конечным образом множеством $\{(2, 0), (1, 0), (0, 0)\}$.

Хэк показал, что для сети Петри S существует такое конечное множество D_t маркировок (расширенных, т. е. включающих ω), что S активна тогда и только тогда, когда никакая маркировка из D_t недостижима. Если маркировка из D_t содержит ω , тогда подразумевается достижимость подмаркировки.

Более того, D_t можно эффективно вычислять. Поскольку D_t конечно, не- ω -компоненты имеют верхнюю границу b . Граница b определяется как такое наименьшее число, что если пассивна для t_j любая маркировка μ , такая, что $\mu(p_i) \leq b + 1$ для всех p_i , то является пассивной для t_j и подмаркировка μ' , такая, что $\mu'(p_i) = \mu(p_i)$, если $\mu(p_i) < b$, и $\mu'(p_i) = \omega$, если $\mu(p_i) = b + 1$. При таком определении b можно построить D_t следующим образом.

1. Вычислить b . Начать с $b = 0$, увеличивать b до тех пор, пока не окажется, что b удовлетворяет описанному определению границы. Проверка каждого b требует проверки всех $(b + 2)^n$ маркировок с компонентами, меньшими или равными $b + 1$.
2. Вычислить D_t проверкой всех маркировок и подмаркировок с компонентами, не превышающими b или равными ω . D_t — это множество пассивных для t_j маркировок из множества $(b + 2)^n$ маркировок.

Построив D_t , можно рассматривать задачу достижимости подмаркировки для каждого элемента D_t . Если какой-либо элемент D_t

Кононюк А.Е. Графы

достигим из начальной маркировки, сеть Петри неактивна, если же никакой элемент D_i недостижим — сеть Петри активна.

Из доказанных теорем мы получаем следующую.

Теорема 7. Следующие задачи эквивалентны:

1. Задача достижимости.
2. Задача активности.
3. Задача активности одного перехода.

2.5.5. Неразрешимые задачи

В предыдущем разделе мы показали, что многие задачи достижимости и активности эквивалентны, но никакого результата относительно разрешимости этих задач еще не получили. Для того чтобы показать разрешимость, необходимо свести задачу для сетей Петри к задаче с известным решением, а для того, чтобы показать неразрешимость, нужно свести задачу, которая известна как неразрешимая, к задаче для сетей Петри. Первый важный результат такого рода был получен Рабином. Он показал неразрешимость задачи: выполняется ли $R(C_1, \mu_1) \subseteq R(C_2, \mu_2)$ для двух сетей Петри — C_1 с маркировкой μ_1 и C_2 с маркировкой μ_2 ? Позднее Хэк показал, что неразрешимой является и задача: выполняется ли $R(C_1, \mu_1) = R(C_2, \mu_2)$? Доказательство этих утверждений основано на *десятой проблеме Гильберта*. (Д. Гильберт на конференции математиков в 1900 г. поставил 23 проблемы, и та, на которую опирался Хэк, была десятой в списке).

Определение 9. Дан полином P от n переменных с целыми коэффициентами; существует ли такой вектор целых (x_1, x_2, \dots, x_n) , что $P(x_1, x_2, \dots, x_n) = 0$? Уравнение $P(x_1, x_2, \dots, x_n) = 0$ называется *диофантовым*.

В общем оно представляет собой сумму членов:

$$P(x_1, x_2, \dots, x_n) = \sum_i R_i(x_1, x_2, \dots, x_n),$$

$$R_i(x_1, x_2, \dots, x_n) = a_i \cdot x_{s_1} \cdot x_{s_2} \cdot \dots \cdot x_{s_n}.$$

Диофантовыми уравнениями являются $x_1 = 0$; $3x_1 \cdot x_2 + 6x_3 = 0$ и т. д.

В 1970 г. Матиясевич доказал, что десятая проблема Гильберта неразрешима: не существует общего алгоритма, определяющего, имеет ли произвольное диофантово уравнение корень (набор значений, для которых полином равен нулю). Этот результат служит основой доказательства того, что задача равенства множеств достижимости сетей Петри неразрешима. Стратегия его заключается в построении для

диофантова полинома сети Петри, которая (в определенном смысле) вычисляет все значения полинома.

2.5.5.1. Задача включения графов полиномов

Доказательство неразрешимости задачи равенства состоит из трех частей (рис. 8).

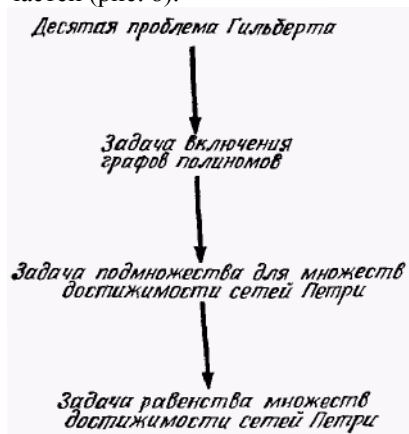


Рис. 8. Сведения, показывающие, что задача равенства (и подмножества) для множеств достижимости сетей Петри неразрешима.

Сначала десятая проблема Гильберта сводится к задаче включения графов полиномов. Затем задача включения графов полиномов сводится к задаче подмножества для множества достижимости сетей Петри. Наконец, задача подмножества для множеств достижимости сетей Петри сводится к задаче равенства множеств достижимости сетей Петри. Это показывает, что десятая проблема Гильберта, известная как неразрешимая, сводится к задаче равенства, которая поэтому также должна быть неразрешимой.

Определение 10. Граф $G(P)$ диофантова полинома $P(x_1, \dots, x_n)$ с неотрицательными коэффициентами — это множество

$$G(P) = \{(x_1, \dots, x_n, y) \mid y \leq P(x_1, \dots, x_n) \text{ и } 0 \leq x_1, \dots, x_n, y\}.$$

Определение 11. Задача включения графов полиномов заключается в определении для двух диофантовых полиномов A и B , выполняется ли $G(A) \subseteq G(B)$.

Кононюк А.Е. Графы

Покажем сначала, что десятая проблема Гильберта сводится к задаче включения графов полиномов.

Теорема 8. Задача включения графов полиномов неразрешима.

Доказательство:

1. Ограничим наше доказательство задачами с неотрицательными решениями. Если (x_1, \dots, x_n) — решение для $P(x_1, \dots, x_n) = 0$ с $x_i < 0$, то $(x_{1m}, \dots, -x_i, \dots, x_n)$ решение для $P(x_1, \dots, -x_i, \dots, x_n) = 0$. Следовательно, для определения того, является ли (x_1, \dots, x_n) решением произвольного полинома (то есть допускающего решения с отрицательными значениями переменных), необходимо только проверить каждый из 2^n полиномов, получающихся в результате изменения знака у некоторого подмножества переменных для неотрицательного решения.

2. Аналогично, поскольку $P^2(x_1, \dots, x_n) = 0$ тогда и только тогда, когда $P(x_1, \dots, x_n) = 0$, необходимо рассматривать только полиномы, значения которых неотрицательны.

3. Сейчас можно разбить любой полином $P(x_1, x_2, \dots, x_n)$ на два полинома, $Q_1(x_1, \dots, x_n)$ и $Q_2(x_1, \dots, x_n)$, такие, что $P(x_1, \dots, x_n) = Q_1(x_1, \dots, x_n) - Q_2(x_1, \dots, x_n)$, помещая все члены с положительными коэффициентами в Q_1 , а все члены с отрицательными коэффициентами в Q_2 . Далее, поскольку $P(x_1, \dots, x_n) \geq 0$ (по п. 2), имеем, что $Q_1(x_1, \dots, x_n) \geq Q_2(x_1, \dots, x_n)$ и $P(x_1, \dots, x_n) = 0$ тогда и только тогда, когда $Q_1(x_1, \dots, x_n) = Q_2(x_1, \dots, x_n)$.

4. Рассмотрим два графа полиномов:

$$G(Q_1) = \{(x_1, \dots, x_n, y) \mid y \leq Q_1(x_1, \dots, x_n)\},$$

$$G(Q_2 + 1) = \{(x_1, \dots, x_n, y) \mid y \leq 1 + Q_2(x_1, \dots, x_n)\}.$$

Теперь $G(Q_2 + 1) \subseteq G(Q_1)$ тогда и только тогда, когда для всех неотрицательных x_1, \dots, x_n и y из $y \leq 1 + Q_2(x_1, \dots, x_n)$ следует, что $y \leq Q_1(x_1, \dots, x_n)$. Это справедливо тогда и только тогда, когда не существует x_1, \dots, x_n и y , таких, что

$$Q_1(x_1, \dots, x_n) < y \leq 1 + Q_2(x_1, \dots, x_n).$$

Но из п. 3 следует, что $Q_1 \geq Q_2$, поэтому

$$Q_1(x_1, \dots, x_n) < y \leq 1 + Q_2(x_1, \dots, x_n) \leq 1 + Q_1(x_1, \dots, x_n),$$

а поскольку все величины целые,

$$y = 1 + Q_2(x_1, \dots, x_n) = 1 + Q_1(x_1, \dots, x_n),$$

что справедливо тогда и только тогда, когда $Q_1 = Q_2$. Таким образом, мы убедились в том, что $G(Q_2 + 1) \subseteq G(Q_1)$ тогда и только тогда, когда не существует таких x_1, \dots, x_n , для которых $P(x_1, \dots, x_n) = 0$.

5. Итак, для определения того, что уравнение $P(x_1, x_2, \dots, x_n) = 0$ имеет решение, необходимо показать только, что не выполняется $G(Q_2 + 1) \subseteq G(Q_1)$.

2.5.5.2. Слабое вычисление

Теперь нам необходимо показать, что сети Петри могут (в определенном смысле) вычислять значение полинома $Q(x_1, x_2, \dots, x_n)$. Мы осмотрительно ограничили полином Q до неотрицательных значений полинома, неотрицательных коэффициентов и неотрицательных значений переменных. Это позволяет нам представить значения переменных и значение полинома числом фишек в позициях сети Петри. Общая схема показана на рис. 9.

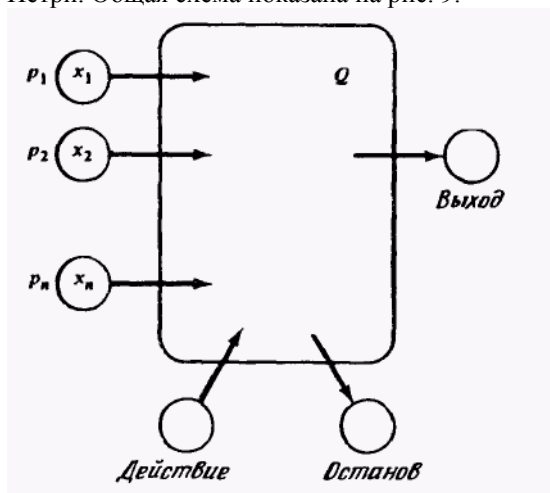


Рис. 9. Базовая структура сети Петри, слабо вычисляющей значение полинома $Q(x_1, x_2, \dots, x_n)$.

Входные значения x_1, x_2, \dots, x_n представляются x_i фишками в p_i для $i = 1, \dots, n$. Первоначально фишка помещается также в позицию «действия». Выполнение сети будет закончено помещением фишки в позицию останова. В это время «выходная» позиция будет иметь y фишек, где $y \leq Q(x_1, \dots, x_n)$.

Эта сеть Петри *слабо вычисляет* значение $Q(x_1, \dots, x_n)$. Слабое вычисление означает, что вычисленное значение не будет превышать

Кононюк А.Е. Графы

$Q(x_1, \dots, x_n)$, но может быть любым (неотрицательным) значением, меньшим $Q(x_1, \dots, x_n)$. Слабое вычисление обязательно для сетей Петри вследствие *разрешающей* природы запусков переходов: сеть Петри нельзя заставить остановиться. При определении графа полинома $G(Q)$ это особенно учитывалось.

Покажем, что можно построить подсеть, слабо вычисляющую функцию умножения (двух чисел). На ее основе мы можем построить составную сеть, которая слабо вычисляет значение каждого члена полинома путем последовательной композиции подсетей умножения. Выход подсети для каждого члена будет помещаться в выходную позицию для полинома. Таким образом, число фишек в выходной позиции будет суммой выходов для каждого члена.

Подсеть умножения показана на рис. 10.

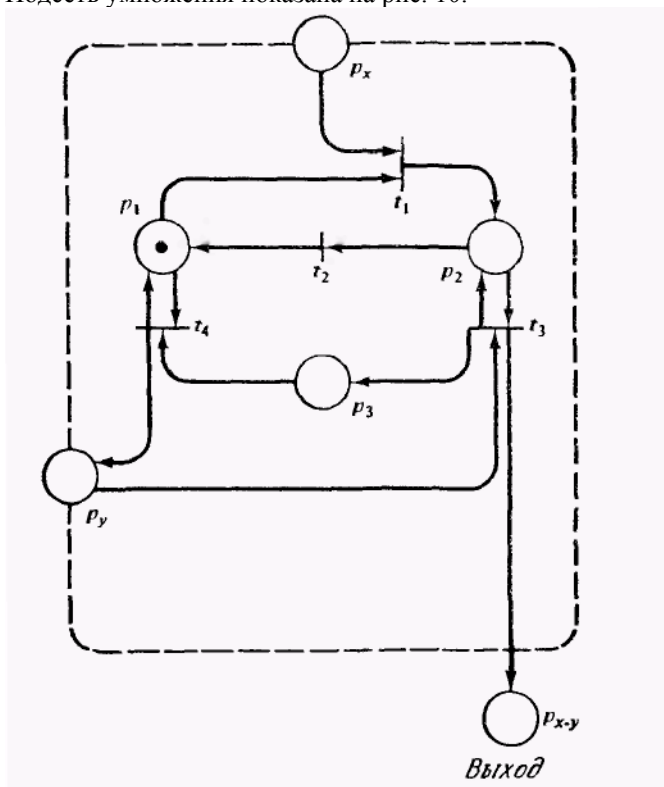


Рис. 10. Подсеть умножителя. Эта подсеть слабо вычисляет произведение x и y .

Кононюк А.Е. Графы

Эта сеть слабо вычисляет произведение чисел x и y , представимых фишками в ее входных позициях, помещая множество фишек в свой выход. Действует сеть совсем просто. Для вычисления произведения x и y сначала запускается переход t_1 , перемещая одну фишку из p_1 в p_2 . Эта фишка разрешает запуск перехода t_3 , который может теперь копировать y фишек из позиции p_y , помещая их в p_3 и вкладывая в выходную позицию $p_{x \cdot y}$ фишек. Теперь можно запустить t_2 , возвращая фишку из p_2 в p_1 . Это разрешает запуск t_4 , который копирует y фишек из p_3 обратно в p_y . Весь этот процесс можно повторить точно x раз, помещая каждый раз в $p_{x \cdot y}$ y фишек. После этого маркировка p_x становится нулевой и сеть останавливается. Общее число фишек в позиции $p_{x \cdot y}$ равно произведению x и y .

Мы описали наилучший случай в том смысле, что число выходных фишек в точности равно $x \cdot y$. Однако фишка в p_2 разрешает и переход t_3 , и переход t_2 , а переход t_2 можно запустить до того, как все y фишек будут скопированы из p_y в p_3 и добавлены в $p_{x \cdot y}$. В этом случае число фишек, помещенных в $p_{x \cdot y}$, будет меньше, чем $x \cdot y$. Поскольку t_3 можно запустить не более y раз для каждого запуска t_1 , а t_1 можно запустить не более x раз, мы можем гарантировать, что число фишек в $p_{x \cdot y}$ никогда не превысит $x \cdot y$, но вследствие разрешающей природы запусков переходов мы не можем гарантировать, что число фишек в $p_{x \cdot y}$ будет в точности равно $x \cdot y$; оно может быть меньше. Следовательно, эта сеть Петри слабо вычисляет произведение x и y . Теперь для того, чтобы слабо вычислить член R_i , являющийся произведением $a_1 \bullet x_{s_1} \bullet x_{s_2} \bullet \dots \bullet x_{s_k}$, построим сеть Петри показанного на рис. 11 вида.

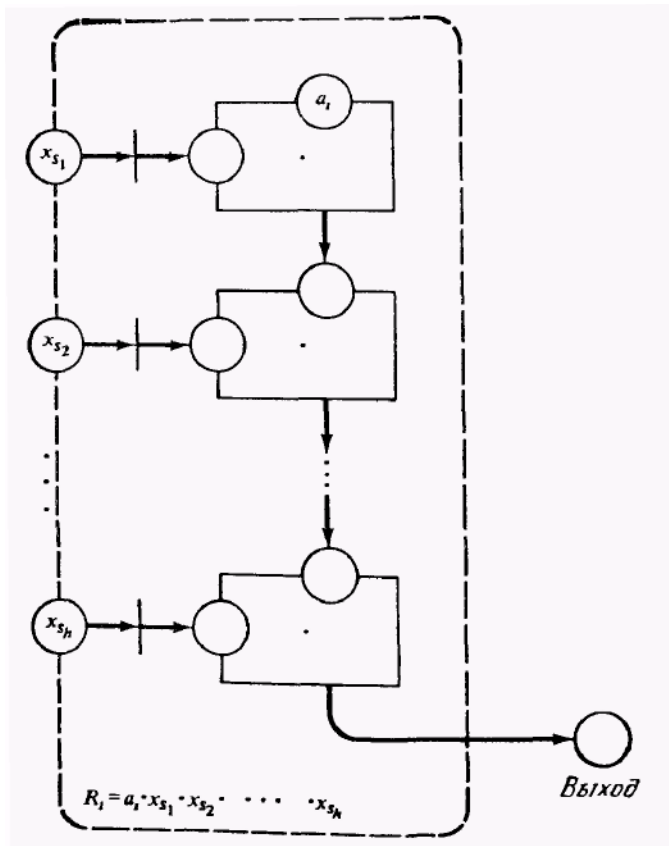


Рис. 11. Сеть Петри, слабо вычисляющая член диофантова уравнения. Каждый блок в сети имеет вид, изображенный на рис. 10.

Поскольку каждая подсеть слабо вычисляет произведение двух членов, вся подсеть слабо вычисляет значение члена R_i .

Далее на рис. 12 показано, как можно слабо вычислять полином $P = R_1 + R_2 + \dots + R_k$.

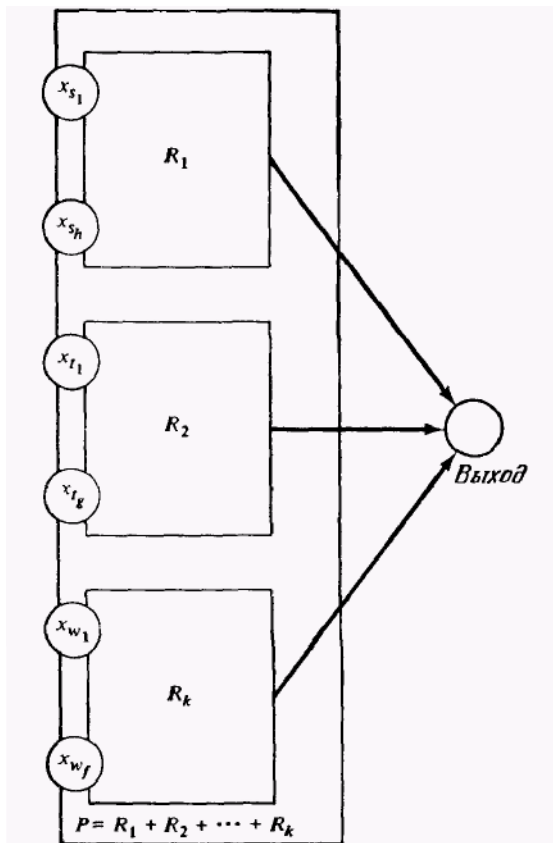


Рис. 12. Сеть Петри, слабо вычисляющая $P(x_1, x_2, \dots, x_n)$ путей использования набора подсетей вида, изображенного на рис. 11.

Каждая подсеть имеет вид, изображенный на рис. 11, и слабо вычисляет значение одного члена полинома. Выходы k подсетей для отдельных членов объединяются вместе, давая общее значение суммы. Теперь для создания конкретных необходимых множеств достижимости вводится несколько управляющих переходов и позиций. Сначала необходимо обеспечить получение произвольного значения для каждой из переменных (x_1, \dots, x_n) и запись этого значения в позиции p_1, \dots, p_n . Для каждой p_i создается переход t_i с пустым входом и выходами в p_i и всякую позицию, являющуюся входом,

Кононюк А.Е. Графы

соответствующим x_i в члене R_j , использующем x_i . Следовательно, в полиноме $x_1 + x_1x_2$ мы должны иметь переход t_1 с выходами в p_1 и во входы x_1 в двух членах x_1 и x_1x_2 , использующих x_1 ; t_2 будет иметь выходы в p_2 и во вход x_2 в члене x_1x_2 .

Эти переходы могут запускаться произвольное число раз, создавая любые значения в p_1, \dots, p_n . Следовательно, для всякого $y \leq P\{x_1, \dots, x_n\}$ достижима маркировка μ с $\mu(p_i) = x_i, \dots, \mu(p_n) = x_n$ и $\mu(\text{выход}) = y$. Значение $y = P(x_1, \dots, x_n)$ может быть достигнуто сначала запуском x_1 раз перехода t_1 , помещая x_1 фишек в p_1 , затем запуском x_2 раз перехода t_2 и т. д., пока не будет запущен x_n раз переход t_n . Далее можно выполнить подсеть для каждого члена R_j полинома, в результате чего значение полинома поместится в выходную позицию.

Для сведения задачи включения графов полиномов к задаче подмножества выполним следующие шаги. Пусть мы хотим определить для полиномов A и B , выполняется ли $G(A) \subseteq G(B)$.

1. Построим сеть Петри C_A , слабо вычисляющую $A(x_1, \dots, x_n)$, и сеть Петри C_B , слабо вычисляющую $B(x_1, \dots, x_n)$.

2. Если число позиций в этих сетях не равно, добавим позиции в ту сеть, где их меньше с тем, чтобы уравнивать число позиций. Эти позиции не имеют начальной маркировки и не связаны с какими-либо переходами в сети.

3. Теперь мы должны устранить влияние всех внутренних позиций на множества достижимости. И в C_A , и в C_B различно множество из $n+1$ позиций — n позиций, соответствующих значениям x_1, \dots, x_n , и $(n+1)$ -я позиция, соответствующая выходу каждой сети. Все другие позиции являются внутренними, маркировка их неважна. Однако может оказаться, что для внутренней позиции p_i в C_A и соответствующей позиции p'_i в C_B существуют неравные маркировки μ в $R(C_A, \mu_A)$ и μ' в $R(C_B, \mu_B)$, поскольку $\mu(p_i) \neq \mu'(p'_i)$ для всех μ'' в $R(C_B, \mu_B)$.

Для того чтобы обойти это препятствие, введем по две новые позиции: q и r в C_A (получая C'_A) и q' и r' в C_B (получая C'_B). В C'_A позиции q и r не связаны с переходами и первоначально r пуста, а q имеет одну фишку. Позиция r' в C_B служит позицией «действия». Она получает начальную маркировку (одну фишку), а каждый переход модифицируется с тем, чтобы включать r' в качестве входа и в качестве выхода. Следовательно, пока фишка остается в r' , сеть C'_B может функционировать, как и прежде. Введем в C'_B новый переход, переводящий разрешающую фишку из r' в q' , запрещая все переходы C'_B и «замораживая» маркировку. Теперь для каждой внутренней позиции (T' и q' не считаются внутренними) введем по два новых перехода.

Кононюк А.Е. Графы

Один из этих переходов, вводимых для каждой внутренней позиции p_i , маркировка которой неважна, имеет в качестве входов позиции q' к p_i , а в качестве выхода — только q' (уменьшая при запуске на 1 маркировку в p_i), другой переход в качестве входа имеет q' , а в качестве выхода и q' , и p_i (увеличивая при запуске на 1 маркировку в p_i). Эти переходы благодаря соответствующей последовательности прибавляющих или убавляющих запусков обеспечивают каждой внутренней позиции возможность иметь произвольную маркировку.

4. Описанная конструкция показана на рис. 13.

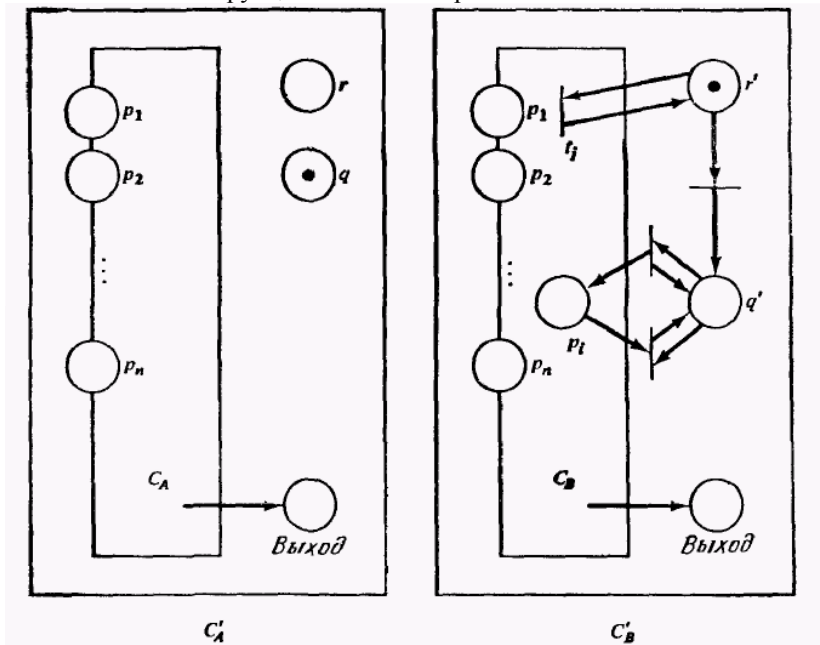


Рис. 13. Сеть Петри, построенная для проверки включения графов полиномов

Для двух сетей Петри C'_A и C'_B с начальной маркировкой μ'_A и μ'_B соответственно, $G(A) \subseteq G(B)$ тогда и только тогда, когда

$$R(C'_A, \mu'_A) \subseteq R(C'_B, \mu'_B).$$

C'_A и C'_B имеют следующие множества достижимости:

Кононюк А.Е. Графы

$C'_A:$	$p_1 \dots p_n$	Выход	r	g	Внутренние позиции
	$x_1 \dots x_n$	$y \leq A(x_1, \dots, x_n)$	0	1	Некоторая произвольная маркировка
$C'_B:$	$p_1 \dots p_n$	Выход	r'	g'	Внутренние позиции
	$x_1 \dots x_n$	$y \leq B(x_1, \dots, x_n)$	1	0	Некоторая произвольная маркировка
	$x_1 \dots x_n$	$y \leq B(x_1, \dots, x_n)$	0	1	Произвольная маркировка

Следовательно, если $G(A) \subseteq G(B)$, то $R(C'_A, \mu'_A) \subseteq R(C'_B, \mu'_B)$. и, наоборот, если $R(C'_A, \mu'_A) \subseteq R(C'_B, \mu'_B)$, то $G(A) \subseteq G(B)$.

Таким образом, мы показали справедливость следующего.

Теорема 9. Задача включения графов полиномов сводима к задаче подмножества для множеств достижимости сетей Петри.

2.5.5.3. Задача равенства

Теперь нам осталось только показать, что задача подмножества для множеств достижимости сетей Петри сводится к задаче равенства.

Предположим, что имеются две сети Петри A и B , и мы хотим определить, выполняется ли $R(A, \mu_A) \subseteq R(B, \mu_B)$ (задача подмножества).

Покажем сейчас, что можно определить такие две сети Петри D и E , что $R(A, \mu_A) \subseteq R(B, \mu_B)$ тогда и только тогда, когда $R(D, \mu_D) = R(E, \mu_E)$.

Основой построения доказательства служит тот факт, что

$R(A, \mu_A) \subseteq R(B, \mu_B)$ тогда и только тогда, когда

$$R(B, \mu_B) = R(A, \mu_A) \cup R(B, \mu_B) .$$

D и E строятся из общей подсети C . Сеть C представляет множества достижимости A и B для получения их объединения. Конструкция иллюстрируется рис.14.

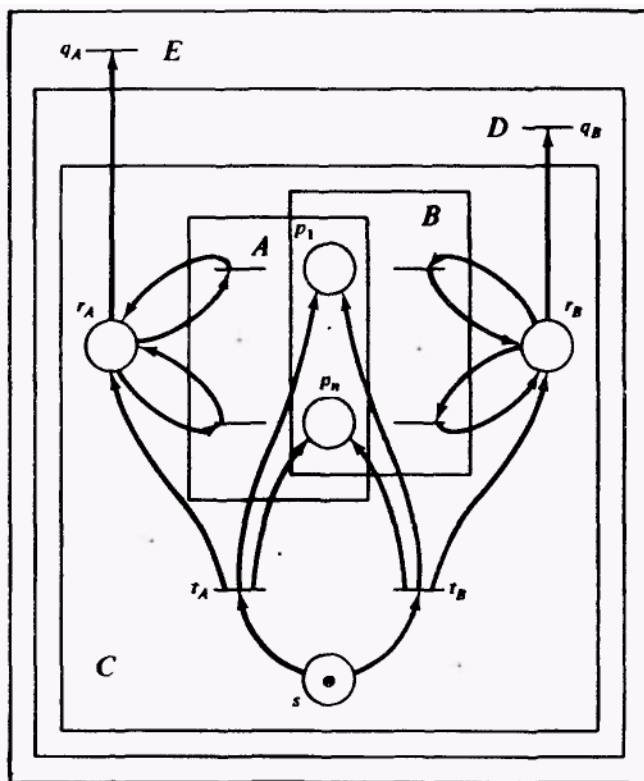


Рис. 14. Построение сетей Петри C , D и E из A и B , используемое для доказательства сводимости задачи подмножества для множеств достижимости к задаче равенства.

Позиции p_1, \dots, p_n действуют или как позиции сети A , или как позиции сети B . Первоначально они не имеют маскировки. Вводятся две новые позиции r_A и r_B , служащие позициями действия для сети A и сети B соответственно. Все переходы сети A модифицируются с тем, чтобы включать в качестве входа и выхода r_A , а все переходы сети B модифицируются с тем, чтобы включать в качестве входа и выхода r_B . Далее вводятся еще одна позиция s , и два новых перехода, t_A и t_B . Начальная маркировка всей сети (включая A и B как подсети с общими позициями; позиции r_A , r_B и s ; переходы t_A и t_B) определяется одной фишкой в s и нулем в остальных. Переход t_A имеет в качестве входа s , а выход его порождает начальную маркировку сети A плюс фишку в r_A ; переходов в качестве входа также имеет s , а выход его создает

Кононюк А.Е. Графы

начальную маркировку сети B плюс фишку в r_6 . Следовательно, если запустится t_A , то подсеть A приобретает свою начальную маркировку, и все ее переходы могут запускаться, как обычно, поскольку в r_A имеется фишка. Подсеть B , однако, полностью запрещена, поскольку фишки в r_B нет. Если первым запустится t_B , то сможет действовать подсеть B , а подсеть A будет запрещена. Тогда множество последовательностей для C — это любая последовательность вида

$$t_A' \langle \text{любая последовательность запусков в } A \rangle$$

или любая последовательность вида

$$t_B' \langle \text{любая последовательность запусков в } B \rangle.$$

Сеть D получается из C введением одного нового перехода q_B . Он имеет в качестве входа позицию r_B , выход его пуст. Заметим, что q_B может запуститься, только если первым был запущен переход t_B , если же первым запустится переход t_A , то r_B будет пустой и q_B не сможет запуститься.

Сеть E строится из D добавлением нового перехода q_A . Он в качестве входа имеет позицию r_A , выхода не имеет. Переход может запуститься, только если первым был запущен t_A . Подчеркнем, что сеть E строится из D , а не (непосредственно) из C . Поэтому E имеет и переход q_A , и переход q_B .

Рассмотрим теперь множества достижимости сетей Петри C , D и E . Множество достижимости C — это все маркировки вида:

s	r_A	r_B	p_1, \dots, p_n
1	0	0	$0, \dots, 0$
0	1	0	Любая маркировка $\mu \in R(A, \mu_A)$ (если запустится t_A)
0	0	1	Любая маркировка $\mu \in R(B, \mu_B)$ (если запустится t_B)

Сеть Петри D добавляет к этому множеству один новый класс маркировок:

Кононюк А.Е. Графы

s	r_A	r_B	p_1, \dots, p_n
1	0	0	$0, \dots, 0$
0	1	0	Любая маркировка $\mu \in R(A, \mu_A)$ (если запустится t_A)
0	0	1	Любая маркировка $\mu \in R(B, \mu_B)$ (если запустится t_B)
0	0	0	Любая маркировка $\mu \in R(B, \mu_B)$ (если запустится q_B)

Сеть Петри E добавляет еще один класс

s	r_A	r_B	p_1, \dots, p_n
1	0	0	$0, \dots, 0$
0	1	0	Любая маркировка $\mu \in R(A, \mu_A)$ (если запустится t_A)
0	0	1	Любая маркировка $\mu \in R(B, \mu_B)$ (если запустится t_B)
0	0	0	Любая маркировка $\mu \in R(B, \mu_B)$ (если запустится q_B)
0	0	0	Любая маркировка $\mu \in R(A, \mu_A)$ (если запустится q_A)

Теперь, если $R(A, \mu_A) \subseteq R(B, \mu_B)$, то последний класс в $R(E, \mu_E)$ (маркировки вида $(0, 0, 0, \mu)$, где $\mu \in R(A, \mu_A)$) включен в последний класс $R(D, \mu_D)$ (маркировки вида $(0, 0, 0, \mu)$, где $\mu \in R(B, \mu_B)$). Поскольку все другие маркировки совпадают, то $R(D, \mu_D) = R(E, \mu_E)$, если $R(A, \mu_A) \subseteq R(B, \mu_B)$. Аналогично, если $R(D, \mu_D) = R(E, \mu_E)$, то $R(A, \mu_A) \subseteq R(B, \mu_B)$, так как для всякой маркировки $(0, 0, 0, \mu)$, где $\mu \in R(A, \mu_A)$, в $R(E, \mu_E)$ должна существовать такая же маркировка в $R(D, \mu_D)$. Но все маркировки с $\mu(s, r_A, r_B) = (0, 0, 0)$ имеют вид $(0, 0, 0, \mu)$, где $\mu \in R(B, \mu_B)$, поэтому $R(A, \mu_A) \subseteq R(B, \mu_B)$.

Таким образом, на основе этой конструкции доказано следующее.

Теорема 10. Задача подмножества для множеств достижимости сетей Петри сводима к задаче равенства для множеств достижимости сетей Петри.

Последние три теоремы приводят к следующему результату.

Теорема 11. Следующие задачи неразрешимы:

1. Задача включения графов полиномов.
2. Задача подмножества для множеств достижимости сетей Петри.
3. Задача равенства для множеств достижимости сетей Петри.

Эти теоремы и их доказательства принадлежат Хэку.

2.5.6. Сложность задачи достижимости

Поскольку задачи подмножества и равенства для множеств достижимости сетей Петри неразрешимы, то возможно, что неразрешима также и сама задача достижимости. Однако в настоящее время вопрос, разрешима ли (или неразрешима) задача достижимости, открыт. Пока не существует ни алгоритма, решающего задачу достижимости, ни доказательства того, что такого алгоритма не может быть.

В 1977 г на Симпозиуме АСМ по теории вычислений было представлено «доказательство» разрешимости задачи достижимости. Однако это «доказательство» имело несколько серьезных огрехов, а попытки устранить их, чтобы получить корректное доказательство, не дали положительных результатов. И все же превалирует ощущение, что задача достижимости разрешима, что алгоритм ее решения существует, и со временем он будет найден.

Если предположить, что алгоритм решения задачи достижимости существует, то можно утверждать, что он очень сложный. Уместен вопрос какова сложность алгоритма решения задачи достижимости? Некоторые оценки его сложности можно получить, не ссылаясь на конкретный алгоритм

Липтон показал, что любой алгоритм, решающий задачу достижимости, потребует по крайней мере экспоненциального (2^{cn}) объема памяти и экспоненциального времени решения. Экспонента от n — это мера размерности задачи, и случай Липтона отражает число позиций и их взаимосвязи с переходами.

Необходимость в экспоненциальном объеме Липтон доказал, показав, что можно построить сеть Петри, в которой позиция действует как счетчик чисел $0, 1, \dots, 2^{2^n}$. При таком представлении алгоритм решения задачи достижимости потребует $\log_2(2^{2^{2^n}}) = 2^n$ бит. При этом в его конструкции используется не более $h \cdot n$ позиции (для некоторой константы K).

Доказательство Липтона основано на возможности построения сети, пересчитывающей до 2^{2^n} фишек только в $h \cdot n$ позициях. Для проверки позиции на нуль необходимо наложить на сеть некоторые

Кононюк А.Е. Графы

ограничения. Можно предложить, разумеется, сети Петри, для которых непосредственная проверка нуля невозможна. Тем не менее для осуществления возможности проверки нуля используется общий метод, заключающийся в использовании таких двух позиций p и p' , что $\mu(p) + \mu(p')$ является константой. Если мы знаем, что $\mu(p) + \mu(p') = k$, тогда можно проверить, является ли $\mu(p)$ нулем, проверяя, имеет ли $\mu(p')$ k фишек; если $\mu(p')$ имеет k фишек, то $\mu(p)$ имеет нуль фишек, и наоборот. То, что маркировка позиции не нулевая, можно проверить с помощью петли. Заметим, что для поддержания способности проверки позиции на нуль необходимо поддерживать свойство постоянности $\mu(p) + \mu(p')$, т. е. сеть должна быть сохраняющей по крайней мере по отношению к этим двум позициям.

Для небольших чисел k можно проверить, равна ли маркировка позиции числу k , делая позицию входом перехода k раз (рис. 15).

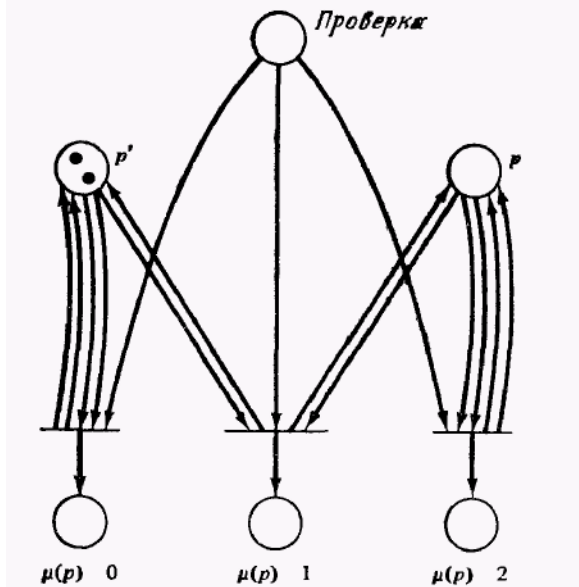


Рис. 15. Проверка маркировки ограниченной позиции на совпадение с 0, 1 или 2. Все переходы должны поддерживать сумму маркировки.

Однако эти дуги увеличивают размерность задачи, поэтому в общем случае так поступать нельзя. Липтон показал, что если постоянная сумма двух позиций (p_k, p'_k) равна k и k является произведением двух целых сомножителей $k = k_1 \cdot k_2$, которые являются постоянными суммами двух других пар позиций (p_{k_1}, p'_{k_1}) и (p_{k_2}, p'_{k_2}) , и мы можем

Кононюк А.Е. Графы

проверить, выполняется ли $p_{k_1} = 0$ и $p_{k_2} = 0$, то можно проверить, выполняется ли $\mu(p) = 0$. Это позволило Липтону строить подсети так, как показано на рис. 16.

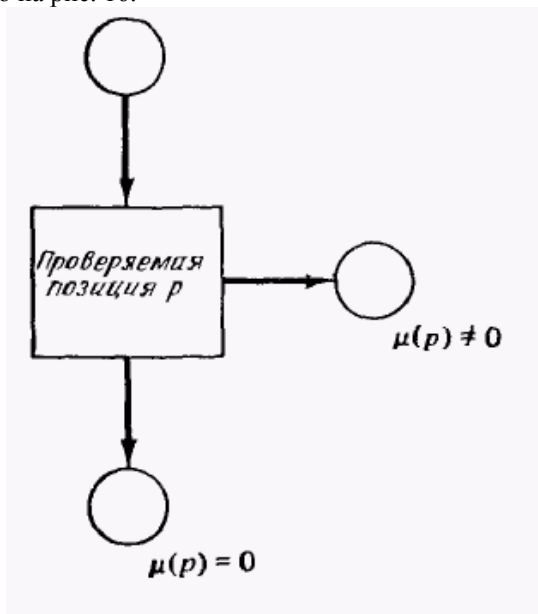


Рис. 16. Вид сетей Петри, используемых Липтоном для построения больших сетей, допускающих проверку большого счетчика на нуль.

Такие сети затем использовались для контроля сетей умножения, подобных сетям, используемым для слабого вычисления графов полиномов (см. рис. 10). Проверка на нуль подсети позволяет сети Петри вычислять точное произведение (а не слабое произведение, которое только ограничено сверху действительным произведением).

Эти простые сети позволили Липтону построить сеть, которая для данного n может порождать точно 2^{2^n} фишек в позиции (p) с нулем фишек в p' и в которой можно проверить $\mu(p)$ на равенство нулю. Число используемых позиций равно произведению константы на n . Существование сети Петри, подобной этой, показывает, что задача достижимости требует по крайней мере экспоненциальных времени и памяти, и, следовательно, затраты на вычисление ее решения слишком велики.

Кононюк А.Е. Графы

Конструкция сети Петри, которая может пересчитывать числа до 2^{2^n} , имеет еще и важное следствие. Построенная таким образом сеть Петри ограничена, поскольку число фишек в любой позиции не может превысить 2^{2^n} . Это означает, что любой алгоритм определения ограниченности сети Петри также должен требовать экспоненциальных времени и памяти. Следовательно, даже простые задачи для сетей Петри, хотя и являются разрешимыми, могут требовать для поиска решения больших затрат времени и памяти.

Необходимо напомнить, что описанные границы являются нижними для наихудшего случая поведения алгоритма. Но может оказаться, что многие задачи решаются для большинства сетей Петри сравнительно эффективно. Оценка сложности показывает, что, даже если для большинства сетей алгоритм требует малых затрат памяти и времени, *существует* сеть Петри, которая для анализа потребует гораздо больше времени и объема памяти.

Хотя оценки сложности получены для наихудшего случая (это означает, что в среднем они намного лучше), они являются, кроме того, нижней границей. Мы знаем, что задача достижимости требует экспоненциального объема памяти, *не менее*. Возможно, что сложность определения достижимости даже не экспоненциальная. Ракофф предложил алгоритм определения ограниченности за экспоненциальное время, поэтому задача ограниченности известна теперь как имеющая экспоненциальную сложность. Задача достижимости известна как по крайней мере экспоненциально сложная (и, возможно, даже неразрешимая).

Результат Мэйра показал, что задачи подмножества и равенства для множеств достижимости *ограниченных* сетей Петри имеют сложность непримитивной рекурсии. Это означает, что некоторые задачи для сетей Петри, несмотря на разрешимость, требуют на вычисление значительных затрат.

Теория вычислимости — начало теории вычислений — была развита из работ Тьюринга, Клини, Гёделя и Чёрча. Девис и Минский предлагают хорошие введения в нее. Карп показал, как сводимость можно использовать для получения результатов по разрешимости и сложности.

Большинство результатов, изложенных в этой главе, взяты из работ Хэка. Хэк — один из основных исследователей по задачам разрешимости в сетях. Результаты по сложности получены Липтоном, Ракоффом и Джоунсом и др.

Кононюк А.Е. Графы

Сеть Петри называется *обратимой*, если для каждого перехода $t_j \in T$ найдется переход $t_k \in T$, такой, что

$$\#(p_i, I(t_j)) = \#(p_i, O(t_k)),$$

$$\#(p_i, O(t_j)) = \#(p_i, I(t_k)).$$

Иными словами, для каждого перехода существует другой переход с обратными входами и выходами. Это позволяет «уничтожать выполненное» последовательностью переходов путем запуска в обратном порядке их дополнительных переходов. Установлено, что для обратимых сетей Петри задачи достижимости и покрываемости разрешимы. Этот результат основан на работах по коммутативным полугруппам. Установите на основе взаимосвязи между обратимыми сетями Петри и коммутативными полугруппами разрешимость достижимости и эквивалентности для обратимых сетей Петри. Для развития теории обратимых сетей Петри рассмотрите также задачу активности, вопросы сложности и языки обратимых сетей Петри.

Представляется весьма полезной связь сетей Петри с *арифметикой Пресбургера*. Арифметика Пресбургера — это арифметическая теория, использующая сложение и вычитание целых чисел. Показано, что можно установить истинность или ложность всех высказываний, образованных из кванторов первого порядка, равенства, операций сложения и вычитания и целых чисел. Первоначальное доказательство использовалось в качестве основы для автоматического доказательства теорем. Предположительно арифметику Пресбургера можно использовать для решения задач анализа сетей Петри. Исследуйте этот вопрос.

2.6. ЯЗЫКИ СЕТЕЙ ПЕТРИ

Материал разделов 2.4, 2.5 касался в основном вопросов, связанных с задачей достижимости, т. е. с достижимыми маркировками. Родственный, но совершенно отличный подход заключается в рассмотрении не того, какие маркировки достижимы, а того, как их можно достичь. Поэтому главным объектом внимания являются переходы и, в частности, последовательности переходов, переводящих одну маркировку сети Петри в другую.

Последовательность переходов — это строка, а множество строк — язык. Таким образом, в этом разделе мы займемся языками, определяемыми сетями Петри, и их свойствами.

2.6.1. Предпосылки изучения языков сетей Петри

Развитие теории сетей Петри обуславливают две основные причины:

1) описание предлагаемых и существующих систем и 2) анализ систем, моделируемых сетями Петри. **Цель анализа сети Петри — определение свойств сети и моделируемой системы.** Одно из самых важных свойств системы — это множество действий, которые могут произойти. Множество всех возможных последовательностей действий *характеризует* систему.

Действия моделируются в сети Петри переходами, осуществление действия — запуском перехода. Последовательности действий моделируются последовательностями переходов. Следовательно, множество разрешаемых последовательностей переходов характеризует сеть Петри и (со степенью корректности моделирования системы сетью Петри) моделирующую систему.

Эти последовательности переходов могут быть крайне важны при использовании сетей Петри. Предположим, что для замены существующей системы спроектирована новая. Поведение новой системы должно быть идентично поведению старой системы (но новая система может оказаться более дешевой, более быстродействующей, более простой для исправлений или иметь другие улучшенные характеристики). Если обе системы моделируются сетями Петри, то поведение этих двух систем должно быть идентичным, и, следовательно, языки их равны. Две сети Петри называются *эквивалентными*, если равны их языки. Это образует формальную основу для установления эквивалентности двух систем.

Эквивалентность важна, в частности, при *оптимизации*. Оптимизация сети Петри подразумевает создание новой сети Петри, являющейся эквивалентной (языки равны), но которая лучше, чем старая (в смысле некоторого функционала качества). Например, если сеть Петри должна непосредственно реализоваться в аппаратуре, то построение сети Петри с меньшим числом позиций, переходов и дуг будет менее дорого, поскольку она имеет меньшее число компонент. Следовательно, одной из оптимизационных задач может быть сокращение $|P| + |T|$ без изменения поведения сети.

В целях оптимизации может оказаться полезным множество *преобразований, сохраняющих язык*. Если преобразование, примененное к сети Петри, порождает новую сеть Петри с тем же языком, то оно является сохраняющим язык. Оптимальную сеть Петри можно получить путем применения сохраняющих язык преобразований к неоптимальной сети Петри. Для практического использования

Кононюк А.Е. Графы

моделирования и анализа систем на основе сетей Петри требуется набор преобразований, сохраняющих язык.

Языки сетей Петри могут быть полезны также для анализа сетей Петри. Ранее были разработаны методы определения частных свойств сетей Петри: безопасности, ограниченности, сохранения, активности, достижимости и покрываемости. Хотя установление этих свойств важно (и трудно), они не единственные, которые анализируются в сети Петри. Возможно, окажется необходимым установить корректность моделируемой системы, показав, что ей присущи специфические свойства. Таким образом, либо для каждого нового свойства нужно разрабатывать новые методы, либо необходимы общие методы анализа сетей Петри.

Исходя из последовательностей действий, возможных в системе, можно поставить множество вопросов. Если определить множество возможных последовательностей действий как язык системы, то система будет анализироваться путем анализа языка. Теперь задачи решаются путем рассмотрения вопроса существования (переведет ли какая-нибудь последовательность действий систему из одного состояния в другое?) или вопроса принадлежности (возможна ли последовательность действий данного вида?). Благодаря этому можно получить общий метод анализа произвольных систем с целью проверки на обладание свойствами, специфическими для них. Риддл занимался исследованиями анализа на основе языка моделируемой системы.

Другое применение языков сетей Петри лежит в области задания и автоматического *синтеза* сетей Петри. Если задать языком требуемое поведение, то можно будет автоматически синтезировать сеть Петри, обладающую данным языком. Полученную сеть Петри можно использовать в качестве контроллера, гарантирующего, что возможны только указанные последовательности и никакие другие. Для определения разрешаемых последовательностей действий были предложены *Е-выражения*. На их основе разработаны методы автоматического построения сетей Петри.

Еще одна причина изучения языков сетей Петри — желание получить информацию о разрешимости ряда задач для сетей Петри. Например нам не известно разрешимы ли задачи об обладании сетями Петри определенными свойствами. Ведутся исследования разрешимости таких основных задач, как достижимость. В частности, одна из областей, в которых рассматриваются вопросы разрешимости, — это *теория формальных языков*. Используя языки сетей Петри, можно перенести понятия и методы теории формальных языков на задачи для сетей Петри. Возможно, это позволит получить некоторые результаты

в задачах разрешимости для сетей Петри. И наоборот, методы сетей Петри могут оказаться весьма полезными для получения новых сведений о формальных языках.

2.6.2. Некоторые понятия теории формальных языков

Развитая теория языков сетей Петри подобна другим частям теории формальных языков. Классическая теория формальных языков представлена в нескольких монографиях. Многие из основных понятий теории языков сетей Петри заимствованы из классической теории формальных языков.

Алфавит — это конечное множество символов. *Строка* — любая последовательность конечной длины из символов алфавита. *Пустая строка* λ — это строка, не имеющая символов, т. е. нулевой длины. Если Σ — алфавит, то Σ^* — множество всех строк из символов Σ , включая пустую строку. Σ^+ является множеством всех непустых строк над алфавитом Σ . $\Sigma^* = \Sigma^+ \cup \{\lambda\}$.

Языком называется множество строк над алфавитом. В общем случае языки могут быть бесконечными, следовательно, одной из проблем является представление языка. Для представления языков было предложено два подхода. Один заключается в определении машины, которая порождает строку из языка, и любая строка из языка порождается ею. Другой подход подразумевает определение грамматики, которое указывает, как последовательным применением ее правил порождения получают строки языка.

Ограничения, накладываемые на вид машин или грамматик, порождающих языки, определяют классы языков. Традиционными **классами языков** являются: **регулярный, контекстно-свободный, контекстно-связанный и языки типа 0, соответствующие конечным автоматам, магазинным автоматам, линейно-ограниченным автоматам и машинам Тьюринга.** Каждый из классов языков порождается соответствующим классом автоматов. Это дает прекрасные средства для установления связи теории сетей Петри с теорией формальных языков: мы определяем класс языков сетей Петри как класс языков, порождаемых сетями Петри. Детали этого определения аналогичны деталям определения любого другого класса языков.

Рассмотрим в качестве примера конечные автоматы и регулярные выражения. Конечный автомат C есть пятерка $(Q, \delta, \Sigma, s, F)$, где Q —

Кононюк А.Е. Графы

конечное множество состояний, Σ — алфавит символов, $\delta: Q \times \Sigma \rightarrow Q$ — функция переходов, $s \in Q$ — начальное состояние, $F \subseteq Q$ — конечное множество *заключительных состояний*. Функция переходов δ естественным образом обобщается на случай отображения из $Q \times \Sigma^*$ в Q . Язык $L(C)$, порождаемый конечным автоматом, — это множество строк над Σ , определяемое следующим образом:

$$L(C) = \{\alpha \in \Sigma^* \mid \delta(s, \alpha) \in F\}.$$

Всякому конечному автомату соответствует язык, а класс всех языков, порождаемых конечными автоматами, называется классом регулярных языков. Конечный автомат определяется своим языком. Если два конечных автомата имеют одинаковые языки, они *эквивалентны*.

2.6.3. Определения языков сетей Петри

Основные понятия, используемые для получения регулярного языка по конечному автомату, применимы и к сетям Петри для образования теории языков сетей Петри. В дополнение к сети Петри, определяемой множеством позиций и переходов (которые приблизительно соответствуют множеству состояний и функции переходов автомата), необходимо определить начальное состояние, алфавит и множество заключительных состояний. Задание этого для сети Петри может привести к различным классам языков сетей Петри. Рассмотрим их по порядку.

2.6.3.1. Начальное состояние

Начальное состояние сети Петри можно определить различными способами. Наиболее общепринятое определение — считать начальным состоянием произвольную маркировку μ . Однако это определение имеет несколько модификаций. Одним из ограничений в определении начального состояния является рассмотрение только маркировок с одной фишкой в *начальной позиции* и нулем фишек в остальных. Другое, более общее определение допускает множество начальных маркировок вместо одной маркировки.

Эти три определения, по существу, одинаковы. Определение начальной позиции является частным случаем определения начальной маркировки, а оно — частным случаем определения множества начальных маркировок. Однако если при определении начальной позиции будет необходимо множество начальных марки-. Ровок $M = \{\mu_1, \mu_2, \dots, \mu_k\}$, то можно просто ввести в сеть Петри

Кононюк А.Е. Графы

дополнительную позицию p_0 и множество переходов $\{t'_1, \dots, t'_k\}$. Переход t'_j будет брать фишку из p_0 и порождать маркировку μ_j . Таким образом, поведение расширенной сети будет идентично поведению сети Петри с множеством начальных маркировок, за исключением того, что каждой последовательности переходов будет предшествовать t_j , если до начала выполнения будет использоваться маркировка μ_j . Мы видим теперь, что эти три определения начальных состояний сети Петри в принципе эквивалентны. Отходя от традиции, определим язык сети Петри как начинающийся с отдельной маркировки μ .

2.6.3.2. Помечение сетей Петри

Как и в случае с начальным состоянием, возможно несколько определений помечения сети Петри. Мы должны определить и алфавит сети Петри, и то, как он связан с сетью Петри. Уже отмечалось, что символы алфавита связываются с переходами, поэтому последовательность запусков переходов порождает строку символов языка. Связь символов с переходами осуществляется функцией *помечения* $\sigma: T \rightarrow \Sigma$. Вариации в определении языка следуют из различных ограничений, накладываемых на функцию помечения.

Свободно помеченная сеть Петри — это помеченная сеть Петри, в которой все переходы помечены по-разному (т. е. если $\sigma(t_i) = \sigma(t_j)$, то $t_i = t_j$). Класс языков свободных сетей Петри является подмножеством класса языков сетей Петри с более общей функцией помечения, в которой не требуются различные метки. Рассматривалась еще более общая функция помечения, допускающая помечение переходов пустым символом, $\sigma(t_j) = \lambda$. λ -помеченные переходы не появляются в предложениях языка сети Петри, и, следовательно, их запуск при выполнении сети Петри не фиксируется. Три класса функций помечения (свободные, без λ - и с λ -переходами) определяют три класса языков сети Петри.

Без дальнейшего исследования нельзя сказать, какое из этих трех определений помечения наиболее приемлемо. Возможно, каждое из них является наиболее удачным для некоторого применения. Поэтому мы вынуждены рассмотреть языки, получающиеся при каждом возможном определении функции помечения.

2.6.3.3. Заключительные состояния сети Петри

Определение заключительных состояний сети Петри оказывает наибольшее влияние на язык сети Петри. Были предложены четыре

основных определений множества заключительных состояний сети Петри. Каждое из них образует свой язык сети Петри.

Одно из определений взято по аналогии с соответствующим понятием для автоматов — множество заключительных состояний F определяется как конечное множество заключительных маркировок. Этим определением мы вводим класс языков сетей Петри L -типа.

Определение 1. Язык L является языком сети Петри L -типа, если существует сеть Петри (P, T, I, O) , помечение переходов $\sigma: T \rightarrow \Sigma$, начальная маркировка μ и конечное множество заключительных маркировок F , такое, что $L = \{\sigma(\beta) \in \Sigma^* \mid \beta \in T^* \text{ и } \delta(\mu, \beta) \in F\}$.

$\delta(\mu, \beta) \in F$.

Класс языков сети Петри L -типа является мощным, но свойственное ему требование, что для порождения предложения необходимо прийти *точно* к заключительному состоянию, противоречит основам подхода сетей Петри. Это замечание основано на том, что если для маркировки μ и перехода t_j определено $\delta(\mu, t_j)$, то для любого $\mu' \geq \mu$ определено $\delta(\mu', t_j)$. Из этого вытекает определение нового класса языков — языков сети Петри G -типа.

Определение 2. Язык L является языком сети Петри G -типа, если существует сеть Петри (P, T, I, O) , помечение $\sigma: T \rightarrow \Sigma$, начальная маркировка μ , и конечное множество заключительных маркировок F такое, что $L = \{\sigma(\beta) \in \Sigma^* \mid \beta \in T^*, \text{ и существует } \mu_f \in F \text{ такое, что } \delta(\mu, \beta) \geq \mu_f\}$.

Третьим классом языков сетей Петри является класс языков сетей Петри T -типа. Эти языки определяются множеством заключительных состояний, используемым в определении языков L -типа, и множеством (не обязательно конечным) терминальных состояний. Состояние μ_i является терминальным, если $\delta(\mu_i, t_j)$ не определено для всех $t_j \in T$. Таким образом, класс языков сетей Петри T -типа задается следующим образом.

Определение 3. Язык L является языком сети Петри T -типа, если существует сеть Петри (P, T, I, O) , помечение $\sigma: T \rightarrow \Sigma$ и начальная маркировка μ , такая, что $L = \{\sigma(\beta) \in \Sigma^* \mid \beta \in T^* \text{ и } \delta(\mu, \beta) \text{ определено, но для любого } t_j \in T, \delta(\delta(\mu, \beta), t_j) \text{ не определено}\}$.

Четвертый класс образован языками сетей Петри P -типа, множества заключительных состояний которых включают все достижимые состояния. Эти языки являются префиксными, так как если $\alpha \in \Sigma^*$ является элементом языка P -типа, то для всех префиксов β элемента α , (т. е. $\alpha = \beta x$ для некоторого $x \in \Sigma^*$) β является элементом того же языка.

Кононюк А.Е. Графы

Определение 4. Язык L является языком сети Петри P -типа, если существует сеть Петри (P, T, I, O) , помечение $\sigma: T \rightarrow \Sigma$ и начальная маркировка μ , такая, что $L = \{\sigma(\beta) \in \Sigma^* \mid \beta \in T^* \text{ и определено } \delta(\mu, \beta)\}$.

2.6.3.4. Классы языков сетей Петри

В дополнение к четырем классам языков сети Петри, основанным на различных заданиях множества заключительных состояний, имеются вариации языков, порожденные различными определениями функции помечения. На рис. 1 приведены 12 классов языков, получающихся в результате комбинаций четырех типов определений заключительных состояний и трех типов функций помечения. Каждая клетка таблицы содержит обозначение соответствующего класса языков сетей Петри.

	<i>Свободные</i>	<i>без λ переходов</i>	<i>с λ-переходами</i>
<i>L-тип</i>	L^f	L	L^λ
<i>G-тип</i>	G^f	G	G^λ
<i>T-тип</i>	T^f	T	T^λ
<i>P-тип</i>	P^f	P	P^λ

Рис. 1. 12 классов языков сетей Петри.

Для задания конкретного языка сети Петри должны быть определены четыре элемента: сеть Петри $C = (P, T, I, O)$; функция помечения $\sigma: T \rightarrow \Sigma$; начальная маркировка $\mu: P \rightarrow N$; множество заключительных маркировок F (для языков L - и G -типа). Назовем $\gamma = (C, \sigma, \mu, F)$ *помеченной сетью Петри* для сети Петри C , помечения σ , начальной маркировки μ и множества заключительных состояний F . Для данной помеченной сети Петри γ можно определить 12 языков: $L(\gamma)$, $G(\gamma)$, $P(\gamma)$, $T(\gamma)$, $L^f(\gamma)$, $G^f(\gamma)$, $T^f(\gamma)$, $P^f(\gamma)$, $L^\lambda(\gamma)$, $T^\lambda(\gamma)$, $P^\lambda(\gamma)$.

Различные определения языков сетей Петри связывают с данной сетью Петри различные языки. Рассмотрим, например, сеть Петри на рис. 2.

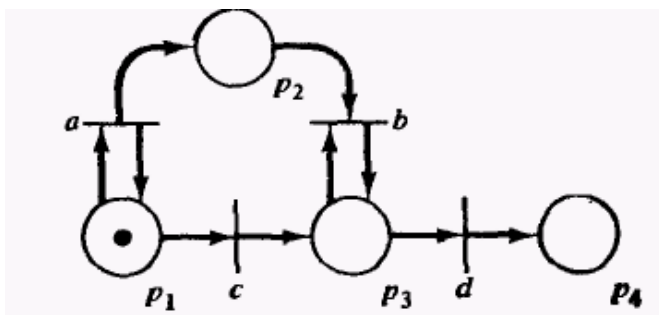


Рис. 2. Сеть Петри, иллюстрирующая различные классы языков. Каждый переход помечается меткой.

На сети задана начальная маркировка $(1, 0, 0, 0)$ и каждый переход t_j помечен $\sigma(t_j)$. Если мы определим $F = \{(0, 0, 1, 0)\}$ (одна фишка в позиции p_3), языком L-типа будет $\{a^n cb^n / n \geq 0\}$, языком G-типа — $\{a^m cb^n / m \geq n \geq 0\}$, языком T-типа — $\{a^m cb^n d / m \geq n \geq 0\}$, а языком P-типа — $\{a^m / m \geq 0\} \cup \{a^m cb^n / m \geq n \geq 0\} \cup \{a^m cb^n d / m \geq n \geq 0\}$.

Языки всех четырех типов для этого примера различны. Данная функция помечения является свободным помечением, но при использовании других функций помечения можно получить и другие языки.

Несмотря на различия в определениях, классы языков сетей Петри тесно связаны. Например, множество свободных помечений включено в множество не- λ -помечений, которое является подмножеством множества λ -помечений. Таким образом,

$$L' \subseteq L \subseteq L^\lambda, \quad T' \subseteq T \subseteq T^\lambda, \\ G' \subseteq G \subseteq G^\lambda, \quad P' \subseteq P \subseteq P^\lambda.$$

Кроме того, всякий язык P-типа является языком G-типа с $F = \{(0, 0, \dots, 0)\}$. Поэтому

$$P' \subseteq G', \quad P \subseteq G, \quad P^\lambda \subseteq G^\lambda.$$

Можно показать также, что всякий язык типа G или G^λ является языком типа L или L^λ соответственно. Пусть L — язык G-типа для сети Петри (P, T, I, O) , начальной маркировки μ и множества заключительных состояний F . Построим новую помеченную сеть Петри с теми же позициями, но с дополнительными переходами, определенными следующим образом:

Кононюк А.Е. Графы

Пусть для каждого $t_j \subseteq T$ B_j означает множество всех собственных подкомплектов $O(t_j)$. Всякий подкомплект из B_j используется для определения нового перехода, помеченного так же и имеющего те же входы, что и t_j , но имеющего в качестве выхода этот подкомплект. Эти новые переходы добавляются к первоначальному множеству переходов.

Например, если рассмотреть переход a в сети Петри на рис. 2, то его входным комплектом является $\{p_1\}$, а выходным комплектом $\{p_1, p_2\}$. Подкомплектами $\{p_1, p_2\}$ являются $\{p_1\}$, $\{p_2\}$ и $\{\}$ ($= \emptyset$). Этот переход влечет введение в сеть трех новых переходов. Все эти новые переходы будут помечены a и иметь входной комплект $\{p_1\}$, но выходными комплектами будут три указанных выше подкомплекта (по переходу на каждый подкомплект). Для переходов b, c, d также будут введены новые переходы, которые будут иметь те же входы, но пустые выходы (поскольку в имеющейся сети выходы образованы одноэлементными комплектами, следовательно, единственным подкомплексом будет \emptyset). Новая сеть Петри показана на рис. 3.

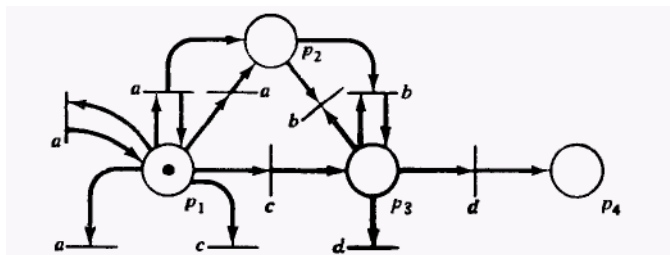


Рис. 3. Сеть Петри, язык L-типа которой совпадает с языком S-типа сети Петри, приведенной на рис. 2.

Эта новая сеть модифицируется таким образом, что лишние фишки, превышающие заключительное состояние из F , не обязательно будут воспроизведены, если выбирается новый переход, имеющий меньше выходов. Таким образом, язык L-типа новой сети совпадает с языком G-типа старой сети.

В описанной конструкции требуется создание новых переходов с теми же метками, что и прежние, поэтому никакого вывода относительно соотношения G^f и L^f из этой конструкции сделать нельзя.

$$G \subseteq L, \quad G^\lambda \subseteq L^\lambda.$$

Кононюк А.Е. Графы

Рассмотренную конструкцию можно также незначительно модифицировать для того, чтобы показать, что обобщение определения языка L-типа (допускающее неполное задание заключительных маркировок) не изменяет классов L и L^λ . Пусть заключительная маркировка сети Петри с n позициями является n -вектором над

$N \cup \{\omega\}$. Если ω является компонентой заключительной маркировки, то это означает, что мы не должны заботиться о вхождении значения этой компоненты в заключительное состояние. Состояние s является заключительным, если существует такая заключительная маркировка f , что для всех i , $1 \leq i \leq n$, из $f_i \neq \omega$ следует $s_i = f_i$. Это определение более общее, чем данное ранее определение языков L-типа.

Рассмотрим сейчас язык, определяемый сетью Петри и не полностью заданной заключительной маркировкой f . Пусть τ —множество всех позиций, для которых $f_i = \omega$. Пусть для каждого $t_j \in T$, удовлетворяющего $O(t_j) \cap \tau \neq \emptyset$, $\rho_j = O(t_j) \cap \tau$ и

$\psi_j = O(t_j) - \rho_j$. Комплект ρ_j включает все позиции, маркировка которых не принимается во внимание, тогда как комплект ψ_j , включает все позиции, маркировка которых должна быть задана точно так же, как в заключительной маркировке f . Введем в сеть новые переходы, их метки и входные комплекты совпадают с метками и входными комплектами t_j , а выходные комплекты образованы $\psi_j + \xi$, где ξ — любой подкомплект ρ_j . Эта конструкция действительно не меняет поведения позиций, не входящих в τ , но позволяет позициям, непринимаемым во внимание, иметь произвольное число фишек, меньшее или равное числу фишек, которые должны быть в первоначальной сети. Таким образом, для каждого предложения обобщенного языка первоначальной сети можно выбрать соответствующие переходы для запуска их в то время, когда сеть достигает такого состояния s , что $s_i = f_i$ для $f_i \neq \omega$, $s_i = 0$ для $f_i = \omega$. Следовательно, язык L-типа для построенной сети Петри с заключительной маркировкой f' (где все ω в не полностью заданной маркировке f заменяются на 0) совпадает с обобщенным языком первоначальной сети (т. е. определенным не полностью заданной заключительной маркировкой f).

В случае языков, определяемых множеством не полностью заданных заключительных маркировок (в отличие от единственной такой маркировки, как в только что рассмотренном случае) на основе того, что языки типа L (и L^λ) являются замкнутыми по отношению к операции объединения (см. разд. 2.6.5.2), можно показать, что данные языки остаются еще языками типа L (или L^λ).

Кононюк А.Е. Графы

Вводя не полностью заданные заключительные маркировки, можно показать, что языки T -типа являются также языками L -типа (за исключением, быть может, свободных языков T -типа).

Заключительным состоянием μ для языка T -типа является такое состояние, что никакой из переходов t_j нельзя запустить (т. е. $\mu \geq I(t_j)$) для всех $t_j \in T$). Таким образом, условие, задающее заключительное состояние для языка T -типа, в точности противоположно условию, задающему заключительное состояние для языка G -типа. (Можно назвать язык T -типа обращением языка G -типа.) Нетрудно видеть, что такое множество маркировок может быть описано конечным множеством не полностью заданных маркировок (как это сделано в разд. 2.5.4). Например, условие $\mu \geq (2, 0)$ и $\mu \geq (1, 1)$ эквивалентно ($\mu = (0, \omega)$ или $\mu = (1, 0)$). Язык T -типа (или в более общем случае обращение языка G -типа) можно поэтому представить обобщенным (т. е. не полностью заданным) языком L -типа и, следовательно, языком L -типа. Таким образом, $T \subseteq L$ и $T^\lambda \subseteq L^\lambda$.

Как известно, всякий язык L^λ -типа можно построить по сети Петри, в которой каждый переход имеет входную позицию, и по единственной заключительной маркировке, являющейся нулевой, т. е. такой, при которой нельзя запустить никакой переход. Если к каждой позиции добавить λ -переход с одним входом и одним выходом, совпадающим с входом (т. е. петлю), то язык не изменится, а нулевая маркировка станет единственной терминальной маркировкой. Следовательно, $L^\lambda \subseteq T^\lambda$, но из предыдущих рассуждений $T^\lambda \subseteq L^\lambda$ поэтому классы языков эквивалентны $T^\lambda = L^\lambda$.

На рис. 4 графически представлены соотношения рассмотренных классов языков сетей Петри. Дуга между классами означает включение одним классом языков сетей Петри другого.

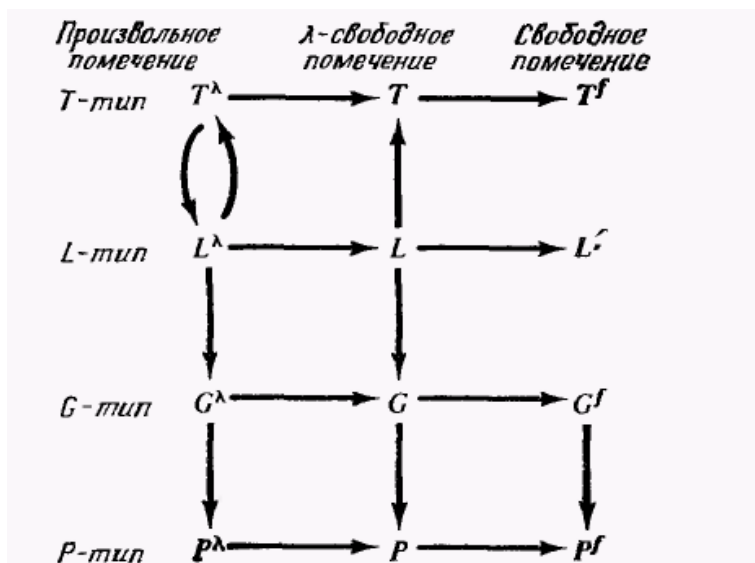


Рис. 4. Известные соотношения между классами языков сетей Петри. Дуга из класса А в класс В означает, что класс А включает класс В.

2.6.4. Свойства языков сетей Петри

Сила сетей Петри отражается в разнообразии классов языков сетей Петри. Новизна исследований объясняет нашу неспособность показать все взаимосвязи между этими языками или обосновать важность только некоторых из классов. Это приводит к широте области исследований, необходимости определения свойств 12 различных классов языков.

Очевидно, что все 12 классов языков исследовать в этой книге невозможно, поэтому мы ограничимся рассмотрением здесь только одного класса языков сетей Петри, а именно языков L-типа. Основными причинами такого решения являются ограниченность объема монографии и то, что этот язык уже исследовался в литературе. Некоторые результаты были получены Хэком и для префиксных языков (т. е. P-типа) и будут представлены в нашем заключении. Напомним, что класс языков L-типа включает классы языков T-, G- и P-типа. Таким образом, языки L-типа в некотором смысле образуют наибольший класс языков сетей Петри, и поэтому понятно, почему они исследовались первыми.

Кононюк А.Е. Графы

При изучении свойств языков сетей Петри L-типа обратимся к двум аспектам. Сначала представим свойства *замкнутости* сетей Петри по отношению к некоторым обычным операциям: конкатенации, объединению, параллельной композиции, пересечению, обращению, дополнению, бесконечной конкатенации и подстановке. Затем рассмотрим взаимосвязь между языками сетей Петри и классическими формальными языками: регулярными, контекстно-свободными, контекстно-связанными и типа 0. Это позволит оценить силу и ограниченность языков сетей Петри (L-типа), а также покажет, как можно исследовать другие классы языков сетей Петри. Хотя нас интересует весь класс языков сетей Петри L-типа, ограничимся рассмотрением только множества сетей Петри *стандартного вида*. Это ограничение вводится для того, чтобы облегчить доказательства и конструкции; оно не ограничивает класс языков сетей Петри. Всякий язык сетей Петри может порождаться множеством сетей; выберем для работы только сети, обладающие определенными свойствами. Для доказательства того, что это не сокращает множество языков, покажем, что для всякого языка сети Петри L-типа существует сеть Петри стандартного вида, порождающая его.

Определение 5. Помеченная сеть Петри $\gamma = (C, \sigma, \mu, F)$ с языком $L(\gamma)$, имеющая стандартный вид, удовлетворяет следующим условиям:

1. Начальная маркировка μ состоит точно из одной фишки в начальной позиции p_s и не имеет фишек в других позициях. $p_s \notin O(t_j)$ для всех $t_j \in T$.

2. Существует позиция $p_f \in P$, такая, что

а) $F = \{p_f\}$ (если $\lambda \notin L(\gamma)$) и $F = \{p_f, p_s\}$ (если $\lambda \in L(\gamma)$) (F состоит из маркировки, содержащей фишку в p_f (если $\lambda \notin L(\gamma)$), а также маркировки, содержащей фишку в p_s $\lambda \notin L(\gamma)$);

б) $p_f \notin I(t_j)$ для всех $t_j \in T$;

в) $\delta(\mu', t_j)$ не определено для всех $t_j \in T$ и $\mu' \in R(C, \mu)$, имеющих в p_f фишку (т. е. $\mu'(p_f) > 0$).

Выполнение сети Петри стандартного вида начинается с одной фишки в начальной позиции. Первый запускаемый переход удаляет эту фишку из начальной позиции, после чего начальная позиция остается уже до конца выполнения пустой. Сеть Петри останавливается, если фишка помещается в заключительную позицию.

Эту фишку нельзя удалить из заключительной позиции потому, что, во-первых, никакой переход не имеет заключительной позиции в качестве входной, а, во-вторых, все переходы не разрешены. Таким образом, выполнение сети Петри стандартного вида достаточено просто. Это очень полезно при построении композиций из сетей Петри. Для

того чтобы показать, что сети стандартного вида не менее мощны, чем обычные сети Петри, докажем следующую теорему.

Теорема 1. Всякая сеть Петри эквивалентна сети Петри стандартного вида.

Доказательство. Доказательство проведем с помощью вспомогательного построения. Пусть $\gamma = (C, \sigma, \mu, F)$ — помеченная сеть Петри с $C = (P, T, I, O)$. Покажем, как построить эквивалентную сеть $\gamma' = (C', \sigma', \mu', F')$ с $C' = (P', T', I', O')$ стандартного вида (рис. 5).

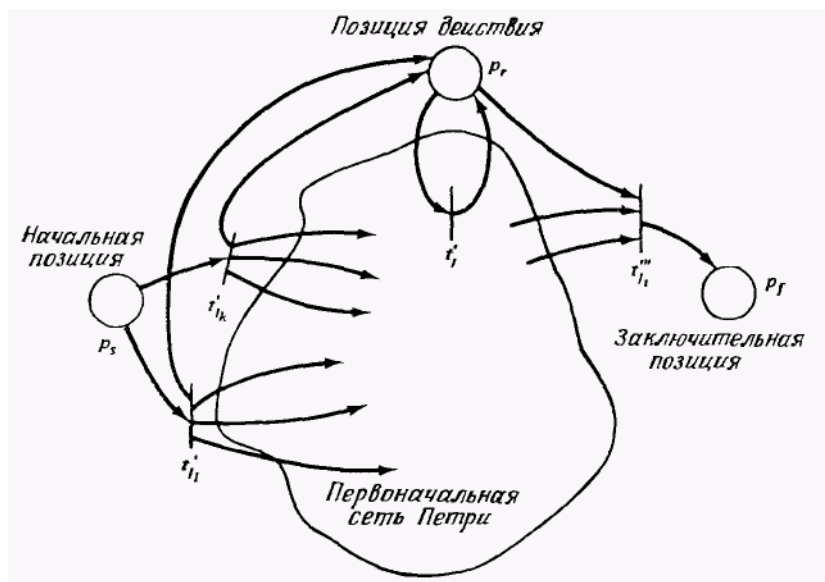


Рис. 5. Приведение сети Петри к стандартному виду. Выполнение этой сети совпадает с выполнением первоначальной сети.

Прежде всего введем три новые позиции p_r , p_f и p_s , которых не было в P . Позиция p_s является начальной, p_f — заключительной, а p_r — позиция «действия»; для того чтобы запуск любого перехода в T был разрешен, в позиции p_r должна быть фишка. В начальной маркировке C будет только одна фишка в p_s ; заключительная маркировка будет содержать одну фишку в p_s (если $\lambda \in L(\gamma)$) или в p_f (если $\lambda \notin L(\gamma)$).

Теперь необходимо гарантировать, что всякая последовательность переходов в C , приводящая из начальной маркировки в заключительную, повторима в C' . Рассмотрим для этого три типа строк в

Кононюк А.Е. Графы

$L(\gamma)$. Во-первых, пустая строка учитывается определением F' . Принадлежность λ языку $L(\gamma)$ определяется проверкой того, является ли начальная маркировка μ заключительной, $\mu \in F$. Во-вторых, для всех строк длины 1 в $L(\gamma)$ введем в C' специальный переход из p_s в p_f следующим образом: для $\alpha \in \Sigma$, такого, что $\alpha \in L(\gamma)$, определим $t_\alpha \in T'$ с $I(t_\alpha) = \{p_s\}$, $O(t_\alpha) = \{p_f\}$. Меткой t_α будет α . Установить принадлежность a языку $L(\gamma)$ можно, рассматривая все переходы $t_f \in T$ с $\sigma(t_f) = a$ и проверяя, не принадлежит ли $\delta(\mu, t_f) \in F$. Наконец, рассмотрим все строки длины больше 1. Эти строки получаются в результате запуска последовательности $t_{j_1} t_{j_2} \dots t_{j_k}$ переходов T . Нам бы хотелось определить последовательность $t'_{j_1} t'_{j_2} \dots t'_{j_k} b$ с новыми переходами a и b . Переход a будет брать

фишку из p_s и порождать начальную маркировку μ сети C плюс фишку в p_r . Каждый переход t'_j в T' идентичен переходу t_j в T , за исключением того, что p_r является для него входной и выходной. Это позволит нам запретить все переходы в T' путем удаления фишки из p_r . Переход b будет удалять фишку из p_r , уничтожать заключительную маркировку C и помещать фишку в p_f . При таком построении фишка будет перемещаться из начальной позиции в заключительную только в результате последовательности $t'_{j_1} t'_{j_2} \dots t'_{j_k} b$, соответствующей последовательности $t_{j_1} t_{j_2} \dots t_{j_k}$, которая переводит μ в заключительную маркировку в C .

К сожалению, поскольку в C' , но не в C будут присутствовать лишние символы, соответствующие переходам a и b , полученная последовательность окажется длиннее. Выходом из такого положения явилось бы пустое помечение a и b , но в языках L-типа оно неразрешено. Для обхода этой трудности мы вынуждены «слить» переходы a и t'_{j_1} в один переход t''_{j_1} , а переходы b и t'_{j_k} — в переход t''_{j_k} . Переход в T' для всякого $t_j \in T$ определяется следующим образом:

1. Вводим в T' t_j с $I(t_j) = I(t_j) \cup \{p_r\}$ и $O(t_j) = O(t_j) \cup \{p_r\}$
2. Если $I(t_j) \subseteq \mu$ (т. е. входы в t_j образуют подмножество (точнее, подкомплект) начальной маркировки, вследствие чего t_j может быть первым запускаемым в C переходом), вводим в T' переход t'' с $I(t'') = \{p_s\}$ и $O(t'_j) = \mu - I(t_j) \cup \{p_r\}$.
3. Если $O(t_j) \subseteq \mu'$ для некоторой $\mu' \in F$ (т. е. t_j может быть последним переходом, запуск которого приводит к заключительной маркировке), вводим переход t_j''' с $I(t_j''') = \mu' - O(t_j) \cup I(t_j) \cup$

$$\cup \{p_r\} \text{ и } O(t_j''') = \{p_f\}.$$

Помечениа' определяется следующим образом:

$$\sigma' (t'_j) = \sigma' (t''_j) = \sigma' (t'''_j) = \sigma (t_j).$$

Любая строка $\alpha \in L(\gamma)$ порождается по определению последовательностью $t_{i_1}, t_{i_2}, \dots, t_{i_l}$, такой, что $\sigma(t_{i_1}, t_{i_2}, \dots, t_{i_l}) = \alpha$. По построению

$$\alpha = \sigma' (t''_{i_1} t'_{i_2} \dots t'_{i_{l-1}} t'''_{i_l}),$$

поэтому $\alpha \in L(\gamma')$. Следовательно, $L(\gamma) = L(\gamma')$ и сети γ и γ' эквивалентны. Сеть γ' имеет по построению стандартный вид.

На рис. 6 приведена простая сеть Петри, не имеющая стандартного вида.

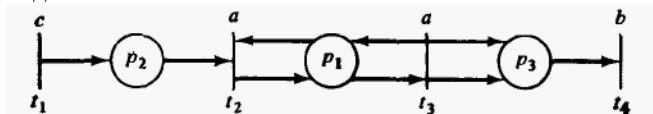


Рис. 6. Сеть Петри нестандартного вида. Позиция p_1 является и начальной, и заключительной.

Применение к этой сети Петри конструкции из доказательства приводит к сети Петри, показанной на рис. 7.

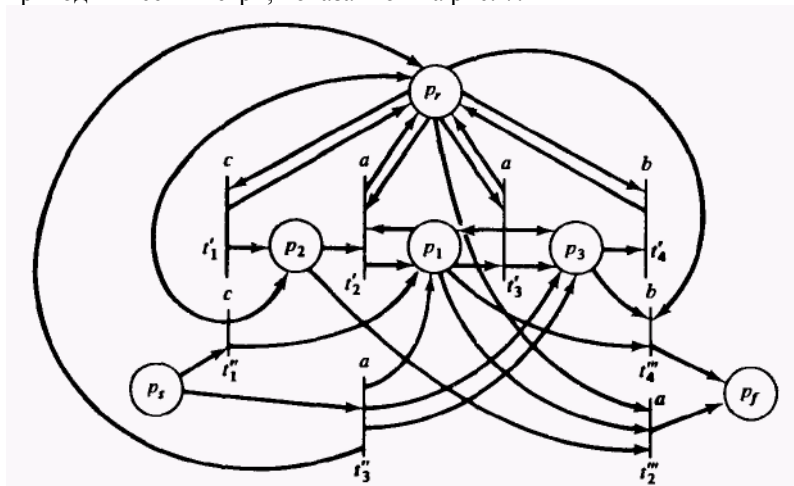


Рис. 7. Сеть Петри стандартного вида, эквивалентная сети Петри, приведенной на рис.6.

2.6.5. Свойства замкнутости

Перейдем к изучению свойств замкнутости языков сетей Петри по отношению к некоторым видам композиции (объединению, пересечению, конкатенации, параллельной композиции, и подстановке) и по отношению к некоторым операциям (обращению, дополнению и бесконечной конкатенации). Изучение этих свойств интересно по двум причинам. Во-первых, оно увеличивает наши знания о свойствах и ограничениях языков сетей Петри как языков. Во-вторых, многие из перечисленных типов композиции отражают процесс композиционного проектирования и конструирования больших систем из малых, поэтому данное исследование может оказаться полезным для разработки методов синтеза. Большинство из свойств замкнутости касаются композиций языков сетей Петри. Рассмотрим два языка сетей Петри, L_1 и L_2 . Мы знаем, что каждый из них порождается некоторой сетью Петри стандартного вида. Поэтому можно рассматривать две помеченные сети Петри стандартного вида, $\gamma_1 = (C_1, \sigma_1, \mu_1, F_1)$ и $\gamma_2 = (C_2, \sigma_2, \mu_2, F_2)$, с $L_1 = L(\gamma_1)$ и $L_2 = L(\gamma_2)$. Поскольку сети стандартного вида, γ_1 имеет начальную позицию $p_{s_1} \in P_1$, а γ_2 — $p_{s_2} \in P_2$. Кроме того, $F_1 = \{ p_{f_1}, p_{f_1} \}$ или $\{ p_{f_1} \}$, а $F_2 = \{ p_{f_2}, p_{f_2} \}$ или $\{ p_{f_2} \}$.

Покажем, как из этих двух помеченных сетей строится новая помеченная сеть Петри $\gamma' = (C', \sigma', \mu', F')$ с языком $L(\gamma')$, который является композицией L_1 и L_2 требуемого типа. Описываемые конструкции будем иллюстрировать примерами сетей Петри.

Начнем рассмотрение свойств композиции языков сетей Петри с конкатенации, объединения, пересечения и параллельной композиции. Затем исследуем обращение и дополнение языка сети Петри и, наконец, подстановку.

2.6.5.1. Конкатенация

Многие системы представляют собой последовательную композицию двух подсистем. Каждую из подсистем можно представить сетью Петри со своим языком. При последовательной комбинации двух систем соответствующее выполнение является *конкатенацией*

Кононюк А.Е. Графы

выполнения из первого языка сети Петри и выполнения из второго. Конкатенация двух языков формально определяется как

$$L_1 L_2 = \{x_1 x_2 \mid x_1 \in L_1, x_2 \in L_2\}.$$

Теорема 2. Если L_1 и L_2 — языки сетей Петри, то $L_1 L_2$ — язык сети Петри.

Доказательство. Определим γ' таким образом, чтобы заключительная позиция γ_1 совместилась с начальной позицией γ_2 . Тогда переход, помещающий фишку в p_{f_1} , будет свидетельствовать о конце выполнения γ_1 а также о начале выполнения γ_2 . Любая строка конкатенации L_1 и L_2 является тогда путем из p_{s_1} в $p_{f_1} = p_{s_2}$ и затем в p_{f_2} в γ' и, следовательно, элементом $L(\gamma')$. Аналогично, если строка порождена сетью C' , она является конкатенацией строки, порожденной γ_1 и строки, порожденной γ_2 .

Формальное определение γ' учитывает пустую строку и поэтому более сложно, γ' можно определить как объединение γ_1 и γ_2 со следующими дополнительными переходами. Для всякого $t_j \in T_2$ с $I_2(t_j) = \{p_{s_2}\}$ вводим новый переход t'_j с $I'(t'_j) = \{p_{f_2}\}$ и $O'(t'_j) = O_2(t_j)$. Если $p_{s_1} \in F_1$, то вводим также t''_j с $I'(t''_j) = \{p_{s_1}\}$ и $O'(t''_j) = O_2(t_j)$.

Помечение определяем, как $\sigma'(t'_j) = \sigma_2(t_j)$ и $\sigma'(t''_j) = \sigma_2(t_j)$.

Новое заключительное множество $F' = F_1 \cup F_2$, если $p_{s_2} \in F_2$, в противном случае $F' = F_2$.

Из этой теоремы следует, что класс языков сетей Петри замкнут по отношению к конкатенации. На рис. 8 показывается описанная конструкция для $L_1 = (a+b)^+$ и $L_2 = a^n b^n$.

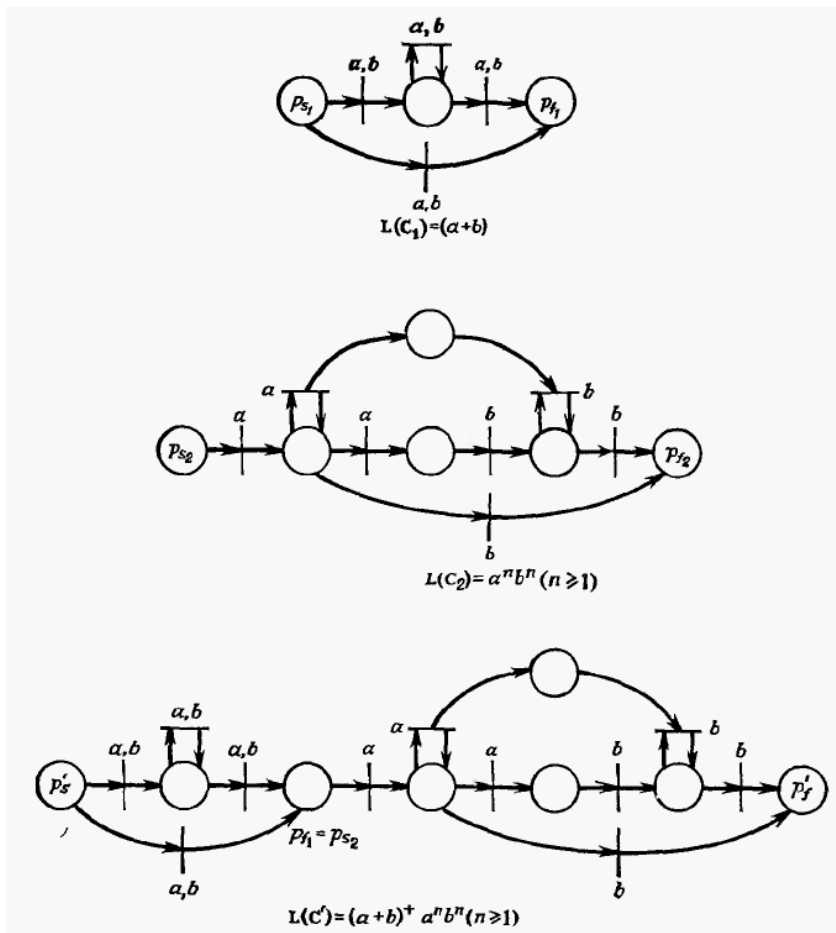


Рис. 8. Конкатенация двух языков сетей Петри. (Каждый переход в C_1 фактически является парой параллельных конфликтующих переходов, помеченных a и b соответственно.)

2.6.5.2. Объединение

Другой общепринятый метод композиции систем — *объединение*. В композиции такого вида будет выполняться любая, но только одна из

Кононюк А.Е. Графы

подсистем. Эта обычная композиция языков аналогична объединению множеств и определяется следующим образом:

$$L_1 \cup L_2 = \{x \mid x \in L_1 \text{ или } x \in L_2\}.$$

Теорема 3. Если L_1 и L_2 — языки сетей Петри, то $L_1 \cup L_2$ — язык сети Петри.

Доказательство. Доказательство аналогично доказательству предыдущей теоремы.

Определим новую сеть Петри γ' , такую, что $L(\gamma') = L_1 \cup L_2$. В новой сети Петри две начальные позиции объединены в одну новую начальную позицию. Первый запускаемый переход удаляет фишку из начальной позиции, после чего либо сеть Петри γ_1 (если переход был из T_1), либо сеть Петри γ_2 (если переход был из T_2) будет действовать в точности так же, как до объединения.

Формально мы определяем новую начальную позицию p'_s и новый переход t'_{j_1} , для каждого $t_{j_1} \in T_1$ с $I(t_{j_1}) = \{p_{s_1}\}$ и t'_{j_2} для каждого $t_{j_2} \in T_2$ с $I(t_{j_2}) = \{p_{s_2}\}$: $I'(t_{j_1}) = \{p_s\}$, $I'(t_{j_2}) = \{p_s\}$ и $O'(t_{j_1}) = O_1(t_{j_1})$, $O'(t_{j_2}) = O_2(t_{j_2})$. Функция помечения: $\sigma'(t'_{j_1}) = \sigma_1(t_{j_1})$ и $\sigma'(t'_{j_2}) = \sigma_2(t_{j_2})$. Новая начальная маркировка содержит одну фишку в p'_s , ноное множество заключительных маркировок $F' = F_1 \cup F_2$. Если $p_{s_1} \in F_1$ или $p_{s_2} \in F_2$, то, кроме того, $p'_s \in F'$.

На рис. 9 показана описанная конструкция для $L_1 = a(a+b)b$ и $L_2 = a^m b^n (m > n \geq 1)$.

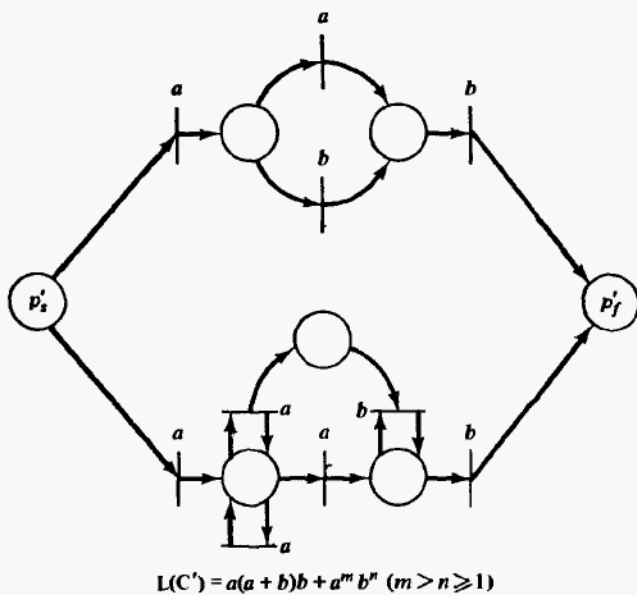
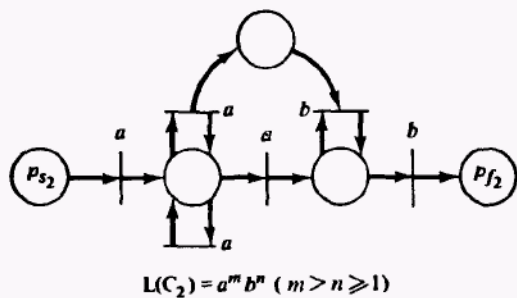
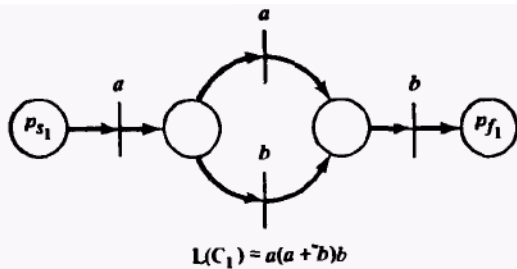


Рис. 9. Объединение двух языков сетей Петри.

2.6.5.3. Параллельная композиция

Еще один способ композиции двух сетей Петри заключается в разрешении *одновременного* выполнения двух систем. Он допускает все возможные «перемешивания» выполнений одной сети Петри с выполнениями другой. Для представления такой параллельной композиции Риддл ввел оператор \parallel , который определяется для $a, b \in \Sigma$, и $x_1, x_2 \in \Sigma^*$ следующим образом:

$$ax_1 \parallel bx_2 = a(x_1 \parallel bx_2) + b(ax_1 \parallel x_2),$$

$$a \parallel \lambda = \lambda \parallel a = a.$$

Параллельной композицией двух языков является

$$L_1 \parallel L_2 = \{x_1 \parallel x_2 \mid x_1 \in L_1, x_2 \in L_2\}.$$

В качестве простого примера рассмотрим языки $L_1 = \{ab\}$ и $L_2 = \{c\}$, язык $L_1 \parallel L_2$ — это $\{abc, acb, cab\}$.

Легко показать, что регулярные, контекстно-связанные языки и языки типа 0 замкнуты по отношению к параллельной композиции, поскольку параллельная композиция двух конечных автоматов, линейно-ограниченных автоматов или машин Тьюринга являются конечным автоматом, линейно-ограниченным автоматом и машиной Тьюринга соответственно. Вследствие того что параллельную композицию двух автоматов с магазинной памятью нельзя преобразовать в другой автомат с магазинной памятью, контекстно-свободные языки не замкнуты по отношению к параллельной композиции. Для языков сетей Петри докажем следующую теорему.

Теорема 4. Если L_1 и L_2 — языки сетей Петри, то $L_1 \parallel L_2$ — язык сети Петри.

Доказательство. Сеть Петри, порождающая параллельную композицию L_1 и L_2 , строится из сетей Петри, порождающих эти языки; в ней в качестве начальной маркировки фишки помещаются в начальные позиции γ_1 и γ_2 , новое множество заключительных маркировок определяется всеми маркировками, принадлежащими F_1 (в позициях P_1) и F_2 (в позициях P_2).

Эта конструкция показана на рис. 10 для $L_1 = a(a + b)^+$ и $L_2 = ca^{3n}cb^{2n}c$.

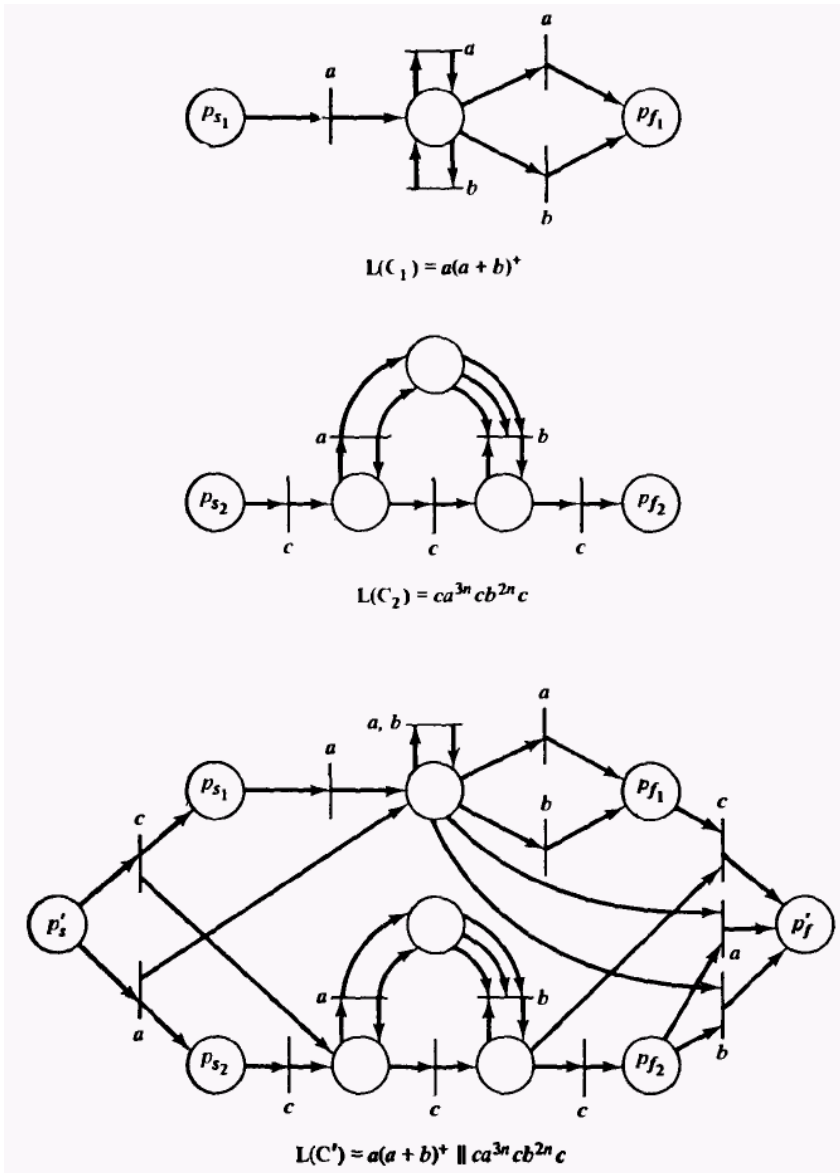


Рис. 10. Параллельная композиция двух языков сетей Петри.

2.6.5.4. Пересечение

Как и в случае объединения, операция пересечения подобна теоретико-множественному определению пересечения и определяется для языков сетей Петри следующим образом:

$$L_1 \cap L_2 = \{x \mid x \in L_1 \text{ и } x \in L_2\}.$$

Теорема 5. Если L_1 и L_2 — языки сетей Петри, то $L_1 \cap L_2$ — язык сети Петри.

Конструкция сети Петри, порождающей пересечение двух языков сетей Петри, несколько сложнее рассмотренных ранее. Всякий раз при порождении строки, когда запускается переход в одной сети Петри, должен запуститься также переход в другой сети Петри с идентичной меткой. Если в каждой сети Петри существует более одного перехода с одной меткой, необходимо рассмотреть все возможные пары переходов двух сетей Петри. Для каждой такой пары мы вводим новый переход, который запускается тогда и только тогда, когда запускаются оба перехода в старых сетях Петри. Это достигается взятием в качестве входного (выходного) комплекта нового перехода суммы комплектов (см. приложение) входных (выходных) комплектов пар переходов старых сетей Петри. Следовательно, если $t_j \in T_1$ и $t_k \in T_2$, такие, что $\sigma(t_j) = \sigma(t_k)$, то вводится переход $t_{j,k} \in T'$ с $I'(t_{j,k}) = T_1(t_j) + I_2(t_k)$ и $O'(t_{j,k}) = O_1(t_j) + O_2(t_k)$. Некоторые из таких переходов будут иметь в качестве входов начальную позицию. Если для перехода $t_{j,k} \in T'$, определенного описанным способом,

$I'(t_{j,k}) = \{p_{s_1}, p_{s_2}\}$, то заменяем этот переход на другой $t'_{j,k}$ с

$I'(t'_{j,k}) = \{p'_s\}$. Эта конструкция, по существу, включает две

первоначальные сети Петри в синхронный режим выполнения. Такая составная сеть порождает пересечение L_1 и L_2 . Описанная конструкция иллюстрируется на рис. 11.

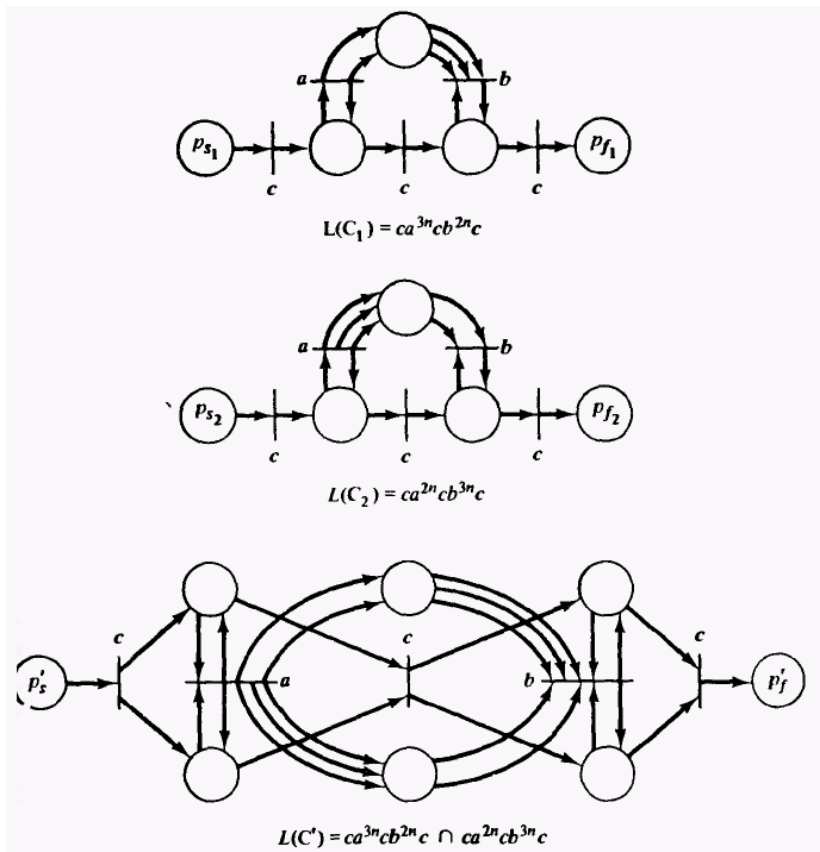


Рис. 11. Пересечение двух языков сетей Петри.

2.6.5.5. Обращение

В отличие от рассмотренных до сих пор операций обращение, по-видимому, представляет только теоретический интерес. *Обращение* предложения x^R — это предложение x , символы которого расположены в противоположном порядке. Мы определяем эту операцию рекурсивно:

$$a^R = a, \quad (ax)^R = x^R a$$

для $a \in \Sigma$, $x \in \Sigma^*$. Обращение языка определяется следующим образом:

$$L^R = \{x^R \mid x \in L\}.$$

Кононюк А.Е. Графы

Теорема 6. Если L — язык сети Петри, то и L^R — язык сети Петри. Конструкция в этом случае тривиальна. Начальная и заключительная маркировки меняются местом, меняются местом также входные и выходные комплекты каждого перехода. Следовательно, $L(\gamma') = L(\gamma)^R$. Построенная сеть просто выполняет первоначальную сеть Петри в обратном порядке и обращает все порождаемые ею строки.

2.6.5.6. Дополнение

Дополнение \bar{L} языка L над алфавитом Σ — это множество всех строк Σ^* , отсутствующих в L . Его можно представить как

$$\bar{L} = \Sigma^* - L \text{ или } \bar{L} = \{x \in \Sigma^* \mid x \notin L\}.$$

Эта операция над языком сети Петри может быть полезной при анализе сетей Петри, так как проверка существования запрещенных состояний или запрещенных последовательностей в дополнении может оказаться легче, чем проверка их несуществования в языке сети Петри.

Замкнутость языков сетей Петри L -типа по отношению к дополнению — задача, которая ждет своего решения. Показано, что некоторые языки сетей Петри не замкнуты по отношению к дополнению, т. е. существуют языки сетей Петри, дополнения которых не являются языками сетей Петри.

2.6.5.7. Повторная композиция

Мы рассмотрели операции объединения, пересечения, конкатенации, параллельной композиции, обращения и дополнения. Для всех операций, кроме последней, мы описали конструкции, показывающие, что языки сетей Петри замкнуты по отношению к ним. Результаты позволяют сделать следующий вывод.

Следствие 1. Языки сетей Петри замкнуты по отношению к любому конечному числу выполнений операций объединения, пересечения, обращения, параллельной композиции и конкатенации, осуществляемых в любом порядке.

Доказательство следует из приведенных выше теорем.

Устраняя ограничение на конечность числа разрешенных операций, можно определить новую операцию. *Бесконечная конкатенация* (звезда Клини) языка — это множество всех конкатенаций (произвольной длины) элементов языка. Она обозначается через L^* и определяется как

$$L^* = L \cup LL \cup LLL \cup \dots$$

Эта операция может оказаться полезной на практике. Бесконечная конкатенация подобна моделированию цикла. Кроме того, известно, что регулярные, контекстно-свободные, контекстно-связанные языки и языки типа 0 замкнуты по отношению к бесконечной конкатенации. К сожалению оказалось, что языки сетей Петри не замкнуты по отношению к бесконечной конкатенации.

Это объясняется конечностью сетей Петри (конечное число позиций и переходов) и разрешающей природой изменения состояний (переходы запускать разрешается, но потребовать их запуска нельзя). Для построения сети Петри, порождающей бесконечную конкатенацию языка сети Петри, потребовалось бы в общем случае повторно использовать некоторую часть сети Петри, что позволяет фишкам появляться и оставаться в некоторых из повторно используемых позиций сети Петри. При последующем повторе эти фишки могут разрешить переходы, которые не должны запускаться.

Доказательство того, что сети Петри не замкнуты по отношению к бесконечной конкатенации, оказалось очень сложным. Идею, положенную в основу доказательства, можно продемонстрировать, рассмотрев пример. Ранее показано, что $a^n b^n (n > 1)$ — язык сети Петри. Мы утверждаем, что $(a^n b^n)^*$ не является языком сети Петри. Любая сеть Петри, порождающая $(a^n b^n)^*$, должна иметь некоторую позицию или множество позиций, кодирующих число n для каждой части строки. Эти фишки управляют порождением символов b . Сеть Петри, порождающая $(a^n b^n)^*$, должна использовать эти фишки более одного раза. Но, поскольку природа сети разрешающая, нельзя гарантировать, что эти позиции станут пустыми перед повторным использованием. Следовательно, для любой сети Петри, пытающейся породить $(a^n b^n)^*$, существуют такие i, j, k , что сеть Петри порождает также некоторую строку следующего вида:

$$a^{n_i} b^{n_i} \dots a^{n_j} b^{n_j} \dots a^{n_k} b^{n_k} \dots a^{n_k'} b^{n_k'}$$

где $n_i > n_j$. Основу формального доказательства того, что языки сетей Петри не замкнуты по отношению к бесконечной конкатенации, дал Косараю.

Читатель, знакомый с лингвистической теорией абстрактных семейств языков, легко докажет, что языки сетей Петри не замкнуты по отношению к бесконечной конкатенации. Хорошо известно, что наименьшие полные абстрактные семейства языков замкнуты по отношению к пересечению, а содержащие $\{a^n b^n\}$ содержат и всякое рекурсивно перечислимое множество. Таким образом, поскольку языки сетей Петри замкнуты по отношению к пересечению и $\{a^n b^n\}$ — язык

Кононюк А.Е. Графы

сети Петри, то, если бы они были замкнуты по отношению к бесконечной конкатенации, они образовали бы абстрактное семейство языков. Однако нам известно, что $\{ww^R\}$ не является языком сети Петри (см. разд. 2.6.6.2), поэтому языки сетей Петри не замкнуты по отношению к бесконечной конкатенации. Эта аргументация принадлежит Мандриоли.

Тем не менее существует подкласс языков сетей Петри, замкнутый по отношению к бесконечной конкатенации, — это класс языков *вполне оканчивающихся* сетей Петри. Это понятие введено для моделей вычисления, основанных на сложных биологических графах. Мы заимствуем его определение и переводим в термины сетей Петри.

Определение 6. Сеть Петри является *вполне оканчивающейся*, если всякий раз, когда ее выполнение заканчивается, 1) в сети остается только одна фишка, находящаяся в заключительной позиции; 2) число фишек, находившихся в сети Петри, конечно.

Заметим сначала, что вторая часть определения не обязательна, поскольку если сеть Петри оканчивает выполнение, то делает это за конечное число шагов и, следовательно, порождает только конечное число фишек. Первая же часть определения обязательна. Сеть Петри можно рассматривать как машину, порождающую строки символов. Мы помещаем фишку на вход машины, и для нас печатается строка символов. В конце концов обнаружится, что машина остановилась (т. е. не существует разрешенных переходов). В обычном подходе, когда не имеется отпечатанного выхода, необходимо анализировать текущее состояние машины и проверять, достигнута ли заключительная маркировка. Если она еще не достигнута, мы не принимаем выданную строку и снова пускаем машину. Если же сеть Петри вполне оканчивающаяся, то нет необходимости анализировать текущее состояние машины, поскольку гарантируется, что заключительная маркировка будет достигнута.

Теперь легко видеть, как можно использовать вполне оканчивающуюся сеть Петри для построения сети Петри, порождающей бесконечную конкатенацию ее языка. Мы просто соединяем заключительную и начальную позиции. Поскольку всякий раз, когда фишка появляется в заключительной позиции, сеть Петри, как мы знаем, бывает пуста, то при повторном использовании сети не будет запущен никакой нежелательный переход.

Этот подкласс сетей Петри не представляет большого интереса, поскольку можно показать, что между вполне оканчивающимися сетями Петри и конечными автоматами существует взаимно однозначное соответствие. Следовательно, языки вполне оканчивающихся сетей Петри являются регулярными, и, как уже известно, этот класс

языков замкнут по отношению к бесконечной конкатенации. Таким образом, мы убедились, что изучение свойств систем, моделируемых языками сетей Петри, ограничено конечными повторениями подсистем и бесконечными повторениями конечных подсистем.

2.6.5.8. Подстановка

Мы уже говорили, что системы можно проектировать и моделировать с помощью сетей Петри иерархическим образом. Это подразумевает сначала спецификацию схемы работы системы на уровне подсистем и затем последовательное уточнение этой схемы путем замены операций на их определения в терминах других операций. В случае применения сетей Петри уточнение может принимать форму подстановки целой сети Петри вместо перехода или позиции. Последнее не вызывает никаких трудностей, в связи с этим мы ограничим наше рассмотрение задачей подстановки подсети вместо перехода (операции) сети Петри.

Подстановку сети Петри вместо операции в другой сети Петри можно рассматривать как композицию языков двух сетей Петри. Поскольку операция представляется символом из Σ , подстановка языка сети Петри L_2 вместо символа σ в другом языке сети Петри L_1 определяется как замена всех появлений σ в L_1 на множество строк L_2 . Языки сетей Петри замкнуты по отношению к подстановке, если результат подстановки языка сети Петри есть также язык сети Петри. Вариациями подстановки являются *конечная подстановка*, при которой L_2 — конечное множество строк, и *гомоморфизм*, при котором L_2 — одна строка.

К сожалению, вновь результат отрицателен: языки сетей Петри не замкнуты по отношению к обычной подстановке. Это следует из примера подстановки $L_2 = a^n b^n$ вместо c в $L_1 = c^*$. Причиной опять является возможность повторного запуска сети Петри. В случае же конечной подстановки или гомоморфизма L_2 является регулярным языком, следовательно, можно построить вполне оканчивающуюся сеть Петри, порождающую его, из чего вытекают следующие теорема и следствие.

Теорема 7. Если L_2 — регулярный язык, а L_1 — язык сети Петри, то результат подстановки L_2 вместо любого символа L_1 — язык сети Петри.

Следствие 2. Языки сетей Петри замкнуты по отношению к конечной подстановке и гомоморфизму.

2.6.6. Языки сетей Петри и другие классы языков

Рассмотрев свойства языков сетей Петри как класса языков, мы обращаемся теперь к исследованию взаимосвязи между языками сетей Петри и другими классами языков. Рассмотрим, в частности, классы регулярных, контекстно-свободных, контекстно-связанных языков и языков типа 0.

2.6.6.1. Регулярные языки

Одним из простейших и наиболее изученных классов формальных языков является класс регулярных языков. Эти языки порождаются регулярными грамматиками и конечными автоматами и характеризуются регулярными выражениями. Задачи эквивалентности

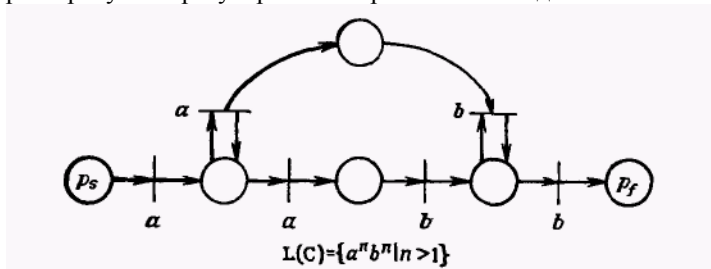


Рис. 6.12. Контекстно-свободный язык сети Петри, не являющийся регулярным.

и включения для двух регулярных языков разрешимы, известны алгоритмы их решения. Такой набор желательных свойств языка сделал справедливым следующее утверждение.

Теорема 8. Всякий регулярный язык — это язык сети Петри.

Доказательство этой теоремы основано на том, что всякий регулярный язык порождается некоторым конечным автоматом, а мы показали, что всякий конечный автомат можно преобразовать в эквивалентную сеть Петри.

Обратная теорема не верна. На рис. 12 изображена сеть Петри, порождающая контекстно-свободный язык $\{a^n b^n | n > 1\}$, который не является регулярным. Таким образом, не все языки сетей Петри являются регулярными.

2.6.6.2. Контекстно-свободные языки

Сеть на рис. 12 свидетельствует о том, что не все сети Петри являются регулярными, поскольку язык показанной сети Петри является контекстно-свободным, но не регулярным. Не все языки сетей Петри являются и контекстно-свободными, о чем свидетельствует сеть, показанная на рис. 13, имеющая контекстно-связанный, но не контекстно-свободный язык.

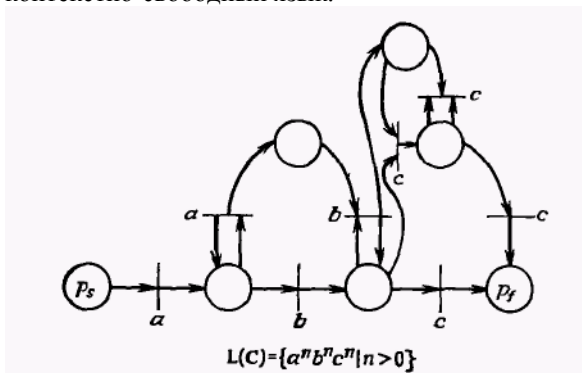


Рис. 13. Контекстно-связанный язык сети Петри, не являющийся контекстно-свободным.

В отличие от случая с регулярными языками существуют также контекстно-свободные языки, не являющиеся языками сетей Петри. Примером такого языка служит контекстно-свободный язык $\{ww^R \mid w \in \Sigma^*\}$. На основе этого доказывается следующее утверждение.

Теорема 9. Существуют контекстно-свободные языки, не являющиеся языками сетей Петри.

Доказательство. Допустим, что существует сеть Петри с n позициями и m переходами, порождающая язык ww^R . Пусть k — число символов в Σ , $k > 1$. Пусть для входной строки xx^R r равно $|x|$, т. е. длине строки x . Поскольку возможно k^r входных строк x , сеть Петри должна иметь k^r различных достижимых состояний, получающихся после запуска r переходов, для того, чтобы запомнить строку x . Если состояний меньше, то для некоторых строк x_1 и x_2 $\delta(\mu, x_1) = \delta(\mu, x_2)$, $x_1 \neq x_2$. Тогда очевидно, что $\delta(\mu, x_1 x_2^R) = \delta(\mu, x_1) x_2^R = \delta(\mu, x_2) x_2^R = \delta(\mu, x_2 x_2^R) \in F$, и сеть Петри будет порождать строку $x_1 x_2^R$, которую не должна порождать. Ранее мы показали, что для каждого перехода существует вектор

Кононюк А.Е. Графы

$v_j = e[j] \cdot D$, такой, что если $S(q, t_j)$ определено, то оно принимает значение $q + v_j$. Следовательно, после подачи r входных символов, т. е. запуска последовательности переходов t_1, t_2, \dots, t_r , получим состояние q_r :

$$q_r = \mu + \sum_j v_{i_j}.$$

Сумму в выражении можно представить другим способом:

$$q_r = \mu + \sum_j f_j v_j,$$

где f_j — число запусков перехода t_j в последовательности t_1, t_2, \dots, t_r ($f = (f_1, f_2, \dots, f_m)$ — вектор запуска). Кроме того, известно, что

$$\sum_j f_j = r.$$

В лучшем случае векторы v_1, v_2, \dots, v_m будут линейно независимы, и каждый вектор запуска (f_1, f_2, \dots, f_m) будет представлять единственное состояние q_r . Так как сумма коэффициентов f_i равна r , вектор (f_1, f_2, \dots, f_m) образует разбиение целого r на m частей. Как показал Кнут, число разбиений целого r на m частей равно

$$\binom{r+m-1}{m-1}.$$

Теперь, поскольку

$$\binom{r+m-1}{m-1} = \frac{(r+m-1) \dots (r+1)}{(m-1)!} < (r+m)^m,$$

то в множестве достижимости после запуска последовательности переходов длины r имеется строго меньше, чем $(r+m)^m$, достижимых состояний. Тогда для достаточно большого r имеем

$$\binom{r+m-1}{m-1} < (r+m)^m < k^r,$$

и, следовательно, невозможно иметь k^r различных состояний для всякой из k^r возможных входных строк. Таким образом, никакая сеть Петри не может порождать язык ww^R .

Доказательство этого факта проливает некоторый свет на ограниченность сетей Петри как машин, порождающих языки, и, следовательно, на природу языков сетей Петри. В сетях Петри нет возможности запомнить произвольно длинную последовательность произвольных символов. Из доказательства становится очевидно, что в сетях Петри можно запомнить последовательности ограниченной

Кононюк А.Е. Графы

длины (но это возможно и в конечных автоматах). Другой особенностью сетей Петри является способность запоминать число появлений символа, как, например, в языке $a^n b^n c^n$, в пределах, не доступных для систем, порождающих регулярные и контекстно-свободные языки. Однако сети Петри не обладают «объемом магазинной памяти», необходимым для порождения контекстно-свободных языков. Скорость роста пространства достижимых состояний с ростом длины входной строки для сетей Петри комбинаторна, но не экспоненциальна, т. е. «объема магазинной памяти» недо статочно.

Причина того, что сети Петри способны порождать языки, которые не могут порождаться автоматами с магазинной памятью, несмотря на меньший размер пространства состояний, заключается в более гибких взаимосвязях между состояниями сети Петри по сравнению с ограниченными путями между состояниями в автомате с магазинной памятью. Это объясняется ограничениями в автомате с магазинной памятью на доступ только к верхнему элементу магазина, тогда как в сетях Петри возможно обращение к любому ее счетчику в любой момент времени.

Теперь, когда мы показали, что не все контекстно-свободные языки являются языками сетей Петри и не все языки сетей Петри являются контекстно-свободными, естественно возникает вопрос: что из себя представляет класс языков, являющихся одновременно контекстно-свободными и языками сетей Петри? Одним подмножеством класса контекстно-свободных языков сетей Петри является класс регулярных языков. Другим — множество ограниченных контекстно-свободных языков, которые исследовались Гинзбургом.

2.6.6.3. Ограниченные контекстно-свободные языки

Контекстно-свободный язык L называется ограниченным контекстно-свободным, если в Σ^* имеются такие слова (строки) w_1, w_2, \dots, w_m , что

$$L \subseteq w_1^* w_2^* \dots w_m^* .$$

Термин «ограниченный» относится к конечному числу слов w_1, w_2, \dots, w_m . Гинзбург разработал детально описанную процедуру проверки свойств ограниченных контекстно-свободных языков. Он заметил, что в то время не было сведений о неразрешимых задачах, касающихся ограниченных контекстно-свободных языков.

Ограниченные контекстно-свободные языки характеризуются следующей теоремой, предложенной Гинзбургом.

Теорема 10. Семейство ограниченных контекстно-свободных языков является наименьшим семейством множеств, определяемых следующим образом:

1. Если W — конечное подмножество Σ^* , то W —ограниченный контекстно-свободный язык.
2. Если W_1 и W_2 — ограниченные контекстно-свободные языки, то $W_1 \cup W_2$ и W_1W_2 — ограниченные контекстно-свободные языки.
3. Если W — ограниченный контекстно-свободный язык и $x, y \in \Sigma^*$, то $\{x^iWy^j/i \geq 0\}$ — ограниченный контекстно-свободный язык.

Мы уже показали, что всякий конечный автомат, а также всякий регулярный язык и всякое конечное подмножество Σ^* являются языками сетей Петри. В разд. 2.6.5.1 и 2.6.5.2 показано, что языки сетей Петри замкнуты по отношению к конкатенации и объединению. Следовательно, необходимо показать только, что п. 3 также выполняется для языков сетей Петри. С этой целью построим сеть Петри $\gamma' = (C', \Sigma', \mu', F')$, порождающую язык $\{x^iWy^j/i \geq 0\}$ по данным сетям Петри стандартного вида $\gamma_x = (C_x, \Sigma_x, \mu_x, F_x)$, $\gamma_y = (C_y, \Sigma_y, \mu_y, F_y)$ и $\gamma_w = (C_w, \Sigma_w, \mu_w, F_w)$, порождающим x , y и W соответственно, γ' объединяет сети Петри γ_x , γ_y , γ_w и новую позицию p таким образом, что всякий раз после выполнения γ_x в p помещается фишка. Позиция p служит счетчиком числа выполнений сети γ_x . После выполнения необходимого числа раз сети γ_x выполнится γ_w и, наконец, выполняется повторно γ_y , удаляя каждый раз одну фишку из p . Поскольку входная строка порождается корректно, только если сеть Петри пуста (за исключением F' , определенного, как F_y), мы уверены, что число выполнений γ_x и γ_y равно.

Эта конструкция, иллюстрируемая на рис. 14 для $x = ab$, $y = b(a + b)$ и $W = b^+a$, показывает, что $\{x^iWy^j/i \geq 0\}$ — язык сети Петри.

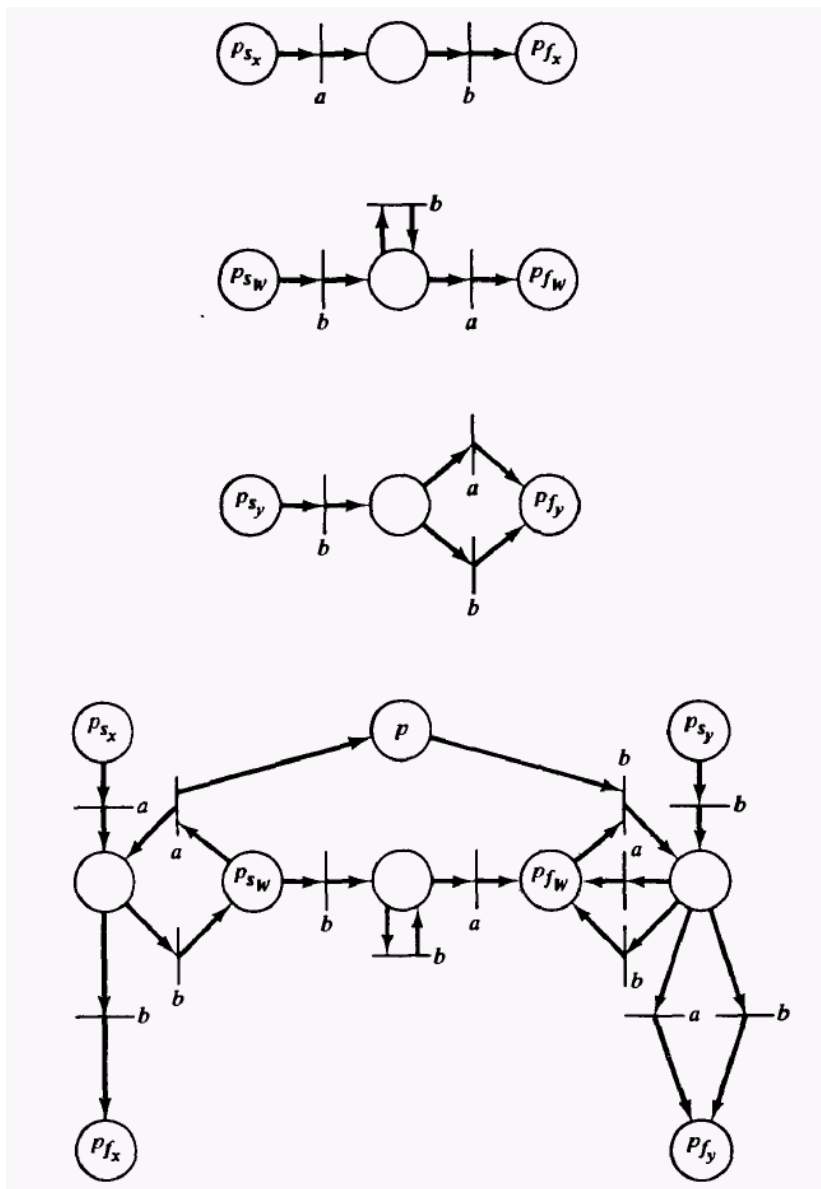


Рис. 14. Сеть Петри, порождающая язык $\{x^i W y^j | i \geq 0\}$. Эта конструкция показывает, что все ограниченные контекстно-свободные языки являются также языками сетей Петри.

Таким образом, любой ограниченный контекстно-свободный язык — это язык сети Петри.

Существуют ли контекстно-свободные языки, являющиеся языками сетей Петри, но не являющиеся ограниченными? К сожалению, ответ положителен. Гинзбург показал, что регулярное выражение $(a + b)^+$ не является ограниченным контекстно-свободным языком. Так как этот язык — контекстно-свободный язык сети Петри, очевидно, что ограниченные контекстно-свободные языки являются лишь собственным подмножеством семейства контекстно-свободных языков сетей Петри. Кроме того, язык $\{(a + b)^+ a^n b^n / n > 1\}$, являющийся контекстно-свободным, не ограничен и не регулярен. Для полной характеристики множества контекстно-свободных языков сетей Петри необходимы дальнейшие исследования.

Существование и регулярных множеств, и ограниченных контекстно-свободных языков в качестве подмножеств класса языков сетей Петри, оказывается полезным, поскольку оба класса языков имеют желательные свойства и некоторые благоприятные для анализа характеристики.

2.6.6.4. Контекстно-связанные языки

Мы исследовали взаимосвязь языков сетей Петри с регулярными языками и с контекстно-свободными языками. Обратимся теперь к контекстно-связанным языкам. На рис. 13 показан язык сети Петри, являющийся контекстно-связанным; мы докажем что все языки сетей Петри — контекстно-связанные. Поскольку известно, что все контекстно-свободные языки являются контекстно-связанными и существуют контекстно-свободные языки, не являющиеся языками сетей Петри, приходим к выводу, что существуют контекстно-связанные языки, не являющиеся языками сетей Петри. Следовательно, включение — собственное.

Теорема 11. Все языки сетей Петри являются контекстно-связанными. Доказательство того, что все языки сетей Петри являются контекстно-связанными, довольно сложно. Существуют два способа показать, что язык контекстно-связанный: построить контекстно-связанную грамматику, порождающую его, или определить недетерминированный линейно-ограниченный автомат, порождающий этот язык. Питерсон показал, как определяется контекстно-связанная грамматика, порождающая язык сети Петри. Здесь мы показываем как линейно-ограниченный автомат может породить язык сети Петри.

Кононюк А.Е. Графы

Линейно-ограниченный автомат подобен машине Тьюринга. Он имеет конечное множество состояний, головку чтения/записи и (бесконечную в обе стороны) ленту. Ограничением, отличающим его от машины Тьюринга, служит то, что длина участка ленты, который можно использовать, ограничивается линейной функцией от длины входной строки. В этом смысле он подобен автомату с магазинной памятью, порождающему контекстно-свободный язык (поскольку максимальный размер магазина ограничен линейной функцией от входной строки), за исключением того, что линейно-ограниченный автомат имеет произвольный доступ к своей памяти, тогда как магазинный автомат имеет доступ к памяти только с одного конца.

Для порождения языка сети Петри ее можно промоделировать, запоминая после каждого входного символа число фишек в каждой позиции. Какова скорость роста числа фишек как функции от длины входной строки? Рассмотрим число фишек, полученное после r запусков переходов. Это число, обозначаемое c , для последовательности переходов $t_{i_1} t_{i_2} \dots t_{i_r}$ равно

$$c = 1 + \sum_j (|O(t_{i_j})| - |I(t_{i_j})|)$$

Так как числа $\#(p_k, (O(t_j)))$ и $\#(p_k, (I(t_j)))$ и, следовательно, $|I(t_j)|$ и $|O(t_j)|$ (мощности входного и выходного комплектов) фиксированы для данной сети Петри, то существует максимальное значение разности l :

$$l = \max_{t_j \in T} (|O(t_j)| - |I(t_j)|).$$

Таким образом, $c = 1 + \sum_j (|O(t_{i_j})| - |I(t_{i_j})|) \leq 1 + r \cdot l$.

Число фишек, а следовательно, и объем памяти, необходимый для запоминания их, ограничены линейной функцией от длины входной строки. Поэтому языки сетей Петри могут порождаться линейно-ограниченными автоматами.

Этим доказательством мы показали, что все языки сетей Петри контекстно-связанные. Результаты нашего исследования взаимосвязи класса языков сетей Петри с другими классами языков сведены на рис. 15 и показаны там в виде графа и диаграммы Венна.

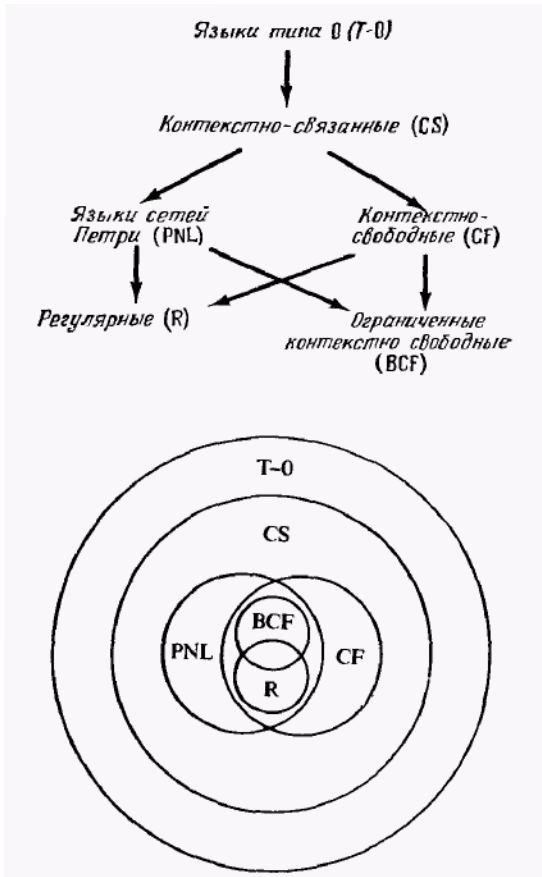


Рис. 15. Взаимосвязь языков сетей Петри с традиционными классами языков.

На рис. 16 приведены свойства языков сетей Петри.

Кононюк А.Е. Графы

Замкнутость по отношению к	L^λ	L	L^λ	P^f	P	P^λ
Объединению	Нет	Да	Да	?	Да	Да
Пересечению	Да	Да	Да	Да	Да	Да
Конкатенации	?	Да	Да	?	Да	Да
Параллельной композиции	?	Да	Да	?	Да	Да
Регулярной подстановке	Нет	λ -свободные	Да	?	Префиксные	Префиксные
Обратному гомоморфизму	Да	Да	Да	Да	Да	Да
Звезде Клини (итерации)	?	Нет	Нет	?	?	?
Дополнению	Нет	?	Нет	Нет	?	Нет
Содержит регулярные языки	Нет	Да	Да	Нет	Префиксные	Префиксные
Симметрическая разность с контекстно-свободными языками (не пуста)	Да	Да	Да	Да	Да	Да
Контекстно-связанный?	Да	Да	?	Да	Да	?

Рис. 16. Свойства некоторых языков сетей Петри .

Хэк исследовал множество задач разрешимости для языков сетей Петри. Задача принадлежности (является строка α элементом языка $L(\gamma)$?) разрешима, тогда как задача пустоты (является язык $L(\gamma)$ пустым?), как легко видеть, эквивалентна задаче достижимости. Неразрешимы задача равенства двух языков сетей Петри и задача их включения.

Эти результаты сведены на рис. 17.

Разрешимость задачи	L^f	L	L^λ	P^f	P	P^λ
Принадлежности	P	P	?	P	P	P
Пустоты	?	?	?	P	P	P
Конечности	?	?	?	P	P	P
Эквивалентности и включения	?	Н	Н	?	Н	Н

Рис. 17. Свойства разрешимости языков сетей Петри L- и P-типа (P — разрешима, Н — неразрешима).

Другой подход к изучению сетей Петри с использованием теории формальных языков рассматривался рядом авторов. Они заметили сходство между запуском переходов и применением правил порождения в выводе, понимая позиции как нетерминальные символы, а фишки — как отдельные экземпляры нетерминальных символов.

Основное отличие заключается в отсутствии в сетях Петри информации об упорядочении, которая содержится в выводе в виде предложений. Оно привело к определению *коммутативных грамматик*, изоморфных сетям Петри. Кроме того, рассматривалась взаимосвязь сетей Петри с матричными, контекстно-рассеянными, нетерминально-ограниченными языками; языками, ограниченными по выводу; равноматричными языками и языками Сциларда.

2.6.7. Помеченные сети и классы языков сетей Петри

Помеченная сеть Петри — эта пара (N, Σ) , где N — сеть Петри, $\Sigma: T \rightarrow A$ — *помечающая функция* над некоторым алфавитом A . Если Σ — частичная функция, т.е. некоторым переходам не сопоставляется никакого символа из A , то эти непомеченные переходы называют *λ -переходами* и помечают одним и тем же "пустым" символом λ . Помечающая функция Σ расширяется на последовательности срабатываний естественным образом:

$$\Sigma(\tau t) = \begin{cases} \Sigma(\tau) \Sigma(t), & \text{если } \Sigma(t) \text{ определено,} \\ \Sigma(\tau) & \text{в противном случае,} \end{cases}$$

где $\tau t \in T^*$. При этом $\Sigma(\lambda) = \lambda$, где λ — пустое слово.

Помеченные сети с λ -переходами удобны в тех случаях, когда при моделировании системы нужно ввести вспомогательные переходы, не связанные непосредственно с событиями системы, а используемые для некоторых специальных целей моделирования. С их помощью можно также "маскировать" те события, которые не должны рассматриваться в данной задаче моделирования.

На рис. 17, а показан пример сети Петри, на рис. 17, б, в и г — помеченные сети над алфавитом $\{a, b, c, d\}$, полученные из первой сети с помощью следующих помечающих функций:

$$\begin{aligned} \Sigma_1 : \Sigma_1(t_1) = a, \Sigma_1(t_2) = c, \Sigma_1(t_3) = b, \Sigma_1(t_4) = d, \\ \Sigma_2 : \Sigma_2(t_1) = a, \Sigma_2(t_2) = a, \Sigma_2(t_3) = b, \Sigma_2(t_4) = b, \\ \Sigma_3 : \Sigma_3(t_1) = \lambda, \Sigma_3(t_2) = a, \Sigma_3(t_3) = b, \Sigma_3(t_4) = c. \end{aligned}$$

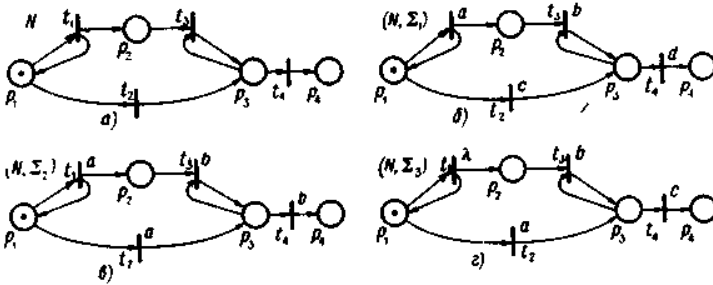


Рис. 17

Если $\tau \in T^*$ — последовательность срабатываний сети Петри N , а (N, Σ) — помеченная сеть, то $\Sigma(t) \in A^*$ называют *помечающей последовательностью*. Если $L(N)$ — свободный язык сети N , то множество $\{\Sigma(\tau) \mid \tau \in L(N)\}$ образует *префиксный язык* помеченной сети (N, Σ) .

Во многих приложениях бывает удобно или необходимо рассматривать не свободный язык сети Петри N , включающий все ее последовательности срабатываний, а его подмножество *терминальных последовательностей*, которое состоит из всех последовательностей, ведущих от начальной разметки M_0 к некоторой фиксированной *терминальной разметке* M_f , т.е. множество $L(N, M_f) = \{\tau \in T^* \mid M_0[\tau] M_f\}$. Множество $L(N, M_f)$ образует *свободный терминальный язык* сети N . Соответственно множество $\{\Sigma(\tau) \mid \tau \in L(N)\}$ образует *терминальный язык* помеченной сети (N, Σ) .

Для сети N на рис. 17, в свободный язык представляет собой множество

$$L(N) = \{t_1^{n_1} t_2^{n_2} t_3^{n_3} t_4^{n_4} \mid (n_1 > 0) \wedge (1 \geq n_2 > 0) \wedge (n_1 \cdot n_2 \geq n_3 \geq 0) \wedge (n_2 \geq n_4 > 0)\},$$

а свободный терминальный язык $L(N, M_f)$, где $M_f = (0, 0, 0, 1)$ (т.е. при M_f только место p_4 содержит фишку), представляет собой множество

$$\underline{L}(N, M_f) = \{t_1^n t_2 t_3^n t_4 \mid n \geq 0\}.$$

Поскольку для помеченной сети на рис. 17, б помечающая функция Σ_1 осуществляет взаимно-однозначное отображение множества T на алфавит A , то ее префиксный язык и терминальный языки получаются из вышеприведенных языков прямой заменой символов-переходов на соответствующие символы из A . Для помеченных сетей на рис. 17, в и г

Кононюк А.Е. Графы

их префиксные и терминальные языки (для $M_f = (0, 0, 0, 1)$) образуют множества:

$$L(N, \Sigma_2) = \{a^n b^m \mid n \geq m \geq 0\},$$

$$L(N, \Sigma_2, M_f) = \{a^n b^m \mid n \geq 0\},$$

$$L(N, \Sigma_3) = \{a^{n_1} b^{n_2} c^{n_3} \mid (1 \geq n_1 \geq 0) \wedge (n_1 = 0 \Rightarrow n_2 = 0) \wedge (n_1 = 1 \Rightarrow n_2 > 0) \wedge (n_1 \geq n_3 \geq 0)\},$$

$$L(N, \Sigma_3, M_f) = \{a b^n c \mid n \geq 0\}.$$

Пусть N — класс всех сетей Петри. На основе введенных выше определений языков разного типа для сетей Петри и помеченных сетей можно образовать различные классы языков сетей Петри, из которых выделим следующие.

1) Класс \mathcal{L}^λ префиксных языков сетей Петри включает префиксные языки всех помеченных сетей, которые можно образовать из сетей класса N с помощью произвольных помечающих функций над произвольными алфавитами. Верхний индекс λ указывает, что помечающие функции могут быть частичными, т.е. помеченные сети могут содержать λ -переходы.

2) Класс \mathcal{L}^{λ_0} терминальных языков сетей Петри включает терминальные языки всех помеченных сетей, образованных из сетей класса N , в том числе помеченных сетей с λ -переходами.

3) Классы \mathcal{L} и \mathcal{L}_0 являются подклассами классов \mathcal{L}^λ и соответственно \mathcal{L}^{λ_0} и включают префиксные и терминальные языки всех тех помеченных сетей, которые не содержат λ -переходов, т.е. которые образованы из сетей класса N только с помощью всюду определенных помечающих функций.

4) Классы \mathcal{L}^t , \mathcal{L}_0^t включают свободные префиксные языки и свободные терминальные языки всех сетей Петри. Если условиться, что существует специальный класс всюду определенных помечающих функций типа $\Sigma: T \rightarrow T$, т.е. функций над алфавитом $A = T$, причем

$\forall t \in T: \Sigma(t) = t$, то классы \mathcal{L}^t и \mathcal{L}_0^t можно считать подклассами классов \mathcal{L} и соответственно \mathcal{L}_0 .

Для сопоставления друг с другом введенных выше языков и классов сетей Петри полезной оказывается специальная стандартная форма помеченных сетей. Сеть, преобразованная в стандартную форму, сохраняет префиксный и терминальный языки, хотя в стандартной форме и появляются новые переходы и места.

Помеченная сеть представлена в стандартной форме, если:

1) выделено специальное "включающее" место on с начальной разметкой $M_o(on) = 1$ и начальная разметка всех остальных мест равна 0;

Кононюк А.Е. Графы

2) терминальный язык сети всегда определяется для одной и той же терминальной разметки $M_f = \bar{0} = (\underbrace{0, \dots, 0}_n)$, где $n = |P|$ — число мест в сети;

3) каждый переход сети имеет хотя бы одно входное место (т.е. разметка 0 является тупиковой для сети).

Лемма 1. Для любой помеченной сети (N, Σ) и любой терминальной разметки M_f этой сети существует представленная в стандартной форме помеченная сеть (N', Σ') такая, что $L(N', \Sigma') = L(N, \Sigma)$ и $L(N', \Sigma', \bar{0}) = L(N, \Sigma, M_f)$.

Доказательство. Пусть $N = (P, T, F, W, M_0)$. Если эта сеть содержит переходы без входных или выходных мест, т.е. не удовлетворяет третьему условию стандартности формы, то в сеть добавляется новое "стандартное" место q , которое делается входным и одновременно выходным для каждого перехода сети (такие места уже вводились выше). Начальная разметка M_0 распространяется на новую сеть так, что $M_0(q) = 1$. Введение места q не изменяет ни условий срабатывания переходов исходной сети, ни разметку ее мест после срабатывания переходов, поэтому новая сеть сохраняет свободный, префиксный и терминальный языки заданной сети (последний с учетом добавления к M_f компоненты $M_f(q) = 1$). На рис. 18, а и б показан пример описанного преобразования помеченной сети, содержащей переходы с пустым множеством входных или выходных мест, в сеть удовлетворяющую третьему условию стандартности.

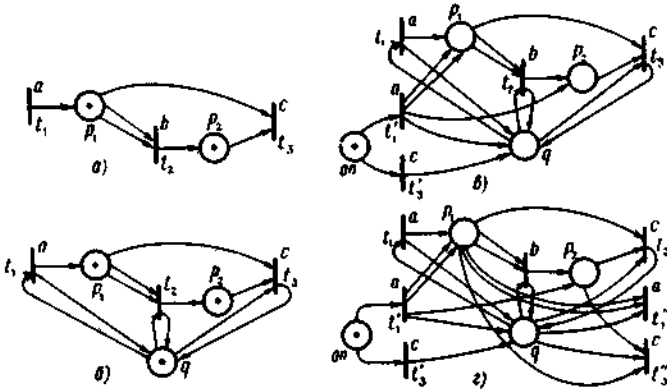


Рис. 18.

Для того чтобы выполнилось первое условие стандартности, добавим к сети включающее место op с единичной разметкой, разметку всех

Кононюк А.Е. Графы

остальных мест сделаем нулевой, а для каждого перехода $\tau \in T$, могущего работать при начальной разметке M_0 , введем новый переход t' с пометкой $\Sigma(t') = \Sigma(t)$. Каждый из новых переходов имеет в качестве единственного места включающее место op , а его выходные дуги заводятся на места сети так, чтобы после срабатывания перехода t' разметка этих мест равнялась бы разметке преобразуемой схемы после срабатывания перехода t при M_0 , а именно:

$$F^*(t') = M_0 - F^*(t) + F^*(t).$$

Легко видеть, что любое слово $\Sigma(\tau) \in L(N, \Sigma)$, $\tau \in L(N)$, принадлежит также языку новой сети и порождается ее последовательностью τ' , отличающейся от τ лишь первым символом перехода (t' в τ' вместо t в τ). Наоборот, любая последовательность срабатываний τ' новой сети должна начинаться некоторым символом перехода t' , так как ни один из "старых" переходов сети не может сработать из-за того, что все места, кроме места op , имеют нулевую разметку. Продолжение последовательности срабатываний τ' должно далее совпадать с продолжением некоторой последовательности срабатываний τ старой сети, начинающейся символом перехода t . На рис. 18, б и в показан пример описанного преобразования. В исходной сети на рис. 18, б при разметке M_0 могут сработать переходы t_1 и t_3 , для которых введены дополнительные переходы t'_1 и t'_3 в "новой сети" на рис. 18, в. Срабатывание перехода t_1 меняет разметку M_0 на разметку $(2, 1, 1)$, где переход q считается третьим в упорядоченном наборе мест, а срабатывание перехода t_3 меняет разметку M_0 на разметку

$$(0, 0, 1). \text{ Поэтому в новой сети } F^*(t'_1) = (2, 1, 1) \text{ и } F^*(t'_3) = (0, 0, 1).$$

Чтобы выполнить второе требование стандартности формы, для каждого перехода $t \in T$ такого, что он последним может сработать перед терминальной разметкой M_f (т.е. $M_f \geq F^*(t)$), добавляется новый переход t' с пометкой $\Sigma(t'') = \Sigma(t)$. Входные дуги, ведущие в переход t'' , заводятся так, чтобы

$${}^*F(t'') = M - F^*(t) + {}^*F(t),$$

а выходных дуг переход t'' не имеет (т.е. $F^*(t'') = 0$).

Из построения следует, что t'' может сработать в новой сети только в том случае, если t может сработать (так как ${}^*F(t'') \geq F^*(t)$), и что срабатывание t'' приводит к нулевой разметке $\bar{0}$ в том и только том случае, если срабатывание t в сети N приводит к разметке M_f . Соответственно последовательности срабатываний τ и τ' такие, что

$$M_0(\tau) M_f, M_0(\tau') \bar{0}$$

Кононюк А.Е. Графы

длина $|\tau| = |\tau'| = 2$, отличаются лишь последними символами переходов (t'' в τ' вместо t в Σ). В случае, если $|\tau| = |\tau'| = 1$, т.е. $M_0 [t > M_f]$, для t добавляется еще переход t''' с $\Sigma(t''') = \Sigma(t)$. Его единственным входным местом служит место op и $F(t''') = \bar{0}$.

На рис.18, *в* и *г* показан пример описанного преобразования для терминальной разметки $M_f(3,0,1)$, где q - третий элемент набора мест. Эта разметка достижима в сети на рис. 18, *б* после последовательностей срабатываний, заканчивающихся символами переходов t_1 или t_3 . В первом случае имеет место изменение разметки $(2, 0, 1) [t_1 > (3, 0, 1)$, во втором $(-4, 1, 1) [t_3 > (3, 0, 1)$. В сеть добавляются переходы t''_1 и t''_3 с

$$\cdot F(t''_1) = (3, 0, 1) - (1, 0, 1) + (0, 0, 1) = (2, 0, 1),$$

$$\cdot F(t''_3) = (3, 0, 1) - (0, 0, 1) + (1, 1, 1) = (4, 1, 1),$$

$$\Sigma(t''_1) = a, \Sigma(t''_3) = c. \square$$

Следствие. Любой язык из классов $\mathcal{L}, \mathcal{L}^\lambda, \mathcal{L}_0, \mathcal{L}_0^\lambda$ может быть порожден некоторой помеченной сетью в стандартной форме как ее терминальный язык для терминальной разметки $M_f = \bar{0}$.

Это позволяет переносить любые результаты о языках сетей, представленных в стандартной форме, на общий случай языков сетей Петри.

Следующие теоремы сравнивают введенные выше классы языков сетей Петри.

Теорема 12.

$$a) \mathcal{L} \subseteq \mathcal{L}_0, b) \mathcal{L} \subseteq \mathcal{L}_0^\lambda.$$

Доказательство. Сначала установим, что $\mathcal{L} \subseteq \mathcal{L}_0$ и $\mathcal{L}^\lambda \subseteq \mathcal{L}_0^\lambda$, а затем убедимся, что эти включения являются на самом деле строгими. Покажем, что для любой помеченной сети (N, Σ) можно построить помеченную сеть (N', Σ') такую, что

$$\mathcal{L}(N, \Sigma) = \mathcal{L}(N', \Sigma') = \mathcal{L}(N', \Sigma', \bar{0}),$$

т.е. префиксный и терминальный (для $M_f = \bar{0}$) языки сети (N', Σ') равны друг другу и равны префиксному языку сети (N, Σ) . При этом, если сеть (N, Σ) не имеет λ -переходов, то сеть (N', Σ') также не должна иметь λ -переходов. Заменяем каждый переход $t \in T$ множеством переходов

Кононюк А.Е. Графы

$$\{t^j \mid \cdot F(t^j) = \cdot F(t) \wedge F \cdot (t^j) \leq F \cdot (t)\}.$$

Здесь индекс j меняется от 0 до k , где k равно числу различных векторов длины $|P|$, меньших или равных $F \cdot (t)$, и пусть при $j=0$ новый переход t^0 является точной копией перехода t , т.е. $P \cdot (t^0) = F \cdot (t)$. Кроме того, полагаем, что для всех $t^j \sum(t^j) = \sum(t)$, т.е. все новые переходы помечаются так же, как и заменяемый ими старый переход. На рис. 19, б показана помеченная сеть, построенная описываемым способом по помеченной сети на рис. 19, а.

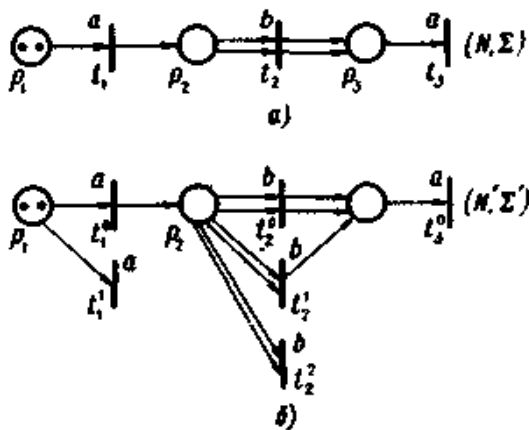


Рис. 19

Каждый из новых переходов t^j может сработать точно при той же разметке, что и замещенный старый переход. Но после срабатывания он может выдать меньше фишек, так из него выходит меньше дуг (при $j \neq 0$), чем из t .

Рассмотрим произвольную последовательность срабатываний τ' сети (N', Σ') , приводящую к некоторой разметке M' , т.е. $M_0[\tau' > M']$. Если заменить в τ' каждый переход t^j на t , то получим последовательность срабатываний τ сети (N, Σ) , причем $M_0[\tau > M \wedge M \geq M']$.

Наоборот, если задана последовательность срабатываний $\tau \in L(N, \Sigma)$ такая, что $M_0[\tau > M]$, то можно последовательно заменить в ней каждый переход t на t^j так, чтобы на каждом очередном шаге замены выбирался

Кононюк А.Е. Графы

t^j с наименьшим вектором $F^*(t^j)$, позволяющим, однако, сохранить оставшуюся часть последовательности. В результате последнее срабатывание "новой" последовательности τ' должно привести к нулевой разметке в сети (N', Σ') . В обоих случаях последовательности срабатываний τ и τ' порождают одно и то же слово $\Sigma(\tau) = \Sigma'(\tau')$.

Например, если выбрать $\tau' = t^0_1 t^0_1 t^2_2$ такую, что $(2, 0, 0) [\tau > (0, 0, 0)$, в сети на рис. 19, б. то для нее существует $\tau \in L(N, \Sigma)$, где (N, Σ) — сеть на рис. 19, а, причем

$$\tau = t_1 t_1 t_2, (2, 0, 0) (\tau) (0, 0, 1), \Sigma(\tau) = aab = \Sigma(\tau').$$

Наоборот, для $\tau = t_1 t_1 t_2 t_3$, где $\tau \in L(N, \Sigma)$ и (N, Σ) — сеть на рис. 19, а и $(2, 0, 0) [\tau > (0, 0, 1)$, существует $\tau' \in L(N', \Sigma')$, где (N', Σ') — сеть на рис. 19, б и $\tau' = t^0_1 t^0_1 t^1_2 t^0_3$, причем $(2, 0, 0) [\tau' > (0, 0, 1)$, а $\Sigma'(\tau') = aaba = \Sigma(\tau)$. Таким образом, из способа построения сети (N', Σ') следует, что $L(N, \Sigma) = L(N', \Sigma') = L(N', \Sigma', \bar{0})$. Например, для сети (N, Σ) на рис. 19, а префиксный язык

$$L(N, \Sigma) = \{a, aa, aab, aaba, aabaa\}.$$

Для сети (N', Σ') на рис. 3.3, б префиксный язык

$$L(N', \Sigma') = L(N', \Sigma', \bar{0}) = \{a, aa, aab, aaba, aabaa\} = L(N, \Sigma).$$

Следствием установленного соотношения между префиксным языком сети (N, Σ) и префиксным и терминальным языками сети (N', Σ') является включение $L \subseteq L_0$.

Чтобы установить справедливость соотношения б), достаточно убедиться, что в любую помеченную сеть Петри можно ввести дополнительные λ -переходы таким образом, чтобы нулевая разметка была достижима от любой разметки сети и префиксные языки новой, и старой сети совпадали. Для этого достаточно для каждого места заданной сети ввести свой λ -переход без выходных дуг, который изымает при своем срабатывании по одной фишке из этого места. Новые λ -переходы могут срабатывать в любое время, уменьшая разметку сети. Для любой последовательности срабатываний τ исходной сети найдется такая последовательность срабатываний τ' новой сети, что $\Sigma'(\tau) = \Sigma(\tau')$. Обратное также верно, поэтому префиксные языки обеих сетей совпадают. Отсюда следует, что $L^\lambda \subseteq L_0^\lambda$. Теперь убедимся, что установленные включения являются строгими.

Легко построить помеченную сеть в стандартной форме, с λ -переходами или без них, порождающую терминальный язык $L = \{t, ttt\}$. Однако, ясно, что никакая сеть не может порождать этот язык в качестве префиксного, так как наряду со словом ttt он должен содержать слово tt . Отсюда следует, что $L \subseteq L_0$ и $L^\lambda \subseteq L_0^\lambda$.

Кононюк А.Е. Графы

Теорема 13.

а) $\mathcal{L}'_0 \subset \mathcal{L}_0$, б) $\mathcal{L}' \subset \mathcal{L}$.

Доказательство. Из определения упоминаемых в формулировке теоремы языков следует, что $\mathcal{L}'_0 \subseteq \mathcal{L}_0, \mathcal{L}' \subseteq \mathcal{L}$. Остается показать, что имеет место строгое включение в обоих этих случаях. Для доказательства справедливости строгих включений в этой и в следующей теоремах будет использован один и тот же прием, а именно: будут предъявлены языки, принадлежащие правым классам языков и не входящие в левые классы языков.

а) Легко построить помеченную сеть в стандартной форме, порождающую терминальный язык $L_1 = \{t, ttt\}$ (см. доказательство теоремы 12), но никакая сеть Петри не может породить такой же свободный терминальный язык. Действительно, пусть переход t — единственный потенциально живой переход в некоторой сети Петри N .

Он может сработать при начальной разметке M_0 , т.е. $M_0 \geq F(t)$. Разметка M , возникающая после срабатывания перехода t , может удовлетворять или не удовлетворять условию $M \geq F(t)$. В первом случае язык $L(N)$ будет также содержать все слова вида t^n . Во втором случае слово t оказывается единственной последовательностью срабатываний, т.е. слово ttt не входит в язык L_1 .

Таким образом, $L_1 \in \mathcal{L}_0$ и $L_1 \notin \mathcal{L}'_0$, т.е. $\mathcal{L}' \subset \mathcal{L}_0$.

б) Легко построить помеченную сеть, порождающую префиксный язык

$$L_2 = \{a, aa, aab, b, ba, baa\},$$

но никакая сеть Петри не может породить L_2 в качестве свободного языка.

Предположим противное, и пусть сеть $N = (P, T, F, W, M_0)$ порождает язык $L(N) = L_2$. Предположим, что первым сработал переход $a \in T$ (слово $a \in L_2$) и изменил разметку M_0 на M . Далее возможны два случая.

1) $M \geq F(b)$, где $b \in T$. В этом случае переход b может сработать непосредственно после a , в результате чего слово $ab \in L(N)$, что противоречит предположению $L(N) = L_2$.

2) $M \not\geq F(b)$. В этом случае в сети N существует место p такое, что $p \in a'$, $p \in b'$ и имеют место два неравенства:

$$M(p) < F(p, b), \quad M(p) < M_0(p).$$

Кононюк А.Е. Графы

Тогда для места p и для разметки M' такой, что $M_0 [a > M [a > M'$, имеет место неравенство

$$M'(p) < M(p) < F(p, b),$$

т.е. переход b не может сработать при разметке M' и слово $aab \notin L(N) = L_2$, что противоречит предположению.

Таким образом, $L_2 \in \mathcal{L}$ и $L_2 \notin \mathcal{L}'$, т.е. $\mathcal{L}' \subset \mathcal{L}$.

Теорема 14.

$$\text{а) } \mathcal{L} \subset \mathcal{L}^\lambda, \text{ б) } \mathcal{L}_0 \subset \mathcal{L}^\lambda.$$

Доказательство. а) Рассмотрим язык

$$L = \{ \alpha a^k \mid \alpha \in \{0, 1\}^* \wedge (0 \leq k \leq n(\alpha)) \},$$

где α — двоичный код, a — символ, $n(\alpha)$ — целое неотрицательное число, задаваемое двоичным кодом α . При этом полагаем, что $n(\lambda) = 0$. Этот язык порождается (как префиксный) помеченной сетью Петри с λ -переходами, показанной на рис. 20.

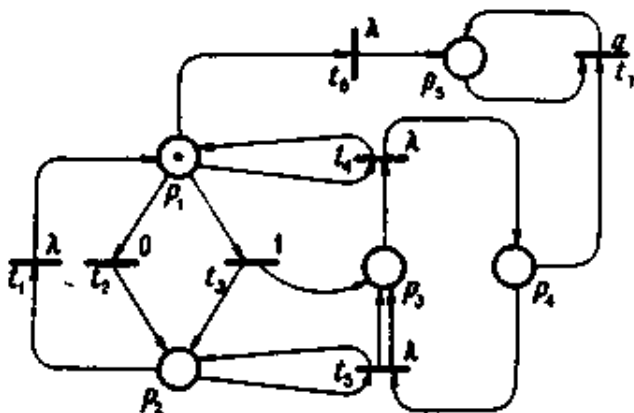


Рис. 20

В этой сети поочередные срабатывания переходов t_2 и t_3 , помеченных соответственно символами 0 и 1, порождают некоторое двоичное слово α . Если при этом срабатывают также переходы t_4 и t_5 , то в местах p_3 и p_4 накапливаются фишки. Можно убедиться, что - суммарное число фишек в обоих местах не может превысить $n(\alpha)$. Если после того, как породится префикса, сработает переходу, он переместит фишку из места p_1 в место p_5 . При дальнейшей работе сети может срабатывать только переход t_7 , помеченный символом a , причем число срабатываний этого перехода не может превысить $n(\alpha)$. Таким образом, сеть

Кононюк А.Е. Графы

на рис. 20 порождает помечающие последовательности, принадлежащие языку L .

Наоборот, любое слово $\alpha\alpha^k$, $0 \leq k \leq n$ (α), из языка L может быть порождено этой сетью. Действительно, эта сеть порождает любое двоичное слово α в качестве помечающей последовательности. Если каждый из λ -переходов t_4 и t_5 срабатывает максимально возможное для них число раз, то в месте p_4 накопится ровно $n(\alpha)$ фишек. После этого переход t_7 может сработать любое число раз, не превышающее $n(\alpha)$.

Покажем теперь методом от противного, что язык L не может быть префиксным языком ни одной помеченной сети Петри без λ -переходов. С этой целью допустим существование такой сети с k местами. Пусть m_0 — суммарное число фишек во всех k местах сети при начальной разметке, а m — максимальное число фишек, которое может добавиться в сети при срабатывании какого-то перехода. Тогда после n срабатываний любых переходов сети последняя содержит не более чем $m_0 + mn$ фишек. Среди k мест они могут быть расположены не более чем $(m_0 + nm + 1)^k$ способами. Таким образом, число различных разметок данной сети, достижимых за не более чем n шагов, не превышает $(m_0 + nm + 1)^k$, что, в свою очередь, строго меньше 2^n при достаточно большом n .

С другой стороны, имеется 2^n различных двоичных слов длины n . Каждое такое слово представляет целое число $n(\alpha)$, где $0 \leq n(\alpha) \leq 2^n - 1$. Пусть все эти слова упорядочены в последовательность $\alpha_0, \alpha_1, \dots, \alpha_{2^n-1}$, где $n(\alpha_i) = i$ для $i = 0, 1, \dots, 2^n - 1$. Для каждого слова α_i из этой последовательности должна существовать по крайней мере одна разметка M_i , которая достигается после порождения помечающей последовательности α_i и начиная от которой можно породить слово $\alpha_i\alpha^i$. Если все такие разметки $M_0, M_1, \dots, M_{2^n-1}$ различны, то мы получаем противоречие, поскольку мы установили, что в данной сети число различных разметок, достижимых в пределах n шагов, строго меньше 2^n .

Предположим, что для некоторых i и j таких, что $i \neq j$, имеет место $M_i = M_j$. Тогда в рассматриваемой сети можно породить слово $\alpha\alpha^r$, где $l = \min(i, j)$, $r = \max(i, j)$. Но это слово не входит в язык L , следовательно, мы вновь приходим к противоречию.

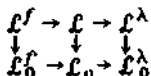
Таким образом, показано, что язык L входит в \mathcal{L}^k и не входит в \mathcal{L} , т.е. $\mathcal{L} \subset \mathcal{L}^k$.

б) Если в качестве терминальной разметки для сети на рис. 20 взять разметку $M_f = \bar{0}$ и добавить λ -переходы, очищающие все места от фишек, то окажется, что язык L порождается как терминальный язык

Кононюк А.Е. Графы

этой сети. В то же время доказательство того факта, что никакая сеть без λ -переходов не может породить язык L , относится как к префиксным, так и к терминальным языкам. Отсюда следует, что $\mathcal{L}_0 \subset \mathcal{L}_0^\lambda$.

Установленные в теоремах 12, 13, 14 соотношения между классами языков сетей Петри можно изобразить следующей схемой, в которой стрелки указывают отношение включения:



Таким образом, оказывается, что классы терминальных языков более мощны, чем классы префиксных языков, помеченные сети Петри оказываются более мощным моделирующим инструментом, чем непомеченные, а λ -переходы еще больше усиливают моделирующие способности сетей.

2.6.8. Характеризация классов языков сетей Петри

В теории алгоритмов и автоматов рассматриваются различные абстрактные системы (машины, автоматы), предназначенные для моделирования функционирования дискретных систем. Их "выразительная мощность", т.е. способность адекватно описывать сложное поведение моделируемых систем, часто характеризуется классами порождаемых ими языков, которые, как и в случае языков сетей Петри, определенным образом кодируют разные возможные способы функционирования систем. Если, например, системы из класса S_1 порождают класс языков \mathcal{L}_1 , а системы из класса S_2 порождают класс языков \mathcal{L}_2 и $\mathcal{L}_1 \supseteq \mathcal{L}_2$, то говорят, что класс систем S_1 мощнее класса систем S_2 , а если $\mathcal{L}_1 \supset \mathcal{L}_2$, то класс систем S_1 строго мощнее класса систем S_2 . Классы S_1 и S_2 равноможны, если $\mathcal{L}_1 = \mathcal{L}_2$. В теории формальных языков выделены и изучены некоторые классы языков, порождаемых системами разного типа. Эти же классы языков порождаются конечными множествами правил, называемых *порождающими грамматиками*. Каждая грамматика представляет собой набор (A, V, Π, S_0) , где

A — алфавит терминальных символов,

V — алфавит нетерминальных символов,

Π - конечное множество *продукций* (или *правил подстановки*),

S_0 - *начальный символ*.

Продукция имеет вид $\alpha \rightarrow \beta$, где $\alpha \in V^*$, $\beta \in (A \cup V)^*$. Продукция $\alpha \rightarrow \beta$ может быть применима к некоторому слову вида $\delta_1 \alpha \delta_2$ и

Кононюк А.Е. Графы

преобразует его в слово $\delta_1\beta\delta_2$. Последовательность слов $\gamma_0, \gamma_1, \gamma_2, \dots, \gamma_n$ такая, что слово $\gamma_i, 1 \leq i \leq n$, получено из слова γ_{i-1} с помощью некоторой продукции из Π , называется *выводом* в данной грамматике, а слово γ_n *выводимо* в ней из слова γ_0 . Язык, порождаемый грамматикой, — это множество всех терминальных слов (слов из A^*), выводимых из слова S_0 с помощью продукций из Π .

Класс языков, порождаемых произвольными грамматиками (нет ограничений на множество продукций Π), совпадает с классом всех рекурсивно перечислимых множеств слов и поэтому может быть назван классом рекурсивно перечислимых языков. Известно, что класс языков, порождаемых машинами Тьюринга или машинами Минского, является классом рекурсивно перечислимых языков. Поскольку это наиболее широкий класс языков, соответствующие классы абстрактных машин можно считать "универсально мощными".

Если каждая продукция в Π имеет вид

$$\alpha_1 S \alpha_2 \rightarrow \alpha_1 \beta \alpha_2,$$

где $S \in V; \alpha_1, \alpha_2 \in V^*, \beta \in (A \cup V)^*$, то грамматика порождает *контекстно-зависимый язык*.

Если каждая продукция в Π имеет вид

$$S \rightarrow \beta, \text{ где } S \in V \text{ и } \beta \in (A \cup V)^*,$$

то грамматика порождает *контекстно-свободный язык*. Класс контекстно-свободных языков является собственным подклассом класса контекстно-зависимых языков. Контекстно-свободные языки порождаются магазинными автоматами и играют важную роль как синтаксические модели современных языков программирования.

Если каждая продукция в Π имеет вид $S \rightarrow S'\beta$ или каждая продукция имеет вид $S \rightarrow \beta S'$, где $\beta \in (A \cup V)^*$, а S - пустой символ или $S \in V$, то грамматика порождает *регулярный язык*. Класс регулярных языков является собственным подклассом класса контекстно-свободных языков. Регулярные языки порождаются конечными автоматами и поэтому называются также *автоматными языками*.

В этом разделе будет охарактеризована выразительная мощность сетей Петри путем сравнения классов языков сетей Петри с вышеперечисленными классами языков.

Прежде всего сравним классы языков сетей Петри к классом \mathcal{L}_a регулярных языков. Регулярный язык порождается *конечным автоматом*, который представляет собой набор (A, Q, I) , где

A — алфавит,

Q - конечное непустое множество *состояний автомата* ($Q \cap A = \emptyset$), которое содержит выделенное начальное состояние q_0 и заключительное состояние q_f .

Кононюк А.Е. Графы

I — программа автомата, представляющая собой множество команд - слов вида $qa \rightarrow q'$, в которых $q, q' \in Q$, $a \in A$, и для любой пары (q, a) существует единственная команда, начинающаяся этими символами.

Конечный автомат представим в виде графа, множеством вершин которого является множество Q . Из вершины q в вершину q' ведет дуга, помеченная символом a , в том и только в том случае, если программа автомата содержит команду $qa \rightarrow q'$. Среди вершин графа выделены начальная, соответствующая состоянию q_0 и заключительная соответствующая заключительному состоянию q_f . Функционирование автомата состоит в процессе продвижения по дугам графа начиная от начальной вершины, и чтения символов, помечающих дуги. Автомат останавливается, если и только если достигает заключительной вершины, а слово, прочитанное автоматом при его движении от начальной до заключительной вершины, называется словом, допускаемым автоматом. Множество всех слов, допускаемых автоматом, образует его язык. На рис. 21,а показан пример конечного автомата, допускающего язык $\{a^n b^m \mid n > 0, m > 0\}$. Этот автомат имеет четыре состояния $\{q_0, q_1, q_2, q_3\}$, заключительное состояние — q_2 (отмечено двойным кружком).

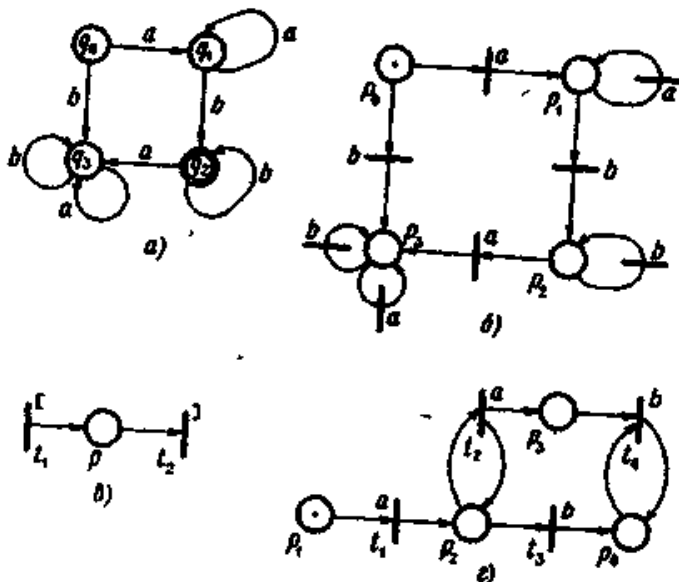


Рис. 21.

Теорема 15.

а) $\mathcal{L}_a \subset \mathcal{L}_0$, б) $\mathcal{L}_a \subset \mathcal{L}$.

Доказательство. а) Любой конечный автомат, порождающий язык L над алфавитом A , можно легко преобразовать в помеченную сеть без λ -переходов, порождающую такой же терминальный язык, следующим простым образом. Каждому состоянию $q_i \in Q$ автомата сопоставляется место p_i в сети Петри, каждая дуга "пересекается" переходом и этот переход помечается тем же символом из A , что и эта дуга. Начальная разметка задается так, что единственную фишку содержит место, соответствующее начальному состоянию, все остальные места имеют нулевую разметку. В качестве терминальной разметки берется разметка, при которой имеет фишку только место, соответствующее заключительному состоянию.

На рис. 21,б показана помеченная сеть, построенная таким образом по автомату на рис. 21,а. Простой анализ способа построения помеченной сети по конечному автомату убеждает в том, что язык, допускаемый автоматом, совпадает с терминальным языком построенной сети. Тем самым показано, что $\mathcal{L}_a \subseteq \mathcal{L}_0$.

Однако существуют терминальные языки сетей, не являющиеся регулярными. Например, на рис. 21,в показана помеченная сеть, в которой помечающая функция сопоставляет одному переходу символ открывающей скобки [, другому — символ закрывающей скобки]. Для терминальной разметки $M_f = \bar{0}$ эта сеть порождает терминальный язык, совпадающий с так называемым скобочным языком, слова которого представляют собой "правильно вложенные" последовательности открывающих и закрывающих скобок. Известно, что скобочный язык является контекстно-свободным (и не регулярным) и задается следующей грамматикой:

$$S_0 \rightarrow S_0 S_0.$$

$$S_0 \rightarrow [S_0],$$

$$S_0 \rightarrow \lambda.$$

Здесь [,] - символы открывающей и закрывающей скобок, λ — пустой символ. Таким образом, класс регулярных языков является собственным подклассом класса \mathcal{L}_0 .

б) Префиксным регулярным языком называют множество всех слов, являющихся префиксами (в том числе пустыми) слов некоторого регулярного языка. Префиксные регулярные языки являются регулярными языками и представляют собой префиксы слов, допускаемых конечными автоматами. Поэтому сеть Петри, построенная по конечному автомату описанным выше способом,

Кононюк А.Е. Графы

порождает префиксный язык, совпадающий с префиксным регулярным языком, допускаемым автоматом. Отсюда следует, что $\mathcal{L}_a \subseteq \mathcal{L}$.

С другой стороны, существуют префиксные языки сетей Петри, не являющиеся регулярными. Например, сеть Петри на рис. 21, в порождает в качестве префиксного языка так называемый префиксный скобочный язык, слова которого являются префиксами правильно вложенных последовательностей открывающих и закрывающих скобок. Этот язык задается следующей контекстно-свободной грамматикой:

$$S_0 \rightarrow S_0 S_0,$$

$$S_0 \rightarrow [S_0],$$

$$S_0 \rightarrow \{S_0,$$

$$S_0 \rightarrow \lambda.$$

Из теорем 12-15 следует, что каждый из классов языков \mathcal{L} , \mathcal{L}^* , \mathcal{L}_0 , $\mathcal{L}^{\neq 0}$ включает все регулярные языки, причем отношение включения является собственным, так как эти классы содержат и нерегулярные языки, такие как скобочный язык или контекстно-свободный язык $\{a^n b^n \mid n > 0\}$. Последний порождается как терминальный язык для $M_f = (0, 0, 0, 1)$ помеченной сетью, показанной на рис. 21, г.

В качестве примеров нерегулярных языков сетей Петри мы выбрали контекстно-свободные языки. Однако не все языки сетей Петри являются контекстно-свободными и, наоборот, не все контекстно-свободные языки порождаются сетями. Следующие две теоремы подтверждают сказанное.

Теорема 16. *Существует терминальный язык, порождаемый помеченной сетью Петри и не являющийся контекстно-свободным.*

Доказательство. Известно, что язык $\{a^n b^n c^n \mid n > 0\}$ не является контекстно-свободным. На рис. 22 показана помеченная сеть, порождающая этот язык в качестве терминального для $M_f = (0, 0, 0, 0, 1)$.

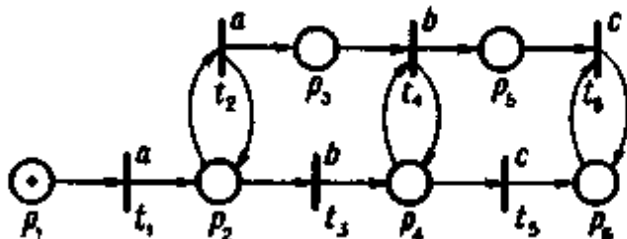


Рис. 22

Теорема 17. *Существует контекстно-свободный язык, не порождаемый ни одной {помеченной} сетью Петри.*

Кононюк А.Е. Графы

Доказательство. Рассмотрим контекстно-свободный язык L в алфавите $\{a, b, c\}$, порождаемый следующей грамматикой:

$$S_0 \rightarrow S_0 c S_0,$$

$$S_0 \rightarrow S,$$

$$S \rightarrow SS,$$

$$S \rightarrow aSb,$$

$$S \rightarrow \lambda.$$

Этот язык обобщает скобочный язык, рассмотренный в доказательстве теоремы 14, если отождествить a с открывающей скобкой $[$, b - с закрывающей скобкой $]$. Символы c как бы разделяют правильно вложенные последовательности скобок.

Предположим, что язык L порождается некоторой помеченной сетью Петри (N, Σ) в качестве терминального языка. Подязык $\{a^n b^n c \mid n > 0\}$ языка L также порождается этой сетью. Тогда эта сеть должна порождать бесконечные последовательности срабатываний. Зафиксируем некоторую бесконечную последовательность срабатываний и соответствующую ей последовательность векторов-разметок. Обозначим через M_k такую разметку, которая возникает после того, как сеть породила префикс a^k . По лемме 2.2 для зафиксированной последовательности разметок существует такая пара чисел i и j , что $i \neq j$ и $M_i \leq M_j$ или $M_j \leq M_i$. Пусть для определенности

$M_i \leq M_j$. Рассмотрим слово $a^i b^i c \in L$. При достижении разметки M_i сеть N порождает префикс a^i этого слова. Рассмотрим слово $a^j b^j c \in L$. При достижении разметки M_j сеть N порождает префикс a^j этого слова. Пусть сеть N' отличается от сети N лишь начальной разметкой, которая заменена на M_i , а сеть N'' также отличается от N лишь начальной разметкой, которая заменена на M_j . Терминальный язык сети (N', Σ) содержит слово $b^i c$. Терминальный язык сети (N'', Σ) содержит, в силу свойства монотонности языков сетей Петри, слово $b^j c$, потому что $M_i \leq M_j$. Следовательно, слово $a^i b^i c$ содержится в терминальном языке сети N , что противоречит предположению о совпадении этого языка с языком L . Таким образом, контекстно-свободный язык L не принадлежит классу \mathcal{L}_0^{λ} . Поскольку этот класс языков — максимальный в семействе классов языков, порождаемых сетями Петри (теорема 13), то язык L не может входить ни в один из классов языков сетей Петри.

Теорема 18. *Класс помеченных сетей Петри строго мощнее класса конечных автоматов, не сравним с классом магазинных автоматов и строго менее мощен, чем класс машин Тьюринга.*

Доказательство. Теорема является непосредственным следствием теорем 12, 13-17.

2.6.9. Разрешимые и неразрешимые свойства языков сетей Петри

Типичная массовая алгоритмическая проблема для формальных языков состоит в том, что требуется установить существование алгоритма, который для произвольного языка L , заданного порождающей грамматикой или другой порождающей системой, устанавливает, обладает ли этот язык некоторым свойством. Например, проблема принадлежности связана с проверкой, принадлежит ли произвольное слово a языку L ; в проблеме пустоты следует выяснить, пусто ли множество L ; в проблеме конечности задача состоит в выяснении, является ли L конечным множеством. Проблемы эквивалентности и включения грамматик или других порождающих языки систем формулируются как проверка равенства или включения порождаемых ими языков (как множеств).

Известно, что в классе \mathcal{L}_a регулярных языков, порождаемых конечными автоматами, все перечисленные проблемы разрешимы. В классе контекстно-свободных языков разрешимы проблемы принадлежности, пустоты и конечности, но не разрешима проблема эквивалентности контекстно-свободных грамматик. В этом параграфе рассматриваются соответствующие проблемы для введенных выше классов языков сетей Петри.

Теорема 19. *Проблема принадлежности разрешима для языков из классов \mathcal{L} , \mathcal{L}_0 , \mathcal{L}^λ .*

Доказательство. Пусть сеть Петри N имеет множество переходов T . Для любого слова $\tau \in T^*$ легко устанавливается, является ли τ последовательностью срабатываний переходов этой сети. Действительно, достаточно проверить, может ли сработать при начальной разметке M_0 переход t , символ которого стоит первым в слове τ , затем, изменив разметку M_0 на M_1 , где $M_0[t > M_1$, проверить, сможет ли сработать при M_1 переход, символ которого стоит вторым в слове τ , и т.д. Так как слово τ содержит конечное число символов, процедура последовательных проверок завершится за конечное число шагов. В любой помеченной сети (N, Σ) без λ -переходов каждой помечающей последовательности α соответствует конечное число последовательностей срабатываний таких, что $\Sigma(\tau) = \alpha$. Поэтому процедура проверки принадлежности слова α префиксному языку $L(N, \Sigma)$ или терминальному языку $L(N, \Sigma, M_f)$ завершается за конечное число шагов. Следовательно, проблема принадлежности в классах \mathcal{L} и \mathcal{L}_0 разрешима.

Кононюк А.Е. Графы

Для выяснения принадлежности слова α языку помеченной сети с λ -переходами может быть применена следующая процедура. Слово $\alpha = a_1 a_2 \dots a_n$ длины n входит в префиксный язык сети $(N_\alpha, \Sigma_\alpha)$, показанной на рис. 23.а.

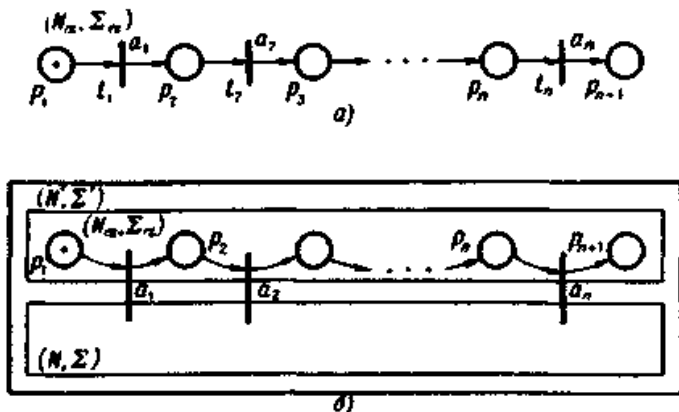


Рис. 23.

В этой сети $\sum_\alpha(t_i) = a_i$ для $1 \leq i \leq n$ и место p_{n+1} может получить фишку в том и только том случае, если выполнится последовательность срабатываний $t_1 t_2 \dots t_n$, порождающая слово α . Пусть (N, Σ) - произвольная помеченная сеть с λ -переходами, для которой выясняется, принадлежит ли α языку $L(N, \Sigma)$. Строится новая сеть, получаемая "пересечением" сетей (N, Σ) и $(N_\alpha, \Sigma_\alpha)$ по одинаково помеченным переходам так, как показано на рис. 18,б. В построенной сети (N', Σ') место p_{n+1} получит фишку в том и только том случае, если $\alpha \in L(N, \Sigma)$. Ранее было установлено, что существует алгоритм, с помощью которого можно узнать, получит ли данное место в сети Петри хотя бы одну фишку. Отсюда следует и разрешимость проблемы принадлежности произвольного слова α языку из класса \mathcal{E}^λ .

Теорема 20. *Проблема достижимости заданной разметки в сети Петри сводима к проблеме принадлежности для языков из класса \mathcal{E}^{λ_0} .*

Доказательство. Пусть сеть Петри N , для которой выясняется вопрос о достижимости некоторой разметки, имеет множество мест

$P = \{p_1, p_2, \dots, p_n\}$ и множество переходов $T = \{t_1, t_2, \dots, t_m\}$.

Преобразуем сеть N в помеченную сеть (N', Σ) с λ -переходами следующим образом (рис. 24). Все переходы сети N оставляем непомеченными (они становятся λ -переходами в сети (N', Σ)). Новые места и переходы добавляются так, как показано на рис. 24.

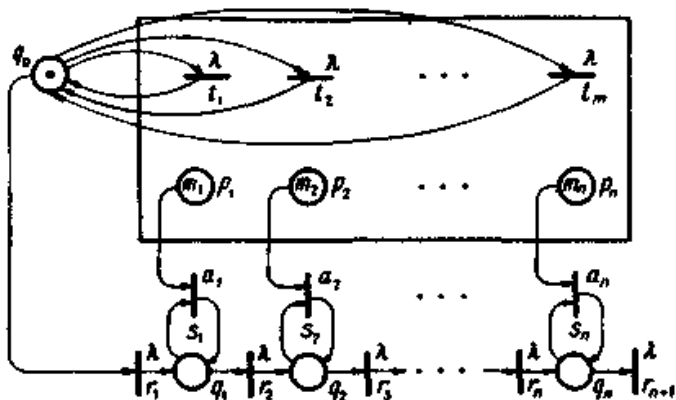


Рис. 24.

Новые переходы $r_1, r_2, \dots, r_n, r_{n+x}$ являются λ -переходами, а переходы S_1, S_2, \dots, S_n помечены символами a_1, a_2, \dots, a_n из алфавита A . Начальная разметка новой сети совпадает для мест p_1, p_2, \dots, p_n с начальной разметкой $M_0 = (m_1, m_2, \dots, m_n)$ сети N , место q_0 имеет одну фишку, места q_1, \dots, q_n имеют нулевую разметку.

Пока место q_0 в порожденной сети содержит фишку, эта сеть работает так же, как сеть N . и достигает некоторой разметки M , после чего может сработать λ -переход r_1 . С этого момента единственный путь достижения нулевой разметки $\bar{0}$ в новой сети состоит в том, чтобы реализовалась терминальная последовательность срабатываний

$$\tau = r_1 s_1^{M(p_1)} r_2 s_2^{M(p_2)} \dots r_n s_n^{M(p_n)} r_{n+1},$$

где $s_i^{M(p_i)}$ означает, что символ перехода s_i входит в τ столько раз подряд, сколько фишек содержит место p_i при разметке M . Последовательность срабатываний τ порождает помечающую последовательность $a_1^{M(p_1)} a_2^{M(p_2)} \dots a_n^{M(p_n)}$. Таким образом, в построенной помеченной сети имеется возможность кодировать достижимые разметки из множества $R(N)$ словами в алфавите A и

$$L(N', \Sigma, \bar{0}) = \{ a_1^{m_1} a_2^{m_2} \dots a_n^{m_n} \mid (m_1, m_2, \dots, m_n) \in R(N) \}.$$

Следовательно, для выяснения принадлежности некоторой разметки (m_1, m_2, \dots, m_n) множеству $R(N)$ достаточно построить указанным способом помеченную сеть (N', Σ) и выяснить, принадлежит ли слово $a_1^{m_1} a_2^{m_2} \dots a_n^{m_n}$ терминальному языку $L(N', \Sigma, \bar{0})$. Сеть на рис. 24 представлена в стандартной форме, поэтому, с учетом леммы 1 и

следствия из нее, проблема достижимости разметки может быть сведена к проблеме принадлежности слова к терминальному языку из класса \mathcal{L}^{λ}_0 .

Формулировку проблемы пустоты для языков сетей Петри приходится модифицировать по сравнению с обычной, так как языки сетей Петри всегда содержат хотя бы одно слово, а именно — пустое слово λ . Для префиксных языков из классов \mathcal{L} и \mathcal{L}^{λ} можно считать, что они не пусты, если содержат хотя бы одно слово, отличное от λ . Терминальный язык $L(N, \Sigma, M_f)$ из класса \mathcal{L}_0 или \mathcal{L}^{λ}_0 можно считать пустым тогда и только тогда, когда пуст свободный терминальный язык $L(N, M_f)$, последний, в свою очередь, пуст тогда, когда $M_f \notin R(N)$. После уточнения формулировок легко доказываются следующие теоремы.

Теорема 21. *Проблема пустоты разрешима для префиксных языков из классов \mathcal{L} и \mathcal{L}^{λ} .*

Доказательство. Для произвольного языка из \mathcal{L} или \mathcal{L}^{λ} , порождаемого помеченной сетью, достаточно проверить, может ли сработать при начальной разметке хотя бы один переход этой сети, помеченный некоторым символом из помечающего алфавита (символом, отличным от λ).

Теорема 22. *Проблема достижимости разметки и проблема пустоты терминальных языков из классов \mathcal{L}_0 и \mathcal{L}^{λ}_0 эквивалентны.*

Доказательство. Так как для любого терминального языка $L(N, \Sigma, M_f)$ верно, что он пуст тогда и только тогда, когда $M_f \notin R(N)$, то проверка пустоты языка сводится к проверке достижимости разметки M_f в сети N , и наоборот.

Теорема 23. *Проблема конечности разрешима для префиксных языков из классов \mathcal{L} и \mathcal{L}^{λ} .*

Доказательство. Пусть задан язык, порождаемый помеченной сетью (N, Σ) . Он бесконечен, если и только если содержит бесконечную помечающую последовательность. Добавим к сети (N, Σ) новое место-счетчик p , которое является выходным местом всех помеченных переходов, и $p' = \emptyset$. В процессе функционирования сети после каждого срабатывания помеченного перехода место-счетчик получает фишку. Это место ограничено тогда и только тогда, когда сеть (N, Σ) не порождает ни одной бесконечной помечающей последовательности. Поскольку проблема ограниченности некоторого места в сети Петри разрешима, разрешимой является и проблема конечности префиксных языков.

Теорема 24. *Проблема достижимости разметки сводима к проблеме конечности терминальных языков из классов \mathcal{L}_0 и \mathcal{L}^{λ}_0 .*

Кононюк А.Е. Графы

Доказательство. С учетом теоремы 22 достаточно показать, что проблема пустоты терминальных языков сводима к проблеме их конечности. Пусть порождающая заданный терминальный язык помеченная сеть представлена в стандартной форме. Добавим к сети помеченный некоторым символом переход и этот переход соединим входной и выходной дугами с включающим местом op . Новый переход не влияет на достижимость терминальной нулевой разметки. Если для исходной сети существует терминальная помечающая последовательность (для $M_f = \bar{0}$), то тогда в модифицированной сети существует сколь угодно длинная терминальная помечающая последовательность, получаемая за счет того, что сеть начинает работу с произвольно большого числа срабатываний добавленного перехода. Поэтому терминальный язык модифицированной сети бесконечен в том и только том случае, если язык исходной сети не пуст.

При решении проблем эквивалентности и включения для языков сетей Петри предполагается, что все помеченные сети определены над одним и тем же алфавитом (полученным, например, объединением алфавитов, для которых заданы разные помечающие функции).

Проблемы эквивалентности и включения оказываются неразрешимыми для классов языков $\mathcal{L}, \mathcal{L}^\lambda, \mathcal{L}_0, \mathcal{L}_0^\lambda$, причем доказательство неразрешимости этих проблем проводится аналогично доказательству неразрешимости проблем R -эквивалентности и R -включения для сетей Петри, а именно сведением к рассматриваемым проблемам неразрешимой проблемы включения графов полиномов. Напомним, что граф $g(f)$ для полинома $f(x_1, x_2, \dots, x_n)$ с неотрицательными целыми коэффициентами представляет собой множество

$$\{(x_1, \dots, x_n, y) \in \mathbf{N}^{n+1} \mid y \leq f(x_1, \dots, x_n)\}.$$

Для сведения проблемы включения графов полиномов к проблеме включения устраивается кодировка графов языками над помечающим алфавитом. При этой кодировке каждому вектору, принадлежащему данному графу полинома, взаимно однозначно сопоставляется слово кодирующего языка. Кодировка опирается на предположенный Париком метод отображения слов в заданном алфавите $A = \{a_1, \dots, a_n\}$ в целочисленные векторы. Отображение Парика для алфавита A представляет собой функцию $\#: A^* \rightarrow \mathbf{N}^n$ такую, что для $\alpha \in A^*$ вектор $\#\alpha$ содержит в качестве i -го компонента целое неотрицательное число, равное числу вхождений символа $a_i \in A$ в слово α .

Например, для алфавита $A = \{a_1, a_2, a_3\}$ и слова $\alpha = a_2 a_1 a_2$ вектор $\#\alpha = (2, 2, 0)$.

Отображение Парика распространяется на языки над алфавитом A естественным образом:

Кононюк А.Е. Графы

$\#L = \{M \in \mathbb{N}^n \mid \exists \alpha \in L: M = \#\alpha\}$.

Лемма 2. Для любого полинома $f(x_1, \dots, x_n)$ с неотрицательными целыми коэффициентами можно построить помеченную сеть (N, Σ) без λ -переходов такую, что префиксный язык $L(N, \Sigma)$ кодирует граф $g(f)$ полинома с помощью отображения Парика для алфавита $A = \{a_0, \dots, a_n, a_{n+1}\}$ следующим образом: $L(N, \Sigma)$ является максимальным языком, для которого $\#L(N, \Sigma) = g(f)$.

Доказательство. Пусть N' — сеть Петри, слабо вычисляющая полином $f(x_1, \dots, x_n)$. Строится помеченная сеть (N, Σ) способом, указанным на рис. 25.

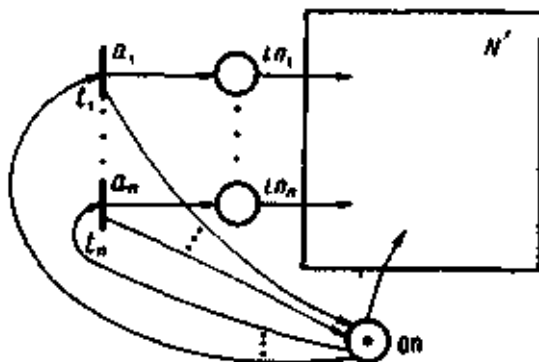


Рис. 25

Добавляются новые переходы t_1, \dots, t_n по одному для каждого входного места in_1, \dots, in_n сети N' , стартовое место on сети N' делается входным и выходным местом каждого из добавленных переходов, переход t_i ,

$1 \leq i \leq n$, соединяется выходной дугой с местом in_i , соответствующим переменной x_i в полиноме. Помечающая функция Σ выбирается таким образом, что $\Sigma(t) = a_{n+1}$ для любого старого перехода t сети N' и $\Sigma(t_i) = a_i$ для каждого нового перехода t_i , $1 \leq i \leq n$, сети N . Следовательно, сеть (N, Σ) не содержит λ -переходов.

Начальная разметка M_0 сети (N, Σ) такова, что $M_0(on) = 1$ и $M_0(in_i) = 0$ для всех i от 1 до n . В сети (N, Σ) ни один из старых переходов, помеченных символом a_{n+1} , не может сработать, пока один из них не удалит (насовсем) фишку из места on . Это означает, что все срабатывания новых переходов в сети (N, Σ) предшествуют срабатываниям старых переходов, общих для сетей N' и N . Поэтому префиксный язык $L(N, \Sigma)$ удовлетворяет следующим свойствам:

Кононюк А.Е. Графы

$$1) L(N, \Sigma) \subseteq \{a_1, \dots, a_n\}^* \circ \{a_{n+1}\}^*,$$

где \circ - операция попарной конкатенации слов из двух множеств слов $\{a_1, \dots, a_n\}^*$ и $\{a_{n+1}\}^*$;

2) для любой помечающей последовательности $\alpha \in L(N, \Sigma)$ верно, что $\#\alpha = (m_1, \dots, m_n, m_{n+1})$, где m_i , $1 \leq i \leq n$, — число срабатываний перехода t_i , а m_{n+1} — число срабатываний всех старых переходов сети N' , причем m_{n+1} не превышает $f(m_1, \dots, m_n)$.

Так как класс префиксных языков \mathcal{L} , является подклассом классов \mathcal{L} , \mathcal{L}_0 , \mathcal{L}^{λ_0} (теоремы 12 и 14), то можно считать, что граф любого полинома с целыми неотрицательными коэффициентами можно кодировать описанным в лемме 2 способом с помощью языков из классов \mathcal{L} , $\mathcal{L}_0, \mathcal{L}^{\lambda_0}$.

Лемма 3. Для любой пары помеченных сетей (N_1, Σ_1) и (N_2, Σ_2) можно построить помеченную сеть (N, Σ) такую, что

$$L(N, \Sigma) = L(N_1, \Sigma_1) \cup L(N_2, \Sigma_2)$$

и

$$L(N, \Sigma, \bar{0}) = L(N_1, \Sigma_1, M_{f_1}) \cup L(N_2, \Sigma_2, M_{f_2}),$$

где M_{f_1} и M_{f_2} — произвольные заданные терминальные разметки первой и второй сетей.

Доказательство. Если сети (N_1, Σ_1) и (N_2, Σ_2) представлены не в стандартной форме, их следует преобразовать в таковую способом, указанным в доказательстве леммы 1. Сеть (N, Σ) строится следующим образом: представленные в стандартной форме сети (N_1, Σ_1) и (N_2, Σ_2) объединяются в одну сеть так, что их включающие места op_1 и op_2 совмещаются в одно место (с одной фишкой в нем). Если обе исходные сети не содержали λ -переходов, то их не будет и в объединенной сети. Последняя удовлетворяет всем требованиям стандартности, а любая порождаемая ею помечающая последовательность является помечающей последовательностью сети (N_1, Σ_1) или сети (N_2, Σ_2) в зависимости от того, переход какой из составляющих сетей первым заберет фишку из объединенного включающего места op . Таким образом, префиксный и терминальный (для $M_f = \bar{0}$) языки сети (N, Σ) являются объединениями соответствующих языков заданных сетей. \square Из леммы 3 следует, что классы языков \mathcal{L} , \mathcal{L}^{λ} , \mathcal{L}_0 и \mathcal{L}^{λ_0} замкнуты относительно операции теоретико-множественного объединения, т.е. язык, полученный объединением любых двух языков из одного из этих классов, также принадлежит этому классу.

Теорема 25. Проблемы включения и эквивалентности неразрешимы для языков сетей Петри из классов \mathcal{L} , \mathcal{L}^{λ} , \mathcal{L}_0 и \mathcal{L}^{λ_0}

Доказательство. Пусть f_1 и f_2 — два полинома с целыми неотрицательными коэффициентами и требуется узнать $g(f_1) \subseteq g(f_2)$. Если

Кононюк А.Е. Графы

бы проблема включения для языков из класса \mathcal{L} была разрешима, то достаточно было бы закодировать $g(f_1)$ и $g(f_2)$ языками $L(N_1, \Sigma_1)$ и $L(N_2, \Sigma_2)$ из \mathcal{L} по способу, указанному в лемме 2, и выяснить, $L(N_1, \Sigma_1) \subseteq L(N_2, \Sigma_2)$. Это означало бы, что проблема включения графов полиномов разрешима, что не верно. Поэтому не является разрешимой и проблема включения для языков из класса \mathcal{L} . Так как $\mathcal{L} \subseteq \mathcal{L}^\lambda$, $\mathcal{L} \subseteq \mathcal{L}_0$ и $\mathcal{L} \subseteq \mathcal{L}_0^\lambda$, то проблема включения неразрешима и для языков из классов $\mathcal{L}^\lambda, \mathcal{L}_0, \mathcal{L}_0^\lambda$.

Так как для произвольных двух языков L_1 и L_2 включение $L_1 \subseteq L_2$ имеет место тогда и только тогда, когда $L_1 \cup L_2 = L_3$ и каждый из классов языков $\mathcal{L}, \mathcal{L}^\lambda, \mathcal{L}_0$ и \mathcal{L}_0^λ замкнут относительно операции объединения, то проблема включения для языков из этих классов легко сводится к проблеме их эквивалентности. Действительно, для того чтобы установить справедливость включения $L_1 \subseteq L_2$, достаточно проверить равенство языков $L_3 = L_2$, где $L_3 = L_1 \cup L_2$. Отсюда следует, что проблема эквивалентности языков сетей Петри (классы $\mathcal{L}, \mathcal{L}^\lambda, \mathcal{L}_0$ и \mathcal{L}_0^λ неразрешима).

В табл. 1 суммированы сведения о разрешимых (P) и неразрешимых (H) свойствах классов языков сетей Петри, а также о свойствах, к анализу которых может быть сведена проблема достижимости (C).

Таблица 1

Проблемы Классы языков	Принадлежности	Пустоты и конечности	Эквивалентности и включения
\mathcal{L}	P	P	H
\mathcal{L}^λ	P	P	H
\mathcal{L}_0	P	C	H
\mathcal{L}_0^λ	C	C	H

Рассмотрим, как сильно локальные изменения в структуре сетей Петри влияют на порождаемые ими языки. Следующие теоремы говорят о том, что такие "малые" изменения, как удаление одного перехода или одной фишки из сети, приводят к нераспознаваемым изменениям в порождаемых языках.

Теорема 26. *Неразрешимой является проблема, изменится ли префиксный или терминальный язык сети, если из нее удалить некоторый переход.*

Доказательство. Пусть сеть N имеет вид, показанный на рис.26,а, где N_1 и N_2 — ее подсети, каждая из которых представлена в стандартной форме со специальными включающими местами op_1 и op_2 , не содержащими, однако, фишек.

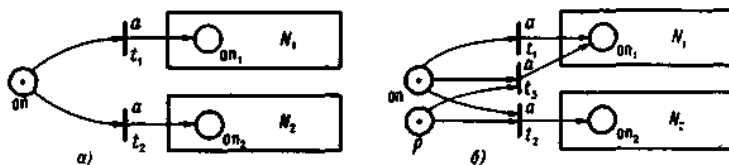


Рис. 26.

Место $оп$ является "включающим" местом в сети N , переходы t_1 и t_2 помечены одним и тем же символом a , который не может помечать переходы подсетей N_1 и N_2 .

Обозначим через L префиксный язык $L(N, \Sigma)$, через L_i — язык $L(N_i, \Sigma)$, где $i = 1, 2$, через L_0 — терминальный язык $L(N, \Sigma, M_f)$ для некоторой терминальной разметки M_f , через L_{0i} — язык $L(N_i, \Sigma, M_f(P_i))$, где P_i — множество мест подсети L_i , $i=1,2$.

Имеют место очевидные равенства:

$$L = a \circ (L_1 \cup L_2) \cup \{\lambda\}, \quad L_0 = a \circ (L_{01} \cup L_{02}).$$

(Считаем, что начальная разметка M_0 сети N не выбирается в качестве терминальной разметки M_f)

Пусть L' и L'_0 обозначают префиксный и терминальный языки сети, полученной удалением из сети N перехода t_2 вместе с инцидентными дугами. Тогда,

$$L' = a \circ L_1 \cup \{\lambda\}, \quad L'_0 = a \circ L_{01}.$$

Из этих равенств следует, что

$$L = L' \iff L_2 \subseteq L_1, \quad L_0 = L'_0 \iff L_{02} \subseteq L_{01}.$$

Следовательно, предположение о разрешимости сформулированной в теореме проблемы приводит к разрешимости проблемы R -включения для языков из классов \mathcal{L} и \mathcal{L}_0 , что противоречит предыдущей теореме.

Теорема 27. *Не разрешимой является проблема, изменится ли префиксный или терминальный язык сети, если из некоторого ее места удалить фишку.*

Доказательство. Пусть сеть N на рис. 26,б отличается от сети, рассмотренной в доказательстве предыдущей теоремы (рис. 26,а), лишь тем, что в нее добавлены еще один переход t_3 , в который ведет дуга из места $оп$ и из которого ведет дуга на "включающее" место подсети N_1 , и еще одно место p , из которого заведены дуги на переходы t_2 и t_3 . Новый переход помечен тем же символом a .

Сеть N' будет отличаться от сети N лишь начальной разметкой, в которой место p не имеет фишки. Пусть L' и L'_0 обозначают соответственно префиксный и терминальный языки сети N' . Тогда

Кононюк А.Е. Графы

$$L = \sigma \circ (L_1 \cup L_2) \cup \{\lambda\}, \quad L_0 = \sigma \circ (L_{01} \cup L_{02}),$$

$$L' = \sigma \circ L_1 \cup \{\lambda\}, \quad L'_0 = \sigma \circ L_{01}.$$

Вновь, как и в предыдущей теореме, имеем

$$L = L' \iff L_2 \subseteq L_1, \quad L_0 = L'_0 \iff L_{02} \subseteq L_{01},$$

откуда следует неразрешимость сформулированной в теореме проблемы.

2. 7. Подклассы сетей Петри

Сети Петри моделируют широкий спектр дискретных систем (с учетом теоремы 18), но для некоторых распространенных специальных классов систем удобно применять сети Петри не общего вида, а некоторые их подклассы, более простые и более адекватные рассматриваемым системам. Кроме того, проблемы анализа свойств сетей общего вида оказываются или неразрешимыми или достаточно сложными. Поэтому вводились и исследовались различные подклассы сетей Петри, получаемые в основном путем упрощения топологии (графовой структуры) сетей; некоторые из этих подклассов рассматриваются в этой главе.

2.7.1. Ординарные сети Петри

В определении сети Петри присутствует функция W , задающая кратность дуг, и для срабатывания перехода t при функционировании сети требуется, чтобы каждое его входное место p имело разметку не меньшую, чем кратность дуги, соединяющей p и t , а после срабатывания перехода t разметка любого его выходного места q увеличивается на кратность дуги, соединяющей t и q . В общем случае кратность дуги может быть любым неотрицательным числом, но если все дуги имеют одну и ту же кратность 1, то сеть относится к подклассу ординарных сетей. Для срабатывания перехода в ординарной сети Петри требуется, чтобы каждое его входное место содержало хотя бы одну фишку, а после срабатывания перехода каждое его выходное место получает дополнительно по одной фишке. Наличие в сетях Петри дуг с кратностью, большей единицы, позволяет естественно моделировать реальные дискретные системы, но во многих случаях теоретического анализа сетей удобней ограничиться рассмотрением ординарных сетей.

Кононюк А.Е. Графы

Возникает вопрос, насколько переносимы результаты, полученные для ординарных сетей, на общий случай сетей. Как показал, что подкласс ординарных сетей не является существенным сужением класса сетей Петри и по отношению к большинству своих сетей оба класса оказываются эквивалентными в том смысле, что для сети Петри с заданным набором свойств можно построить ординарную сеть, обладающую тем же набором свойств.

Предложенное Хаком преобразование произвольной сети Петри $N = (P, T, F, W, M_0)$ в ординарную сеть $N' = (P', T', F', M'_0)$ состоит в следующем.

1) Для каждого места $p \in P$ определяется максимальная кратность $n(p)$ дуг, инцидентных этому месту, по формуле

$$n(p) = \max_{t \in T} (F(p, t) + F(t, p)).$$

Так, для места p на рис. 1, а $n(p) = 3$.

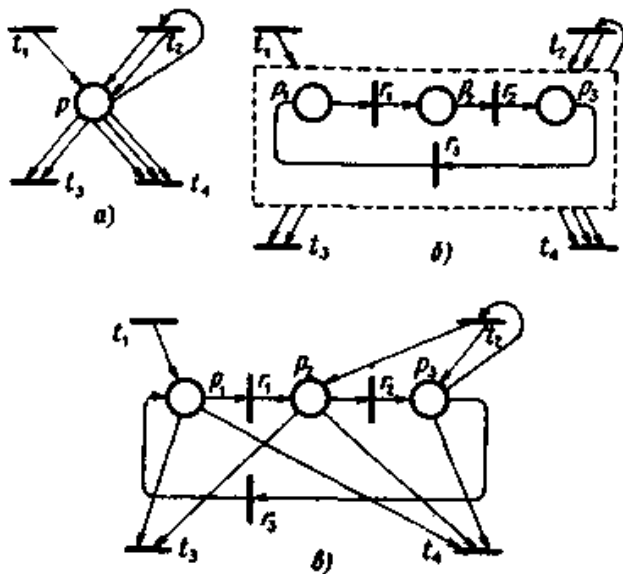


Рис.1

2) Каждому месту $p \in P$ будет соответствовать в сети N' множество $P'(p)$ из $n(p)$ мест $p^1, p^2, \dots, p^{n(p)}$, где $n(p)$ — определенная выше максимальная кратность дуг для места p . Таким образом, общее число мест в P' равно сумме максимальных кратностей для всех мест из P , т.е.

Кононюк А.Е. Графы

$$P' = \bigcup_{p \in P} P'(p).$$

3) Каждому переходу $t \in T$ соответствует в T' единственный переход, обозначаемый тем же символом tt , но в сети N' появляется также множество $T'(p) = \{t_1, t_2, \dots, t_n(p)\}$ новых переходов, которые связывают места $p^1, p^2, \dots, p^{n(p)}$ из множества $P'(p)$ в кольцевую сеть, как показано на рис. 1, б. При этом, если $n(p) = 1$, то новые переходы не вводятся. Таким образом,

$$T' = T \cup \left\{ \bigcup_{p \in P} T'(p) \right\}.$$

4) Для каждой дуги сети N , связывающей место p с некоторым переходом t и имеющей кратность $W(p, t)$, заводятся $W(p, t)$ дуг, связывающих t с местами $p^1, p^2, \dots, p^{n(p)}$. При этом распределение дуг в сети N' по местам $p^1, p^2, \dots, p^{n(p)}$ произвольно, лишь бы не возникали ситуации, когда переход и место связаны более чем одной дугой. На рис. 1, в показано распределение дуг в простой сети, построенной по сети на рис. 1, а. Начальная разметка $M'_0(p)$ места $p' \in P'(p)$ в сети N' определяется следующим образом:

$$M'_0(p^1) = M_0(p), \quad M'_0(p^i) = 0 \quad \text{для } i > 1$$

Анализ способа, которым конструируется ординарная сеть N' по сети N , позволяет установить связь между свободными языками $L(N)$ и $L(N')$ обеих сетей и множествами достижимых разметок $R(N)$ и $R(N')$. В кольцевой подсети ординарной сети N' , образованной множеством мест $P'(p)$ и множеством переходов $T'(p)$, фишки могут беспрепятственно перемещаться по местам и распределяться в них произвольным образом, если срабатывают только переходы из $T'(p)$, причем сумма фишек в местах из множества $P'(p)$ остается постоянной. Если в сети N переход $t \in T$ может сработать при разметке $M \in R(N)$, то переход $t \in T \cap T'$ может сработать в сети N' при разметке $M' \in R(N')$ такой, что

$$\forall p \in P: \sum_{p' \in P'(p)} M'(p') = M(p).$$

Таким образом, можно утверждать, что если τ' — последовательность срабатываний из $L(N')$, то ее проекция на T является последовательностью срабатываний из $L(N)$, а для любой достижимой в сети N разметки $M' \in R(N')$ существует разметка $M \in R(N)$ такая, что

Кононюк А.Е. Графы

$$\forall p \in P: M(p) = \sum_{p' \in P'(p)} M'(p').$$

Теорема 1. *Описанное преобразование сетей Петри в ординарные сети Петри сохраняет основные свойства исходных сетей — живость, ограниченность, консервативность.*

Доказательство. Поскольку язык $L(N)$ состоит из последовательностей срабатываний, являющихся проекциями последовательностей срабатываний из $L(N')$ на T , то в ординарной сети N' любой переход t , принадлежащий $T \cap T'$, является живым тогда и только тогда, когда t — живой в сети N . Если все переходы из $T \cap T'$ живы в N' , то и все "новые" переходы в N' , принадлежащие множеству $\bigcup_{p \in P} T'(p)$, живы в силу устройства кольцевых подсетей,

образованных местами из $P'(p)$ и переходами из $T'(p)$. Таким образом, ординарная сеть Петри N' жива тогда и только тогда, когда жива исходная сеть Петри N .

Тот факт, что сеть N' ограничена (консервативна) тогда и только тогда, когда исходная сеть N ограничена (консервативна), непосредственно следует из установленной выше связи между достижимыми разметками в множествах $R(N)$ и $R(N')$.

Если исходная сеть Петри помечена символами из некоторого алфавита A , то описанное преобразование позволяет получить помеченную ординарную сеть Петри, порождающую те же префиксный и терминальный языки, что и исходная сеть. Для этого достаточно дополнить правила преобразования правилом конструирования помечающей функции Σ' для ординарной сети. Она выражается через помечающую функцию Σ исходной сети следующим образом:

$$\forall t \in T: \Sigma'(t) = \begin{cases} \Sigma(t), & \text{если } t \in T. \\ \lambda, & \text{если } t \in T' \setminus T. \end{cases}$$

Другими словами, "старые" переходы сохраняют в ординарной сети помечающие символы (в том числе символы λ , если исходная сеть содержит λ -переходы), а все дополнительные переходы становятся λ -переходами. Из соотношения между свободными языками исходной и построенной сетей и между помечающими функциями Σ и Σ' следует, что как префиксные, так и терминальные языки этих сетей совпадают. Это означает, что класс всех префиксных языков ординарных помеченных сетей с λ -переходами совпадает с классом \mathcal{L}^λ префиксных языков сетей Петри, а класс терминальных языков ординарных помеченных сетей с λ -переходами совпадает с классом \mathcal{L}^{λ_0} терминальных языков сетей Петри. Предлагаемое преобразование приводит в общем случае к тому, что помеченная сеть без λ -переходов

Кононюк А.Е. Графы

преобразуется в ординарную помеченную сеть, содержащую λ - переходы. Теорема 14 говорит о невозможности избавиться в общем случае от λ -переходов, и, следовательно, не ясно, можно ли помеченную сеть Петри без λ -переходов преобразовать в ординарную помеченную сеть без λ -переходов. Однако Хак предложил новое преобразование, результатом применения которого к сети без λ -переходов является ординарная помеченная сеть (в общем случае также с λ -переходами), которую можно преобразовать в эквивалентную сеть без λ -переходов. Таким образом, для любой помеченной сети Петри без λ -переходов существует эквивалентная ей помеченная ординарная сеть без λ -переходов, т.е. классы префиксных и терминальных языков ординарных сетей без λ -переходов совпадают соответственно с классами \mathcal{L} и \mathcal{L}_0 . Обсужденные факты позволяют сформулировать следующую теорему.

Теорема 2. *Ординарные сети Петри генерируют те же классы языков \mathcal{L} , \mathcal{L}_0 , \mathcal{L}^λ , \mathcal{L}_0^λ , что и сети Петри с кратными дугами.*

Если класс помеченных ординарных сетей равноможен классу помеченных сетей с кратными дугами по отношению к префиксным и терминальным языкам, то иная ситуация складывается со свободными языками.

Теорема 3. *Класс свободных языков сетей Петри строго включает класс свободных языков ординарных сетей Петри.*

Доказательство. Поскольку класс ординарных сетей является, по определению, подклассом класса сетей Петри, достаточно показать, что существует сеть с кратными дугами, которая порождает свободный язык, не порождаемый никакой ординарной сетью. Такая сеть показана на рис.2.

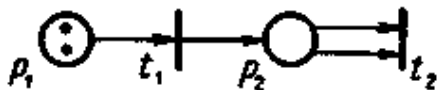


Рис. 2.

Она порождает свободный префиксный язык $\mathcal{L}(N) = \{t_1, t_1 t_1, t_1 t_1 t_2\}$, а для терминальной разметки $M_f = \{0,0\}$ она порождает терминальный язык $\mathcal{L}(N, M_f) = \{t_1 t_1 t_2\}$. Никакая ординарная сеть Петри не может порождать такие свободные языки. Предположим противное. Тогда сеть $N' = (P', T', F', M'_0)$, порождающая эти языки, должна иметь ровно два перехода — t_1 и t_2 .

1) Предположим, что $t_2 = \emptyset$ или $\forall p \in t_2: M'_0(p) > 0$. Поскольку сеть N — ординарная, то в префиксном языке $L(N) = L(N')$ должно содержаться слово, начинающееся символом перехода t_2 , но это не так.

2) Предположим, что $\exists p \in t_2: M'_0(p) = 0$. Обозначим через $P^0(t_2)$ множество всех входных мест перехода t_2 , имеющих нулевую разметку. Здесь возможны два подслучая.

2.1) Существует место p такое, что $p \in P^0(t_2) \wedge p \notin t_1$. В этом случае нулевую разметку места p не может изменить ни переход t_1 , ни переход t_2 , поэтому t_2 оказывается мертвым переходом, что противоречит вхождению слова $t_1 t_1 t_2$ в язык $L(N') = L(N)$.

2.2) $P^0(t_2) \subseteq t_1$, т.е. любое входное место перехода t_2 с нулевой разметкой является одновременно выходным местом перехода t_1 . В этом случае язык $L(N')$ должен содержать слово $t_1 t_2$, поскольку он содержит слово t_1 , а срабатывание t_1 увеличивает на единицу разметку всех мест из $P^0(t_2)$, после чего переход t_2 может сработать (так как сеть N' — ординарная).

Перечисленные случаи исчерпывают все возможные варианты сетевых связей между переходами t_1 и t_2 , но ни один из них не приемлем для порождения свободного языка $L(N') = L(N)$. Поэтому не существует ординарной сети Петри, порождающей такой язык.

2.7.2. Автоматные сети и синхронизационные графы

Два наиболее простых подкласса сетей Петри образуются за счет наложения строгих топологических ограничений на структуру сети, т.е. ограничений на отношения инцидентности F , связывающее места и переходы сети.

Сеть Петри с множеством переходов T называется *автоматной*, если $\forall t \in T: |t^+| = |t^-| = 1$, т.е. если каждый переход сети имеет ровно одно входное и ровно одно выходное место. Пример автоматной сети показан на рис.3.

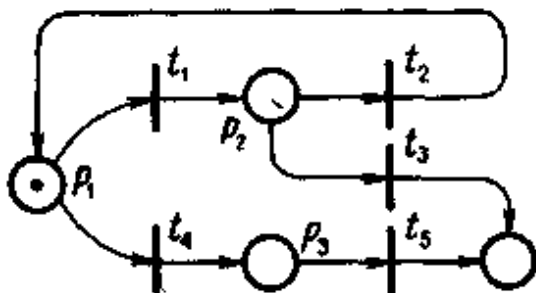


Рис.3.

Сеть Петри с множеством мест P называется *синхронизационным графом* (или *синхрографом*, или *маркированным графом*), если $|\cdot p| = |p \cdot| = 1$, т.е. если в каждое место сети входит ровно одна дуга и из каждого места исходит ровно одна дуга. Пример синхрографа показан на рис.4.

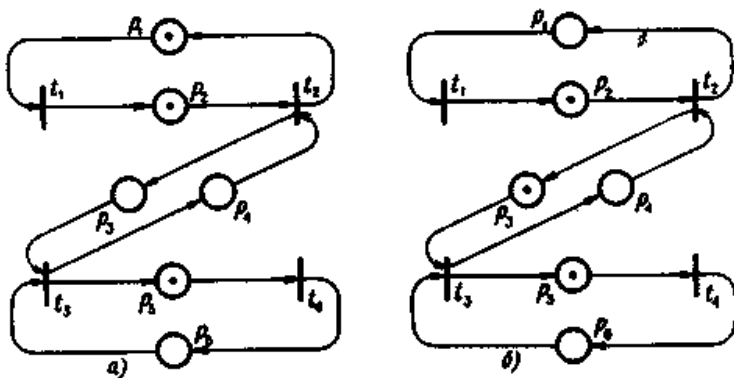


Рис. 4.

Оба подкласса являются действительно простыми в том смысле, что они способны моделировать только простые дискретные системы, а анализ их математических свойств несложен. Оба они являются подклассами класса ординарных сетей Петри.

Из определения автоматной сети следует, что граф сети связан (из любой вершины графа можно пройти в любую вершину по пути вдоль дуг и против дуг). При своем срабатывании любой переход t изымает ровно одну фишку из своего входного места p_1 и помещает ровно одну

Кононюк А.Е. Графы

фишку в свое выходное место p_2 . Поэтому, если переход t сработал при разметке M и изменил ее на разметку M' , то $M'(p) = M(p)$ для всех мест p , не совпадающих с p_1 и p_2 .

$$M'(p_1) + M'(p_2) = M(p_1) - 1 + M(p_2) + 1 = M(p_1) + M(p_2).$$

Таким образом,

$$\sum_{p \in P} M'(p) = \sum_{p \in P} M(p).$$

и число фишек в автоматной сети остается постоянным при ее работе, т.е. автоматная сеть консервативна и ограничена. Отсюда следует также, что автоматная сеть безопасна тогда и только тогда, когда ее начальная разметка содержит ровно одну фишку. Автоматная сеть жива тогда и только тогда, когда она представляет собой сильно связный граф, т.е. из любой вершины сети существует путь вдоль дуг в любую вершину, и ее начальная разметка содержит хотя бы одну фишку.

Так как автоматная сеть ограничена, то ее граф разметок конечен, и, следовательно, в классе автоматных сетей разрешимы проблемы достижимости разметки, проблемы живости, проблемы R -включения и R -эквивалентности, проблема эквивалентности по языкам и все другие проблемы анализа свойств сетей. Все это следствие того, что автоматные сети представляют собой, по существу, сетевую форму задания конечных автоматов. Это непосредственно следует из конечности графа разметок любой автоматной сети. Этот граф представляет собой не что иное, как граф конечного автомата, в котором множество состояний образовано множеством достижимых в сети разметок, а алфавит — символами переходов сети. Поэтому на автоматные сети распространяются все результаты теории конечных автоматов, и изучать этот класс в рамках теории сетей Петри не имеет смысла.

В ряде случаев при определении автоматной сети накладывается дополнительное ограничение, чтобы ее начальная разметка имела ровно одну фишку. Такие сети моделируют последовательные дискретные системы, в которых невозможны никакие параллельные события (исходное определение автоматной сети допускает моделирование так называемого конвейерного параллелизма в дискретных системах, когда несколько фишек независимо перемещаются в сети). В то же время автоматные сети позволяют изображать конфликтные ситуации, когда одно и то же место-условие является входным (или выходным) для нескольких переходов-событий.

В отличие от автоматных сетей синхрографы могут описывать параллелизм событий, но не допускают изображения конфликтных

Кононюк А.Е. Графы

ситуаций. Основные свойства сетей, и в том числе живость и достижимость разметки, распознаваемы в классе маркированных графов, причем на основе простого анализа структуры графа и его начальной разметки. Структурные компоненты, в терминах которых формулируются условия ограниченности, живости и пр., — это простой путь и цикл в сети. *Простой путь* в сети (P, T, F) представляет собой последовательность (x_1, x_2, \dots, x_n) элементов сети такую, что $x_i Fx_{i+1}$ для $1 \leq i \leq n - 1$ и $x_i \neq x_j$ для любых двух элементов сети, кроме, быть может, $i = 1$ и $j = n$. Простой путь называется (простым) *циклом*, если $x_1 = x_n$. В синхрографе на рис.4,*a* простыми циклами являются пути

$(t_1, p_2, t_2, p_1, t_1), (t_2, p_3, t_3, p_4, t_2), (t_3, p_5, t_4, p_6, t_3).$

Будем говорить, что простой путь (в том числе цикл) содержит n фишек при разметке M , если n - сумма фишек в проекции $M(P')$ разметки M на множество мест $P' \subseteq P$, входящих в этот путь.

Теорема 4. *При функционировании синхрографа число фишек, содержащихся в любом его цикле, остается постоянным.*

Доказательство. Пусть (x_1, x_2, \dots, x_n) - цикл ($x_1 = x_n$) в синхрографе N и пусть, для определенности, $x_1 = x_n$ — переход t . Тогда этот цикл можно записать как последовательность $(t_1, p_1, t_2, \dots, p_m, t_1)$.

Если $\sum_{i \neq t}^m M(p_i)$ — число фишек в местах p_1, \dots, p_m цикла при некоторой разметке $M \in R(N)$, то при смене разметки M на M' такую, что $M[t] > M'$, возможны два случая: t принадлежит этому циклу или t не принадлежит ему. В первом случае разметка мест p_1, \dots, p_m не изменяется при смене M на M' , так как эти места инцидентны только переходами цикла и не могут получить или потерять фишки при срабатывании переходов, не входящих в цикл. Во втором случае срабатывание любого из переходов цикла приведет к перемещению ровно одной фишки из его входного в его выходное место. Оба эти места принадлежат циклу, и поэтому

$$\sum_{i=1}^m M'(p_i) = \sum_{i=1}^m M(p_i).$$

На рис. 4,*a* цикл $(t_1, p_2, t_2, p_1, t_1)$ содержит две фишки при любой достижимой разметке, цикл $(t_2, p_3, t_3, p_4, t_2)$ - ни одной фишки, цикл $(t_3, p_5, t_4, p_6, t_3)$ - одну фишку. Простой путь (в том числе цикл) будем называть *пустым* {*непустым*) при разметке M , если он не содержит ни одной фишки (если он содержит хотя бы одну фишку) при этой разметке. Цикл $(t_2, p_3, t_3, p_4, t_2)$ - пустой в синхрографе на рис. 4,*a*.

Теорема 5. *Переход t — потенциально живой в синхрографе, если и только если он не входит ни в один пустой при M_o цикл и ни в один пустой при M_o путь из пустого цикла.*

Доказательство. В силу теоремы 4, если переход t принадлежит пустому при M_o циклу, то он никогда не сработает, так как его входное место, входящее в пустой цикл, никогда не получит фишки. Путь из цикла в синхрографе может начинаться только переходом. Поскольку в пустом цикле этот переход мертв, то и все остальные переходы пустого пути, начинающегося в пустом цикле, мертвы.

Для любого перехода в синхрографе можно выписать следующее множество всех простых путей (в том числе циклов):

- (1) каждый путь или заканчивается этим переходом или является циклом, содержащим этот переход,
- (2) ни один из путей не является подпутем другого пути из этого множества,
- (3) каждый путь, если это не цикл, начинается некоторым переходом, который или не имеет входных мест или принадлежит некоторому циклу.

Если переход потенциально живой, то все переходы из построенного множества путей также потенциально живы. Это имеет место только тогда, когда каждый путь из этого множества обладает одним из следующих свойств:

- 1) он начинается переходом без входных мест (этот переход всегда может сработать),
- 2) он начинается в цикле, содержащем хотя бы одну фишку при M_o ,
- 3) он является циклом, содержащим хотя бы одну фишку при M_o .

Поэтому, если переход является потенциально живым, то он не может принадлежать пустому при M_o циклу или пустому при M_o пути из такого цикла.

В синхрографе на рис. 4,а потенциально живыми являются переходы t_1, t_4 , мертвыми — переходы t_2, t_3 .

Теорема 6. *Синхрограф является живым, если и только если любой цикл в нем не пуст при начальной разметке M_o .*

Доказательство. При выполнении условия теоремы любой переход в синхрографе является потенциально живым, так как для него выполнены условия теоремы 5. При функционировании синхрографа ни один из циклов не может стать пустым, поэтому условия теоремы 5 выполнены для любой достижимой разметки и, следовательно, любой из переходов сети является живым, вся сеть - жива.

Если синхрограф является живой сетью, то он не может содержать пустых при M_o циклов, так как в этом случае переходы пустого цикла будут мертвыми.

Кононюк А.Е. Графы

Синхрограф на рис. 4,а не является живым, так как он содержит пустой цикл $(t_2, p_3, t_3, p_4, t_2)$. В синхрографе на рис. 4,б, отличающемся от предыдущего начальной разметкой, все циклы непусты, поэтому он живой.

В то время как любая автоматная сеть ограничена, синхронизационная сеть может содержать неограниченные места, как, например, сеть на рис. 5, в которой таким местом является место p_2 .

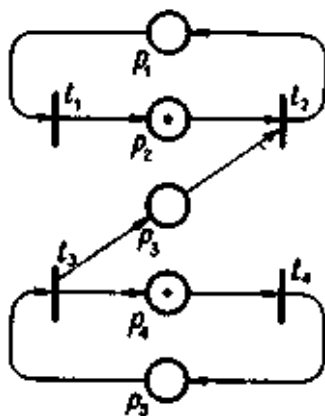


Рис. 5.

Распознавание ограниченности и безопасности синхрографов может быть выполнено так же, как и в общем случае сетей Петри, но для живых синхрографов проверку на ограниченность и безопасность можно сделать на основе анализа их структурных свойств и начальной разметки.

Теорема 7. *Живой синхрограф является ограниченным, если и только если каждое его место входит в некоторый цикл.*

Доказательство. Если место p входит в цикл синхрографа, то максимальное число фишек, которое может в нем накопиться, не превышает общего числа фишек в этом цикле, а оно постоянно (теорема 4), следовательно, место p ограничено. Если место p не входит ни в какой цикл, то оно принадлежит некоторому пути, который начинается или переходом без входных мест или переходом в некотором цикле. В обоих случаях переход $t = \cdot p$ и переход $t' = p \cdot$ живы, так как жива сеть. При этом число срабатываний перехода t не зависит от числа срабатываний перехода t' , так как они не входят в некоторый цикл. Поэтому возможна такая последовательность срабатываний, в

Кононюк А.Е. Графы

которой t срабатывает сколь угодно большее число раз, чем t' , и, следовательно, местор неограничено.

В живом синхрографе на рис. 5 место p_3 не ограничено, так как оно не входит ни в один цикл. В живом синхрографе на рис. 4,б все места ограничены и он ограничен, так как выполнено условие теоремы 7.

Теорема 8. *Живой синхрограф безопасен, если и только если каждое его место входит в некоторый цикл, содержащий ровно одну фишку.*

Доказательство. Следствие из теоремы 7.

Проблема достижимости разметки разрешима в классе синхрографов, так как она эквивалентна проблеме живости, а последняя разрешима в этом классе (теорема 6).

2.7.3. Свободные сети

Свободная сеть (или сеть со свободным выбором) — это сеть Петри (P, T, F, M_0) такая, что $\forall (p, t) \in F: p' = \{t\} \vee t' = \{p\}$, т.е. любая дуга, ведущая от места к переходу, или начинается местом, из которого не выходит ни одной другой дуги, или заканчивается переходом, в который не ведет никакая другая дуга.

Пример свободной сети показан на рис. 6,в.

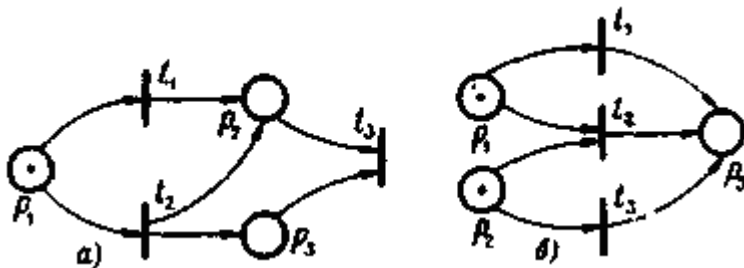


Рис. 6

В свободной сети каждое конфликтное место (место, из которого выходит более чем одна дуга) является входным для такого множества переходов, в котором при любой разметке любой (но один) из переходов может сработать или все переходы не могут сработать. Например, в сети на рис. 6,б множество переходов $\{t_1, t_2\}$, которые имеют в качестве входного места конфликтное место p_1 , не удовлетворяет таким условиям, так как после срабатывания перехода t_3 переход t_2 не может сработать. Эта сеть не является свободной.

Класс свободных сетей, как видно из его определения, строго включает классы автоматных сетей и синхронизационных графов, он является

Кононюк А.Е. Графы

подклассом ординарных сетей Петри. Для этого класса Коммонером и Хаком были найдены необходимые и достаточные условия живости и безопасности. Они сформулированы с использованием следующих двух специальных типов подмножеств множества мест сети.

Ловушкой называется непустое подмножество $Q = \{q_1, \dots, q_n\}$ множества мест P такое, что $Q \subseteq \cdot Q$, где $\cdot Q = \bigcup_{i=1}^n \cdot q_i$, а ${}^{\circ}Q = \bigcup_{i=1}^n {}^{\circ}q_i$.

Другими словами, любой переход, входное место которого принадлежит Q , содержит в Q и некоторое свое выходное место. Когда такой переход срабатывает и забирает фишку из ловушки, то он немедленно возвращает фишку в ловушку. Таким образом, если Ловушка *размечена*, т.е. содержит хотя бы одну фишку, то при функционировании сети она всегда остается размеченной, так как любая фишка, попавшая в ловушку, не может исчезнуть из нее.

Тупиком называется непустое подмножество $R = \{r_1, \dots, r_n\}$ множества

мест P такое, что ${}^{\circ}R \subseteq R$. Другими словами, любой переход, выходное место которого принадлежит R , содержит в R и свое входное место. Для того чтобы переход, содержащий свое выходное место в тупике, сработал и поместил фишку в тупик, нужно, чтобы он забрал по крайней мере одну фишку из тупика. Таким образом, если тупик *пуст*, т.е. не содержит ни одной фишки, то этот переход мертв. Пустой тупик остается пустым при функционировании сети, так как разметка мест в тупике не может возрасть.

Заметим, что ловушка и тупик не исключают друг друга. В сильно связной сети множество P всех ее мест обладает свойством $\cdot R = R$, т.е. является одновременно ловушкой и тупиком.

В сети на рис. 7 множество мест $Q = \{p_1, p_2\}$ образует ловушку, так как $\cdot Q = \{t_2, t_3\}$, ${}^{\circ}Q = \{t_1, t_2, t_3\}$ и, следовательно, $\cdot Q \subseteq {}^{\circ}Q$.

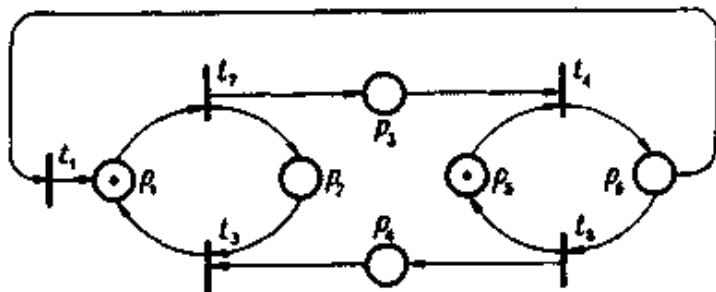


Рис. 7

Кононюк А.Е. Графы

Множество мест $R = \{p_5, p_6\}$ образует тупик, так как $R = \{t_4, t_5\}$, $R' = \{t_4, t_5, t_1\}$ и $R \subset R'$.

Критерий живости свободной сети основан на следующих рассуждениях. Как отмечено выше, переход, содержащий выходное место в тупике, не сможет сработать, если тупик пуст. Таким образом, сеть, при функционировании которой некоторый тупик может оказаться пустым, не будет живой. Чтобы удержать фишки в тупике нужно, чтобы тупик содержал размеченную ловушку. Прежде чем оформить это рассуждение в теорему, докажем две вспомогательные леммы.

Для сети с множеством мест P и для разметки M обозначим через $P^0(M)$ множество мест $\{p \in P \mid M(p) = 0\}$, т.е. множество мест без фишек при разметке M , а через $P^+(M)$ — множество $\{p \in P \mid M(p) > 0\}$, т.е. множество мест, содержащих фишки при M .

Лемма 1. Для произвольного подмножества переходов S в сети Петри и для произвольной разметки M , если $(S \cap P^0(M)) \subset S$, то или в S существует переход, могущий сработать при M , или существует пустой при M тупик R такой, что $S \subseteq R'$.

Доказательство. Предположим, что выполнено условие леммы и в S нет перехода, могущего сработать при M , т.е. $\forall t \in S: t \notin P^+(M)$, или что то же самое, $\forall t \in S: t \cap P^0(M) \neq \emptyset$ или $\forall t \in S: t \in (t \cap P^0(M))$.

Отсюда следует, что $S \subseteq (S \cap P^0(M))$. По условию леммы $(S \cap P^0(M)) \subset S$, поэтому $(S \cap P^0(M))$ - пустой тупик.

Лемма 2. Пусть в свободной сети Петри (P, T, F, M_0) достижима разметка M , а $S \subseteq T$ - множество переходов, мертвых при M . Тогда существует достижимая от M разметка M' и пустой при M' тупик R такой, что $S \subseteq R'$.

Доказательство леммы проводится индукцией по мощности множества S . Если S содержит все переходы сети, то непосредственно применима лемма 1. Это базис индукции. Каждый шаг индукции связан с уменьшением числа переходов в S , а при одном переходе в S лемма формулирует достаточные условия живости этого перехода.

Предположим, что S совпадает с T , т.е. $|T \setminus S| = 0$.

Так как T - множество всех переходов сети, то включение $(S \cap P^0(M)) \subset S$ тривиально истинно. В силу леммы 1, если ни один из переходов множества S не может сработать, то существует пустой при M тупик R такой, что

Кононюк А.Е. Графы

$R \subseteq P^0(M)$ и $S \subseteq R^*$.

Пусть теперь $S \subset T$, т.е. $|T \setminus S| > 0$.

Построим последовательность достижимых от M разметок

$M_0, M_1, \dots, M_i, \dots, M'$ такую, что $M = M_0$ и при M' имеется пустой тупик R

такой, что $R \subseteq P^0(M)$ и $S \subseteq R^*$.

Покажем, что все переходы множества $(^*S)$ мертвы. Предположим, что

в S есть переход t , для которого существует входное место p в *t , из

которого ведет дуга на другой переход t' , принадлежащий $(^*t)$, и переход t' — потенциально живой. По условию леммы переход t

мертвый, поэтому $t' \in p \setminus S$. Но тогда место p является входным для нескольких переходов, включая t и t' . Так как сеть свободна, то переходы t и t' имеют по одной входной дуге, и если может сработать t' , то и t также может сработать, что противоречит условию.

Пусть $M = M_i$ - текущая разметка при функционировании сети. Возможны два случая.

1) $(^*S \cap P^0(M_i)) \subseteq S$. В этом случае применима лемма 1. Так как все переходы из S мертвы, то должен существовать пустой при

M_i тупик $R = (^*S \cap P^0(M_i))$ такой, что $S \subseteq R^*$. Этот случай доказывает лемму при $M' = M_i$.

2) $(^*S \cap P^0(M_i)) \not\subseteq S$. Тогда существует переход $t \in ((^*S \cap P^0(M_i)) \setminus S)$, для которого возможны два подслучая.

2.1) t - мертвый переход. Это означает, что все переходы множества $S' = S \cup \{t\}$ мертвы. Множество $T \setminus S'$ содержит на один элемент меньше, чем множество $T \setminus S$. По индуктивному предположению, должна существовать последовательность срабатываний τ , ведущая от разметки M_i к некоторой разметке M' такой, что для нее существует пустой при M' тупик $R \subseteq P^0(M')$ и $S \subseteq R^*$. Так как $S \subseteq S'$, то лемма доказана для разметки M' и тупика R .

2.2) t - потенциально живой переход. Пусть M_{i+1} - разметка, при которой может сработать t , и $M_i[\tau > M_{i+1}$. Поскольку ни один из переходов из множества $(^*S)$ не может появиться в последовательности τ , то

$$(^*S \cap P^+(M_i)) \subseteq (^*S \cap P^+(M_{i+1})).$$

Так как t срабатывает и посылает фишки в $(^*S \cap P^0(M_i))$, то $|^*S \cap P^0(M_{i+1})| < |^*S \cap P^0(M_i)|$.

Кононюк А.Е. Графы

Пусть теперь $M_i + 1$ - текущая разметка. Повторяя для нее те же рассуждения, что и для разметки M_i , мы или приходим к подслучаю 2.2, при котором мощность множества $(S \cap P^0(M_i))$ уменьшается, или доказываем лемму, приходя к первому случаю или подслучаю 2.1.

Теорема 9. *Если в свободной сети каждый тупик содержит размеченную ловушку, то сеть жива.*

Доказательство. Из леммы 2 следует, что если никакой из тупиков свободной сети никогда не сможет стать пустым ни при какой достижимой разметке, то никакой переход t сети не может стать мертвым ни при какой разметке. Действительно, в противном случае достаточно взять $S = \{t\}$ и t -тупиковую разметку M , применить лемму 2 и прийти к существованию пустого тупика. Таким образом, если все тупики сети не пусты ни при одной достижимой разметке, то все переходы сети живы. В свою очередь, если тупик, содержит размеченную ловушку, то он никогда не сможет стать пустым. \square

Теорема 9 устанавливает достаточные условия живости в классе свободных сетей Петри; показано, что если свободная сеть жива, то каждый ее тупик должен содержать размеченную ловушку, в противном случае среди достижимых разметок сети существует t -тупиковая разметка для некоторого перехода t сети. Таким образом, условие теоремы 9 является на самом деле необходимым и достаточным, но мы опускаем здесь доказательство необходимости.

Проиллюстрируем применимость коитерия живости свободных сетей к примерам сетей, рассмотренных в этой главе. В автоматной сети на рис. 3 (автоматная сеть — частный случай свободной) имеются следующие тупики (исключая тривиальный - множество всех мест):

$R_1 = \{p_1, p_2, p_3\}$, $R_2 = \{p_1, p_2\}$, $R_3 = \{p_1, p_3\}$, и следующие нетривиальные ловушки: $Q_1 = \{p_1, p_2, p_4\}$, $Q_2 = \{p_3, p_4\}$. Как видно, ни один из этих тупиков не содержит ловушки, поэтому сеть на рис. 3 не жива. В синхронизационных графах на рис. 4, отличающихся лишь начальной разметкой, нетривиальные тупики:

$$\begin{aligned} R_1 &= \{p_1, p_2, p_3, p_4, p_5\}, & R_2 &= \{p_1, p_2, p_3, p_4\}, & R_3 &= \{p_3, p_4, p_5, p_6\}, \\ R_4 &= \{p_4, p_5, p_6\}, & R_5 &= \{p_1, p_2, p_3\}, & R_6 &= \{p_1, p_2\}, \\ R_7 &= \{p_3, p_4\}, & R_8 &= \{p_5, p_6\}. \end{aligned}$$

нетривиальные ловушки:

$$\begin{aligned} Q_1 &= \{p_1, p_2, p_3, p_4, p_6\}, & Q_2 &= \{p_1, p_2, p_3, p_4\}, & Q_3 &= \{p_3, p_4, p_5, p_6\}, \\ Q_4 &= \{p_3, p_5, p_6\}, & Q_5 &= \{p_1, p_2, p_3\}, & Q_6 &= \{p_1, p_2\}, \\ Q_7 &= \{p_3, p_4\}, & Q_8 &= \{p_5, p_6\}. \end{aligned}$$

Кононюк А.Е. Графы

Любой тупик в обеих сетях содержит ловушку. В сети на рис. 4,б все ловушки размечены, поэтому эта сеть - живая. В сети на рис.4,а тупик $R_7 = \{p_3, p_4\}$ содержит ловушку $Q_7 = \{p_3, p_4\}$, но эта ловушка не содержит фишек, поэтому сеть на рис. 4,в не является живой. В синхрографе на рис. 5 каждый из тупиков $\{p_1, p_2\}$, $\{p_4, p_5\}$, $\{p_3, p_4, p_5\}$ содержит размеченную ловушку $\{p_1, p_2\}$ или $\{p_4, p_5\}$. В свободной сети на рис. 7 нетривиальные тупики:

$$R_1 = \{p_1, p_2, p_4, p_5, p_6\}, R_2 = \{p_1, p_2, p_3, p_5, p_6\}, \\ R_3 = \{p_1, p_2, p_5, p_6\}, R_4 = \{p_4, p_5, p_6\}, R_5 = \{p_5, p_6\}.$$

нетривиальные ловушки:

$$Q_1 = \{p_1, p_2, p_4, p_5, p_6\}, Q_2 = \{p_1, p_2, p_3, p_5, p_6\}, \\ Q_3 = \{p_1, p_2, p_5, p_6\}, Q_4 = \{p_1, p_2, p_4\}, Q_5 = \{p_1, p_5\}.$$

Все ловушки размечены, но тупики R_4 и R_5 не содержат ни одну из ловушек сети, поэтому сеть на рис. 7 не является живой.

Действительно, последовательность срабатываний $t_2 t_4 t_1 t_2$ приводит к тупиковой разметке $(0, 2, 1, 0, 0, 0)$.

Заметим, что условия живости для автоматных сетей и синхрографов могут быть получены из теоремы 10 как следствия.

Условия безопасности для свободных сетей Петри будем выводить только для живых сетей. Для живых свободных сетей Петри имеются необходимые и достаточные условия их безопасности, формулируемые без привлечения графов разметок.

В сети Петри (P, T, F, M_0) замкнутой подсетью, задаваемой подмножеством мест $Q \subseteq P$, называется сеть (Q, T', F', M'_0) , в которой

$T' = Q \cap Q'$, $F' = F \cap (Q \times T \cup T \times Q)$, $M'_0 = M_0(Q)$. Другими словами, замкнутая подсеть, задаваемая подмножеством мест Q , включает места из Q , все переходы, инцидентные Q , и все дуги, связывающие Q и T' , а начальная разметка подсети является проекцией начальной разметки сети на места из Q . Например, в сети на рис. 8,а замкнутая подсеть, задаваемая множеством мест $\{p_1, p_4\}$, выглядит так, как показано на рис. 8,б, а замкнутая подсеть, задаваемая множеством мест $\{p_2, p_3\}$, показана на рис. 8,в.

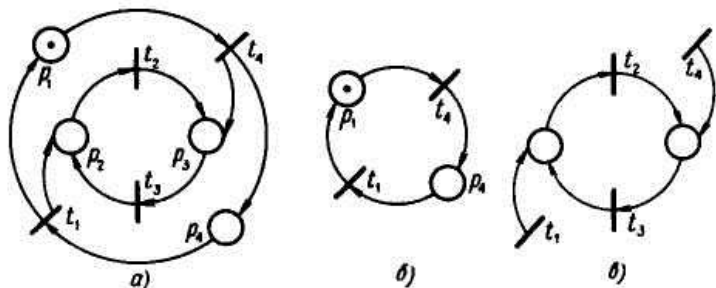


Рис. 8

Будем говорить, что сеть Петри покрыта заданной совокупностью замкнутых подсетей, если каждое место сети входит в одну из подсетей этой совокупности. Соответствующая совокупность подсетей покрывает сеть. Например, подсети на рис. 4,б и в покрывают сеть на рис. 4,а.

Критерий безопасности живой свободной сети связан с покрытием ее замкнутыми подсетями, которые представляют собой сильно связанные автоматные сети, каждая из которых содержит лишь одну фишку. Заметим, что такая подсеть сама является живой и безопасной сетью. Поэтому можно сказать, что свободная сеть жива и безопасна, если существует покрытие ее живыми и безопасными подсетями. Вновь мы остановимся только на доказательстве достаточности условий живости и безопасности, хотя была доказана и их необходимость.

Лемма 3. Пусть сеть жива и безопасна при начальной разметке M_0 . Если изменить начальную разметку, убрав из непустого места фишку, то полученная сеть не будет живой.

Доказательство. Если бы это было не так, то вернув в полученную живую сеть одну фишку в то же самое место, мы получили бы, как легко видеть, небезопасную сеть.

Лемма 4. В живой небезопасной свободной сети существует неограниченное место.

Доказательство. Из теоремы 9 следует, что если изменить начальную разметку сети, удовлетворяющей условию данной теоремы, удалив из сети все фишки, кроме фишек в ловушках, то полученная сеть останется живой.

Пусть сеть работает до тех пор, пока одно из мест не получит две фишки (если при M_0 не было таких мест). В каждом месте, в котором больше одной фишки, пометим каким-либо образом (например, закрасим в красный цвет) все фишки, кроме одной. Пусть сеть продолжит работу так, чтобы из мест убирались только не отмеченные фишки, а отмеченные не двигались. При этом каждый раз, когда в

Кононюк А.Е. Графы

некотором месте появляется больше одной фишки, повторяем процедуру отметки и задержки отмеченных фишек. На каждом таком шаге число отмеченных фишек увеличивается. Задержание отмеченных фишек не препятствует срабатыванию переходов, так как в живой сети любая достижимая разметка не является t -тупиковой ни для какого перехода t и для любого перехода каждое его входное место, имеющее ненулевую разметку, содержит хотя бы одну неотмеченную фишку. Описываемый процесс функционирования сети с задержкой фишек может продолжаться бесконечно, а так как число отмеченных фишек растет, то найдется место, в котором оно будет расти неограниченно.

Теорема 11. *Если живая свободная сеть покрывается совокупностью сильно связанных автоматных плотных подсетей, то она безопасна.*

Доказательство. Число фишек в любой из покрывающих автоматных подсетей постоянно. Общее число фишек в свободной сети ограничено сверху суммой фишек во всех покрывающих плотных подсетях. Тогда, по лемме 4, если сеть жива, то она должна быть безопасной, иначе в ней есть неограниченное место.

Свободная сеть на рис. 8,а содержит только один тупик $\{p_1, p_2, p_3, p_4\}$, т.е. все множество мест сети. Тупик содержит размеченную ловушку, совпадающую с самим туликом, поэтому эта сеть - живая. Сеть на рис. 8,а покрывается разными совокупностями плотных подсетей, например, автоматными подсетями, показанными на рис. 8,б и в. Однако только подсеть на рис. 8,б является сильно связной сетью. Поэтому сеть на рис. 8,а не является безопасной. В этом легко убедиться, выполнив, например, последовательность срабатываний $t_4 t_1 t_4$ которая приведет к разметке $(0, 1, 2, 1)$.

2.8. Расширенные и ограниченные модели сетей Петри

Ранее мы показали, что сети Петри могут быть использованы для моделирования самых различных систем: аппаратного и программного обеспечения вычислительных машин, химических, социальных систем и т. д. Однако это разнообразие примеров лишь показывает, что сети Петри могут адекватно моделировать некоторые системы, но могут существовать системы, которые нельзя должным образом моделировать сетями Петри, т. е. мощность моделирования сетей Петри имеет пределы.

Кроме того, показано, что не все вопросы анализа сетей Петри разрешимы. Так, неразрешимы задачи эквивалентности и включения

Кононюк А.Е. Графы

для множеств достижимости сетей Петри и языков сетей Петри, хотя эти задачи могут оказаться очень важными для получения оптимальных сетей Петри. Даже те вопросы анализа, которые разрешимы, очень трудны, имеется в виду то, что они требуют большого объема вычислений.

В этом разделе мы исследуем предложения, которые были сделаны для преодоления двух ограничений сетей Петри: ограничений на *мощность моделирования* сетей Петри и ограничений на *мощность разрешения* сетей Петри. Во-первых, рассмотрим некоторые предложения по расширению модели сети Петри. Расширение модели сети Петри должно увеличивать мощность моделирования сетей Петри, но при этом оно может уменьшать их мощность разрешения. Влияние любого расширения на мощность разрешения расширенной модели требует тщательного изучения.

Отметив, что расширение модели сети Петри может приводить к уменьшению мощности разрешения, мы рассмотрим также, как мощность разрешения может быть увеличена посредством ограничения модели сети Петри. Предложены различные подклассы моделей сети Петри, которые обусловлены ограничениями на структуру сети Петри. Необходимо исследовать влияние этих ограничений как на мощность моделирования, так и на мощность разрешения.

Такое исследование покажет, как взаимосвязаны мощность моделирования и мощность разрешения, а также укажет границы обеих для модели сети Петри.

2.8.1. Границы возможностей моделирования с помощью сетей Петри

Исследователи, использовавшие сети Петри для моделирования систем, обнаружили, что возможности моделирования сетями Петри реальных систем ограничены. Этим объясняется появление тенденции к расширению модели. Имеется несколько типов расширений. Патил предложил расширить сети Петри путем введения понятия *области ограничения*. Область ограничения — это множество позиций. Правило запуска модифицируется таким образом, что переход может быть запущен тогда и только тогда, когда в результирующей маркировке не все позиции, входящие в область ограничения, одновременно имеют фишки (не пусты). Например, если $\{p_1, p_4\}$ есть область ограничения, то в любой момент времени либо p_1 либо p_4 должны быть пусты. Если p_1 не пуста, то фишка не может быть

Кононюк А.Е. Графы

помещена в p_4 до тех пор, пока все фишки из p_1 не будут удалены, и наоборот.

Ное в своей модели операционной системы CDC 6400 ввел другое расширение: переход *исключающее ИЛИ* (рис. 1).

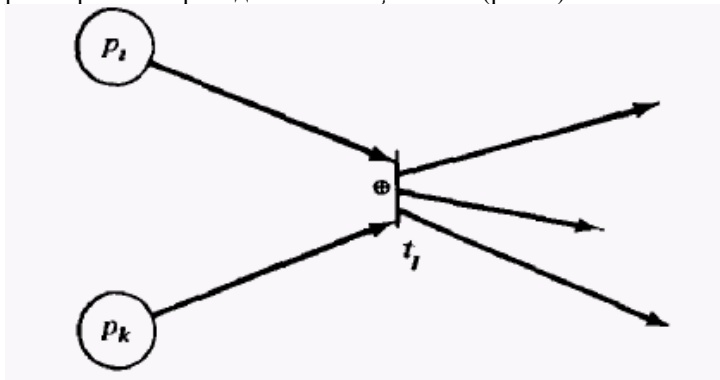


Рис. 1. Переход исключающее ИЛИ. Переход t_j может быть запущен, если одна из позиций p_i и p_k имеет фишку.

В обычных сетях Петри переход запускается, когда все его входы имеют фишки. Такое правило называется *логикой И*, поскольку мы должны иметь фишки i в первом входе, i во втором входе, i в третьем входе и т. д. Переход исключающее ИЛИ может запускаться тогда и только тогда, когда точно один из его входов имеет фишки, а все другие фишек не имеют. Следовательно, разрешающее правило состоит в том, чтобы имел фишку *или* первый, *или* второй вход (но не оба одновременно). Когда переход запускается, он удаляет фишку только из входа с фишками.

Подобное расширение было использовано Баером в его модели компилятора. Баер ввел понятие *переключателя* (рис. 2).

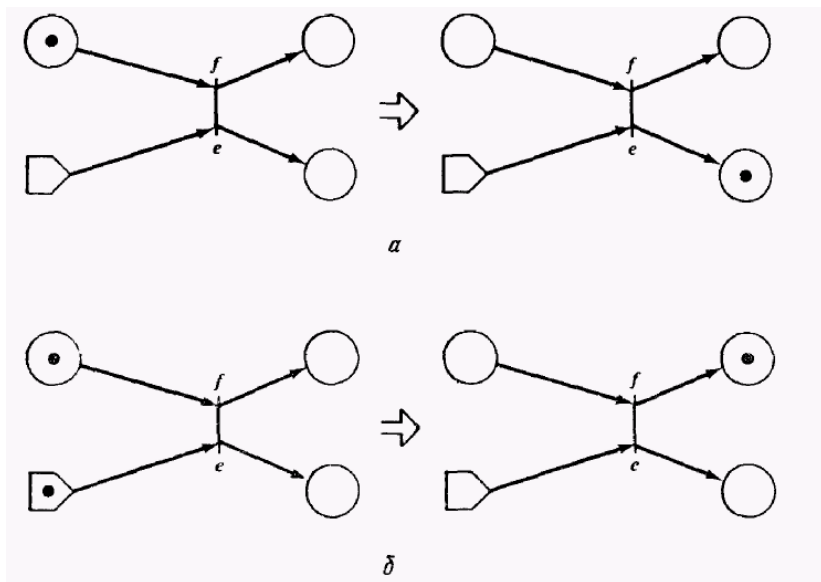


Рис. 2. Запуск переключателя. Позиция переключающего входа изображена в виде пятиугольника.
a — пустой переключатель; *b* — полный переключатель.

Переключатель — это специальный переход со специальным входом, называемым переключающим, и точно двумя выходами (один помечен символом e для пустого переключающего входа, а другой помечен символом f — для непустого переключающего входа). Переключаемый переход запускается, когда он разрешен (независимо от состояния специального переключающего входа). Когда он запускается, фишка помещается в выход, помеченный символом e , если переключающий вход пуст, или в выход, помеченный символом f , если переключающий вход не пуст. Таким образом, в зависимости от состояния переключателя запуск переключаемого перехода приведет к одной из двух возможных маркировок. Фишка удаляется из переключающего входа, если он имел ее, поэтому после того, как переключаемый переход запустится, переключающий вход будет пуст. Рассмотренные расширения сетей Петри были предложены для решения определенных задач, с которыми встретились исследователи при попытках моделировать реальные системы. Однако акцент в этих работах сделан на моделирование, а не на теоретическую мощьность сети Петри, поэтому в них не рассматривается, были ли эти расширения необходимыми или достаточными для решения общих

Кононюк А.Е. Графы

проблем моделирования. Фактически во всех случаях рассматриваемые сети были безопасными и, следовательно, множество достижимости — конечным, т. е. эти сети представляются конечными автоматами, которые в свою очередь легко представляются ординарными сетями Петри. Очевидно, что в описанных случаях были введены не расширения, а некоторые удобства. Ранее показано, что многие расширения ограниченной модели сети Петри, например, допускающие кратные дуги и петли, в действительности являются не расширениями, а средствами, облегчающими моделирование.

Из сказанного выше не становится ясным, что является ограничением (если оно существует) сетей Петри. Ответ на этот вопрос был найден после получения ответа на подобный вопрос о P- и V-операциях Дейкстры над семафорами.

Дейкстра определил свои P- и V-операции над семафорами для обеспечения синхронизации и связи в системах взаимодействующих процессов. Семафор может рассматриваться как целочисленная переменная, которая принимает только неотрицательные значения. V-операция над семафором S увеличивает значение семафора на единицу: $S = S + 1$. P-операция, наоборот, уменьшает S на единицу до тех пор, пока результат не становится равным нулю; при $S = 0$ процесс, прежде чем продолжать свою работу, должен ждать момента, когда S можно будет уменьшить. Связь между семафорами и сетями Петри была выявлена ранее.

Поскольку P- и V-операции были предложены как механизм для решения всех задач синхронизации программ, то естественно возникает вопрос о *полноте*, т. е. об их способности к решению всех задач координации. Патил в 1971 г. предложил доказательство того, что P- и V-операции недостаточно мощное средство для решения всех задач координации. Его подход был весьма прост: он сформулировал задачу синхронизации, которая не может быть решена с помощью P- и V-операций, — это *задача о курильщиках сигарет*.

Задача о курильщиках сигарет включает (по меньшей мере) четыре процесса, которые моделируют агента и трех курильщиков. Каждый курильщик непрерывно изготавливает сигарету и курит ее. Чтобы сделать сигарету, необходимы три составные части: табак, бумага и спички. Один из курильщиков всегда имеет бумагу, другой — табак, а третий — спички. Агент обладает бесконечными запасами всех трех составных частей. Агент кладет две составные части на стол. Курильщик, имеющий третий недостающий ингредиент, может сделать и закурить сигарету, сигнализируя об этом агенту. Тогда агент помещает другие два из трех ингредиентов, и цикл повторяется.

Кононюк А.Е. Графы

Если semaфор поставить в соответствие каждой составной части, задача о курильщиках формулируется в терминах semaфоров. Semaфоры первоначально равны нулю. Агент увеличит два из трех semaфоров с помощью V-операций, а затем ждет semaфора «сделано». Соответствующий процесс курильщика уменьшает два semaфора (с помощью P-операций), а затем, произведя действия «сделать сигарету» и «закурить сигарету», увеличивает semaфор, указывая «сделано». Задача заключается в том, чтобы разработать программу процессов курильщиков для того, чтобы определить, какой из трех процессов должен действовать в очередной момент. Действия агента фиксации и не могут быть изменены.

Рис. 3 иллюстрирует очевидное «решение».

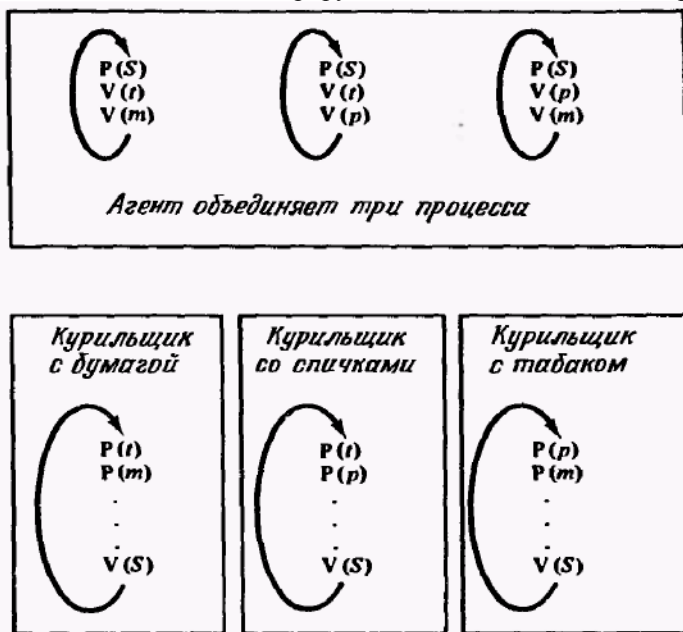


Рис. 3. Задача о курильщиках сигарет.

Предположим, агент кладет табак и бумагу [$V(t)$, $V(p)$]. Тогда курильщик с бумагой может взять табак [$P(t)$], а курильщик с табаком может взять бумагу [$P(p)$], что в результате приводит к тупику.

Патил доказал, что *никакая* последовательность P- и V-операций не может корректно решить эту задачу. Это было показано с помощью доказательства того, что все P- и V-«решения» могут быть

Кононюк А.Е. Графы

промоделированы сетями Петри определенного вида (каждый переход имеет не более двух входов), но что решением является сеть Петри другого вида, и нет способа преобразовать сеть одного вида в сеть другого вида, не допуская возможности возникновения тупика.

Существовали некоторые сомнения, связанные с решением Патила, — особенно в том, что касается массивов семафоров, но тем не менее идея решения верна. Следуя работе Патила, получена задача, которая не может быть решена R- и V-операциями или сетями Петри. То же ограничение мощности моделирования было ранее определено Келлером.

Предположим, что мы имеем два процесса производителя и два процесса потребителя. Первый процесс: производитель P_1 создает элементы данных для первого процесса потребителя C_1 , а другой производитель P_2 — для второго потребителя C_2 . Элементы данных, которые произведены, но еще не использованы, помещаются в буфер: буфер B_1 для пары (P_1, C_1) и B_2 для (P_2, C_2) . Передача данных из буферов к потребителям происходит через общий канал. Канал может передавать только по одному элементу за сеанс, причем из любого буфера любому потребителю. Производители просто помещают данные в буфера. Потребители должны координировать свои действия по использованию канала. Управляющий потребитель приказывает каналу передать данные из соответствующего буфера. Все это схематично изображено на рис. 4.

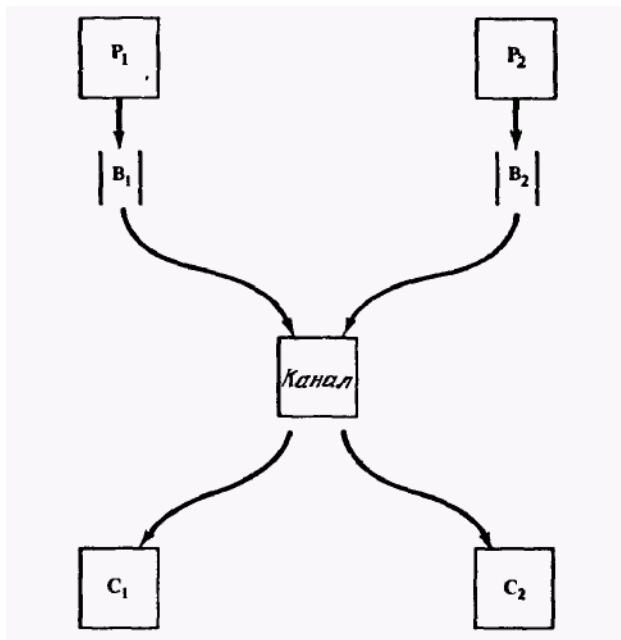


Рис. 4. Процессы производителя/потребителя с буферизацией и совместно используемым каналом.

Главная проблема, связанная с этой системой, состоит в распределении канала. Пара производитель/потребитель (P_1, C_1) должна иметь *приоритет* по отношению к (P_2, C_2) на использование канала. А именно, канал никогда не должен передавать элементы данных из буфера B_2 потребителю C_2 до тех пор, пока буфер B_1 не пуст. Это правило приоритета не позволяет системе быть моделируемой сетью Петри.

Идея доказательства относительно проста. Предположим, что мы находимся в состоянии, когда имеется μ элементов в двух буферах B_1 и B_2 . Если сейчас производитель P_1 делает перерыв, то все элементы из буфера B_1 будут в конце концов переданы производителю C_1 , и буфер B_1 будет пуст. Это позволит переместить элементы из буфера B_2 потребителю C_2 . Таким образом, существует путь из состояния μ в состояние μ' , в котором потребитель C_2 может использовать канал. Если сейчас в действительности производитель P_1 произведет дополнительно k элементов данных, то мы будем в состоянии $\mu + k$, а не в состоянии μ . Но из-за нечувствительности условий запусков переходов сетей Петри к числу фишек в позиции последовательность

Кононюк А.Е. Графы

запусков, которая переводила нас из μ в μ' , будет все еще допустимой и переведет нас из состояния $\mu + k$ в состояние $\mu' + k$. И поскольку потребитель C_2 мог использовать канал в μ' , а сеть Петри является не чувствительной к числу фишек в позиции, то потребитель C_2 может использовать канал, несмотря на присутствие k элементов в буфере B_1 . Таким образом, нечувствительность условий запусков переходов сети Петри к числу фишек в позиции не позволяет корректно моделировать эту приоритетную систему.

Более конкретно ограничение на моделирование с помощью сетей Петри состоит в неспособности проверить на наличие точно определенной маркировки в некоторой неограниченной позиции и осуществить действие в зависимости от результатов проверки. Это ограничение общеизвестно как неспособность к *проверке на нулевую маркировку* в некоторой позиции, и поэтому это свойство известно как проверка на нуль. Сети Петри не могут проверить неограниченную позицию на нуль. [Если позиция ограничена, то нуль может быть выявлен. Для ограниченной позиции p_i с границей k мы можем создать дополнительную позицию p_i — такую, что сумма $\mu(p_i) + \mu(p'_i)$ является константой, равной k для всех достижимых маркировок. Это позволяет нам проверить, равняется ли $\mu(p_i)$ нулю, проверяя, равно ли $\mu(p'_i) k$]

2.8.2. Расширения

Как указанное ограничение мощности моделирования сети Петри соотносится с предложенными расширениями сетей Петри? Все предложенные расширения направлены на создание в сетях Петри возможности проверки на нуль. Самым простым расширением сетей Петри, которое допускает проверку на нуль, являются *сдерживающие* дуги. Сдерживающая дуга показана на рис. 5.

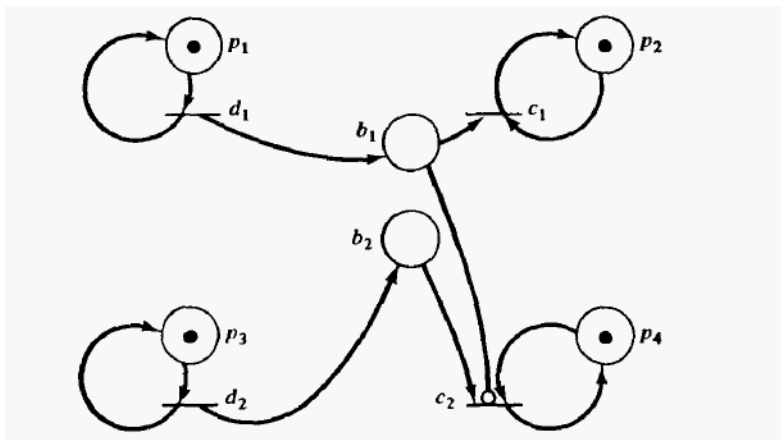


Рис. 7.5. Расширенная сеть Петри со сдерживающей дугой.

Сдерживающая дуга из позиции p_i в переход t_j имеет маленький кружок, а не стрелку у конца дуги, присоединенного к переходу. Это обозначение перешло из теории переключательных функций, где маленький кружок означает «не». Правило запуска изменяется следующим образом: переход является разрешенным, когда фишки присутствуют во всех его (обычных) входах и отсутствуют в сдерживающих входах. Переход запускается удалением фишек из всех его (обычных) входов.

Таким образом, в расширенной сети Петри с рис. 5 переход c_2 может быть запущен, только если фишки присутствуют в позициях b_2 и p_4 и отсутствуют в b_1 . Эта сеть является решением для задачи о приоритетном совместном использовании канала, чтобы показать ограниченность сетей Петри.

Сети Петри со сдерживающими дугами являются интуитивно самым прямым подходом к увеличению мощности моделирования с помощью сетей Петри. К тому же верно, что все другие предложенные расширения сетей Петри либо на самом деле не являются расширением (то есть они фактически эквивалентны обычным сетям Петри), либо являются эквивалентными сетями Петри со сдерживающими дугами. Ниже обсудим несколько предложенных расширений для того, чтобы проиллюстрировать эту точку зрения.

2.8.2.1. Области ограничения

Области ограничения были предложены Патилом для увеличения мощности моделирования сетей Петри. В контексте Патила области ограничения были только средством для того, чтобы сделать процесс моделирования проще, а не для того, чтобы увеличить мощность моделирования, поскольку все позиции в работе Патила были ограничены. Однако определение областей ограничения не исчерпывается ограниченными сетями Петри, а для более общего класса сетей Петри верно, что они эквивалентны сетям Петри со сдерживающими дугами.

Чтобы показать эквивалентность областей ограничения и сдерживающих дуг, предположим, что мы имеем сеть Петри $S = (P, T, I, O)$ с областью ограничения $Q \subseteq P$. Мы должны гарантировать, что в любой достижимой маркировке не все позиции в Q имеют фишки. Это может случиться только тогда, когда переход t_j , будучи запущенным, помещает фишки в те позиции области ограничения, которые не содержали фишек до того, как переход был запущен. Таким образом, для каждого перехода t_j , выходные позиции которого являются членами области ограничения, мы должны гарантировать, что по меньшей мере один из членов области ограничения не будет маркирован после запуска этого перехода. Для обеспечения этого мы создадим для каждой позиции p_k из области ограничения Q , не вошедшей в $O(t_j)$, новый переход $t_{j,k}$. Этот переход $t_{j,k}$ идентичен t_j , за исключением того, что он имеет сдерживающую дугу из p_k к $t_{j,k}$. Действие, производимое запуском $t_{j,k}$, совпадает с действием, производимым запуском t_j , и, если t_j может быть запущен без нарушения накладываемого областью ограничения, то по меньшей мере один из $t_{j,k}$ также может быть запущен.

В качестве примера такого построения рассмотрим сеть Петри на рис. 6.

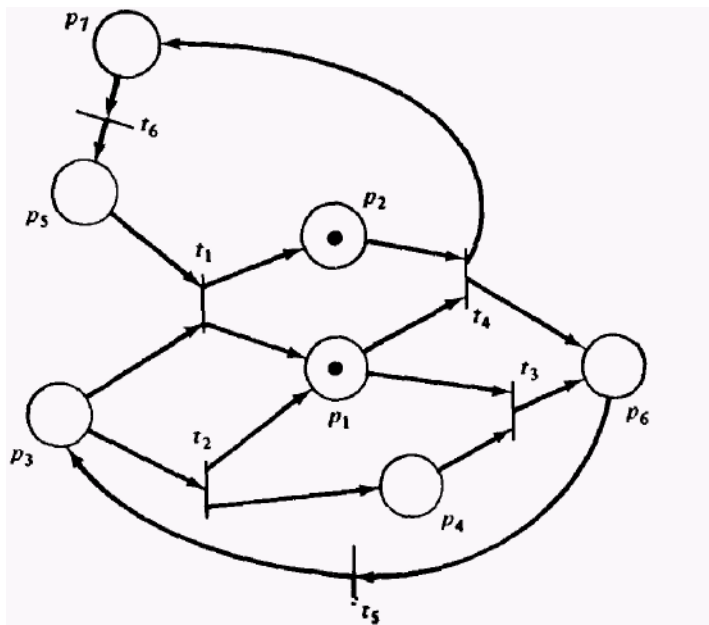


Рис. 6. Сеть Петри с областью ограничения $\{p_3, p_7\}$, которая означает, что в произвольной достижимой маркировке фишки могут быть либо в p_3 , либо в p_7 , но не в обеих одновременно.

Если мы введем область ограничения $\{p_3, p_7\}$ (то есть при любой маркировке позиции p_3 и p_7 не должны одновременно иметь фишки), то эквивалентной сетью Петри со сдерживающими дугами будет сеть с рис. 7.

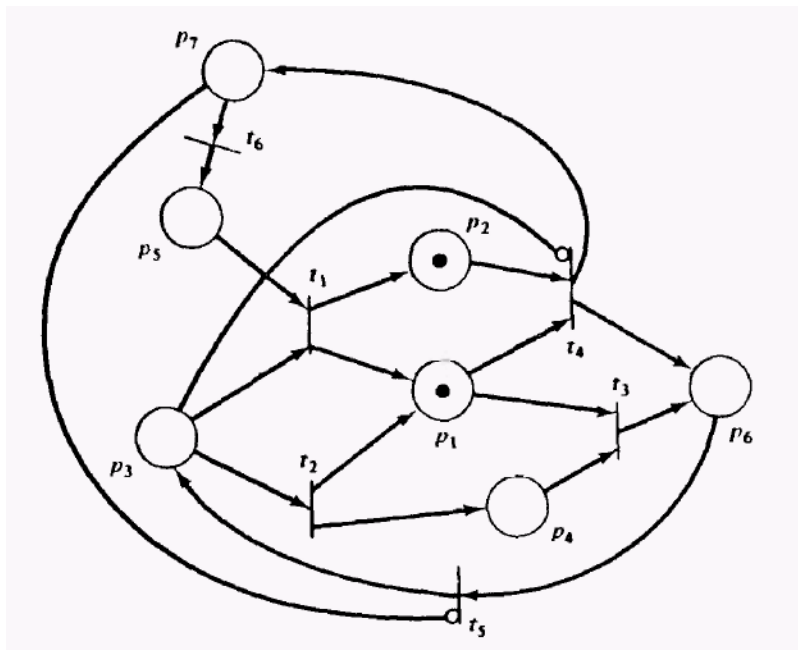


Рис. 7. Сеть Петри со сдерживающими дугами, соответствующая сети Петри с ограничениями (рис. 6). Сдерживающие дуги гарантируют невозможность одновременного появления фишек в позициях p_3 и p_7 .

Преобразование сдерживающих дуг в области ограничения является более сложным. Мы не можем просто потребовать, чтобы выход перехода не был бы маркирован в одно время со сдерживающим входом, поскольку фишки могут быть помещены в выходные позиции другими переходами. Мы должны сосредоточить свое внимание на переходе t_j . Потребуем расщепления каждого перехода t_j на два перехода, t_j' и t_j'' , и позицию p_j' . Мы определим $I(t_j') = I(t_j)$ (без сдерживающих дуг) и $O(t_j'') = O(t_j)$. Позиция p_j' представляет запуск t_j , поэтому $O(t_j') = \{p_j'\} = I(t_j'')$. Это показано на рис. 8.

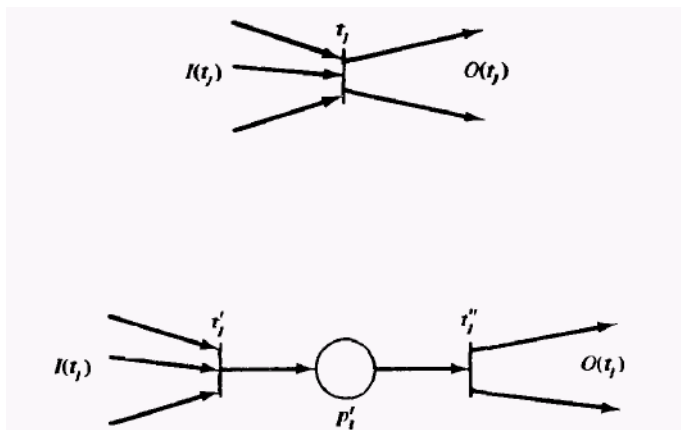


Рис. 8. Преобразование перехода в переход-НАЧАЛО и переход-КОНЕЦ с позицией, представляющей запуск перехода.

Теперь определим область ограничения $\{p_i, p_i'\}$ для каждой позиции p_i , которая является сдерживающим входом для t_j . Это обеспечит то, что переход не может быть запущен, если маркировка p_i ненулевая.

2.8.2.2. Переходы исключают ИЛИ и переключатели

Переход *исключающее ИЛИ* t_j с входом $I(t_j)$ требует, чтобы один и только один из его входов был не пустым, для того чтобы этот переход был разрешенным. Эта конструкция эквивалентна множеству переходов по одному для каждого элемента в $I(t_j)$. Каждый переход имеет один (обычный) вход, а остальные входы являются сдерживающими дугами. На рис. 9 представлен пример.

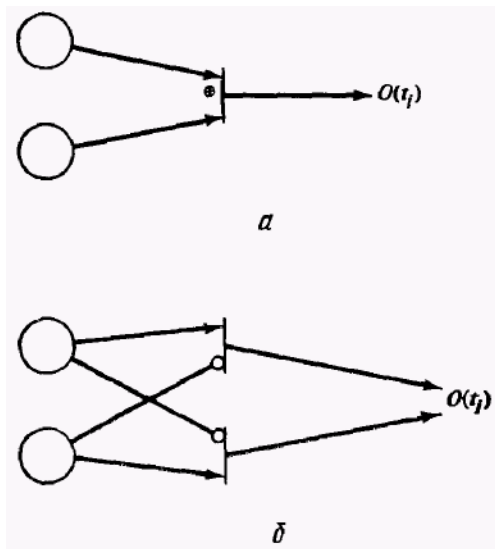


Рис. 9. Интерпретация перехода исключающее ИЛИ с помощью сдерживающих дуг.

Переключатели также могут быть легко преобразованы в сдерживающие дуги. Это показано на рис. 10.

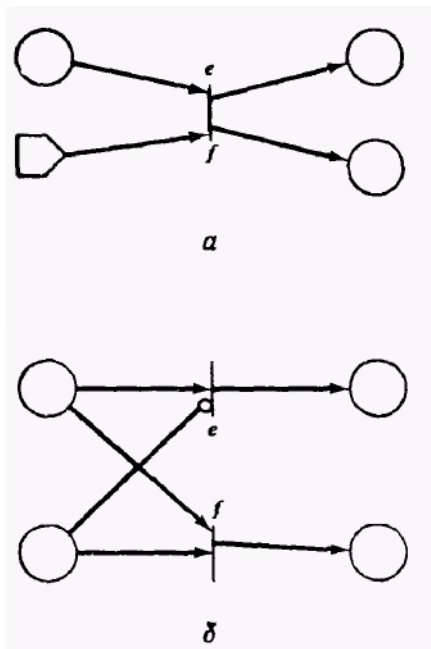


Рис. 10. Преобразование переключателей в сети Петри со сдерживающими дугами.

Способ преобразования сдерживающих дуг в переключатели или в переходы исключающее ИЛИ не ясен, но такой способ определенно существует.

2.8.2.3. Другие расширения

Имеются еще два других важных расширения сети Петри. Переходам могут быть поставлены в соответствие приоритеты так, что если t_i и t_k оба допустимы, то переход с высшим приоритетом будет запущен первым. Во временных сетях Петри каждому переходу t_j сопоставляются два момента времени $\tau_{1,j}$ и $\tau_{2,j}$. Переход t_j может быть запущен, только если он был разрешен к моменту времени $\tau_{1,j}$. Если он является разрешенным, то *должен* быть запущен до наступления момента времени $\tau_{2,j}$. Оба этих расширения могут использоваться для проверки на нуль.

В случае приоритетов легко проверить, есть ли фишка в позиции p_i (рис.11).

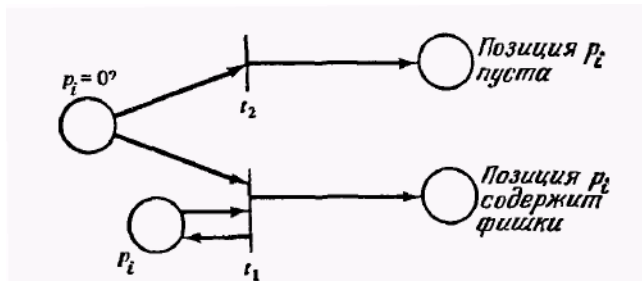


Рис. 11. Использование приоритетов для проверки позиций p на нуль. Переход t_1 имеет по сравнению с t_2 более высокий приоритет.

Если мы помещаем фишку в позицию « $p_i = 0?$ » и назначаем переходу t_1 более высокий приоритет, чем переходу t_2 , то в результате в одной из двух позиций справа появится фишка. Выбор позиции зависит от маркировки позиции p_i . Справедливость этого определяется тем фактом, что переход t_1 может быть запущен, только если он разрешен, а он является разрешенным, только если позиция p_i имеет фишку. Если t_1 не может быть запущен из-за того, что p_i пуста, то тогда и только тогда будет запущен переход t_2 .

Хэк показал способ преобразования сетей Петри с приоритетами в сети Петри со сдерживающими дугами, и наоборот. Временные сети Петри также могут осуществлять проверку позиции на нуль, моделируя приоритеты. Если мы имеем два перехода t_j и t_k и устанавливаем $\tau_{2,j} < \tau_{1,j}$, то переход t_j имеет приоритет над переходом t_k , поскольку t_j должен запускаться (если он разрешен) до того, как t_k мог бы быть разрешен для запуска.

2.8.3. Расширенные сети Петри и регистровые машины

Мы показали, что все предложенные расширения модели сетей Петри допускают возможность проверки позиции на нуль. Насколько это важно по отношению к мощности разрешения сетей Петри? Влияет ли это на возможность анализа сетей Петри?

Проверка на нуль уменьшает мощность разрешения сети Петри. Агервала, Хэк, Томас и др. показали, что появление способности проверки на нуль у модели сетей Петри позволяет сетям Петри моделировать машину Тьюринга. Таким образом, сети Петри с проверкой на нуль дают схему моделирования, с помощью которой можно моделировать любую систему. Однако почти все вопросы

Кононюк А.Е. Графы

анализа сетей Петри становятся неразрешимыми, поскольку они неразрешимы для машин Тьюринга.

Доказательство эквивалентности расширенных сетей Петри и машин Тьюринга относительно просто. Легче всего его представить в терминах регистровых машин Шепардсона и Стургиса или программных машин Минского.

Регистровая машина есть абстрактная модель ЭВМ с несколькими регистрами, которые используются для хранения произвольно больших чисел. Для манипулирования этими регистрами пишется программа. Программа есть последовательность инструкций вида «увеличить регистр n на 1», «уменьшить регистр n на 1 (только если регистр n не равен 0)», «перейти к предложению s , если регистр n не равен нулю», и т.д. Ниже представлена программа сложения содержимого регистра 2 с регистром 1.

1. Если регистр 2 равен нулю, то идти к инструкции 5.
2. Вычесть 1 из регистра 2.
3. Прибавить 1 к регистру 1.
4. Идти к инструкции 1.
5. Стоп.

Шепардсон и Стургис показали, что регистровая машина со следующими инструкциями эквивалентна машине Тьюринга.

1. $P(n)$: увеличить регистр n на 1.
2. $D(n)$: уменьшить регистр n на 1 (регистр n не равен нулю).
3. $J(n) [s]$ перейти к предложению s , если регистр n равен нулю.

Таким образом, если регистровая машина может быть преобразована в эквивалентную сеть Петри, то тем самым будет показано, что расширенные сети Петри эквивалентны регистровым машинам. Это преобразование относительно просто.

Для представления регистровой машины расширенной сетью Петри представим n регистров, используемых в программе, n позициями p'_1, p'_2, \dots, p'_n . Мы также используем $s + 1$ позицию для представления положения счетчика инструкций либо перед предложением 1 (начальная маркировка), либо после предложения i для $i = 1, \dots, s$ в программе из s предложений. Каждая инструкция в программе представляется переходом. На рис. 12 показано, как каждая из трех вышеприведенных инструкций представляется переходом в расширенной сети Петри.

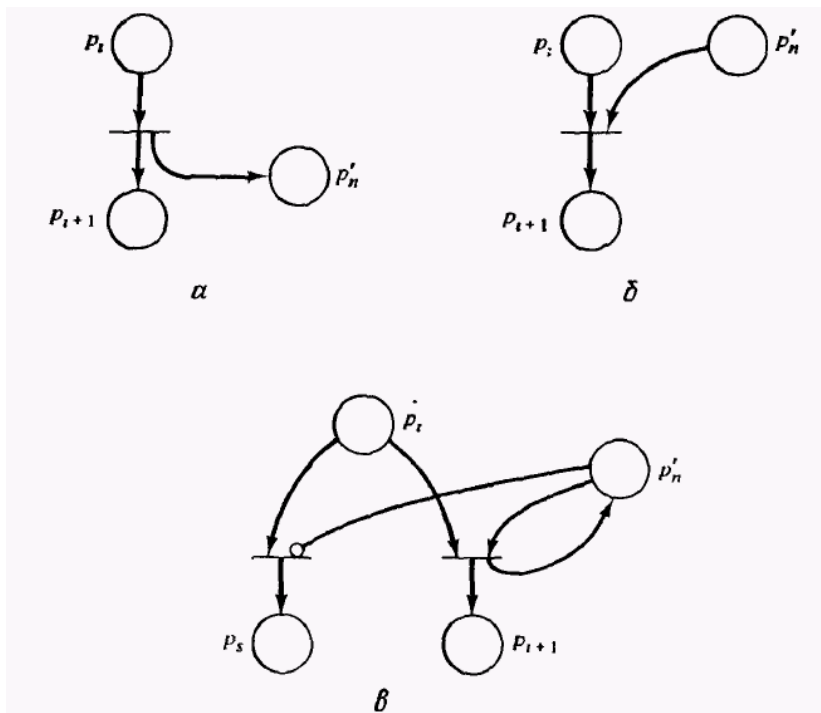


Рис. 12. Преобразование инструкции (номер i) регистровой машины в переход расширенной сети Петри со сдерживающими дугами.
 α — $P(n)$: увеличить содержимое регистра n на 1;
 δ — $D(n)$: уменьшить содержимое регистра n на 1 (содержимое регистра n должно быть положительным), β — $J(n)[s]$: переход к инструкции s в случае нулевого значения регистра n .

Из этого видно, что регистровая машина может быть преобразована в расширенную сеть Петри, и, следовательно, расширенная сеть Петри эквивалентна машине Тьюринга. Эта эквивалентность машине Тьюринга разрушает все надежды на возможность анализа расширенных сетей Петри. Однако это же доказывает, что расширенные сети Петри могут моделировать любую систему (или по крайней мере любую вычислимую систему). Таким образом, мы видим, что увеличение мощности моделирования в этом случае приводит к определенному уменьшению мощности разрешения.

Кононюк А.Е. Графы

Отметим также, что ключевым моментом в доказательстве эквивалентности сетей Петри, регистровых машин и машин Тьюринга является способность к проверке одной позиции на нуль. Таким образом, все предложенные расширения — области ограничения, переходы исключающее ИЛИ, переключатели, приоритеты, интервалы времени и сдерживающие дуги — расширяют модели сетей Петри до уровня машин Тьюринга.

Существуют другие предложения по расширению, которые не поднимают сети Петри до уровня машин Тьюринга. Первыми в качестве расширений были предложены петли и кратные входные и выходные дуги. Но как это было показано ранее, такие сети Петри фактически эквивалентны простым сетям Петри. Аналогично добавление входов ИЛИ, выходов ИЛИ, выходов исключающее ИЛИ не увеличило бы мощность моделирования сетями Петри. Вообще, кажется, что любое расширение, которое не позволяет проверку на нуль, в действительности не увеличивает мощность моделирования (или не уменьшает мощность разрешения) сетей Петри, но приводит просто к другой эквивалентной формулировке модели сети Петри (может возрасти *удобство* моделирования). В то же самое время любое расширение, которое разрешает проверку на нуль, увеличивает мощность моделирования до уровня машин Тьюринга и сводит мощность разрешения к нулю.

2.8.4. Подклассы сетей Петри

Цель расширения сетей Петри состоит в увеличении их мощности моделирования. К несчастью, побочным эффектом такого расширения является значительное уменьшение мощности разрешения расширенных сетей Петри. Мощность разрешения обычных сетей Петри также сомнительна из-за их сложности и обширности (вспомните результаты по сложности задач достижимости и ограниченности). Это привело к появлению некоторых исследований подклассов сетей Петри. Цель этих исследований состоит в определении разумных структурных ограничений, налагаемых на сети Петри, которые увеличивают мощность разрешения ограниченных моделей сетей Петри, не ограничивая существенно мощность моделирования.

Многие задачи, связанные с подклассами сетей Петри, могут быть решены. И цель этой части исследований сетей Петри достаточно проста: определить подкласс сетей Петри, который может моделировать большой класс систем (все или почти все интересующие нас системы), но для которого еще существуют простые процедуры

Кононюк А.Е. Графы

анализа (по крайней мере для интересующих нас проблем). Необходимо также, чтобы существовал простой способ для определения, является ли какая-либо сеть Петри членом определенного подкласса. Все определенные подклассы являются *синтаксическими* или *структурными* подклассами, и можно легко проанализировать структуру сети Петри для выяснения, является ли эта сеть Петри членом определенного подкласса. В этом их отличие от подклассов, которые можно определить в соответствии с динамическими свойствами, такими, как устойчивые сети Петри или ограниченные сети Петри. Такие подклассы могут иметь очень хорошие свойства, но для них очень трудно определить, является ли произвольная данная сеть Петри устойчивой или ограниченной.

Достаточно полно изучены только два главных подкласса модели сетей Петри: *автоматные* сети Петри и *маркированные графы*. Кроме того, Хэк изучил подкласс, названный сетями Петри со *свободным выбором*, и сформулировал предположения, что другой подкласс, *правильные* сети Петри могут иметь хорошие свойства с точки зрения разрешимости. Мы представим каждый из этих классов и укажем их основные свойства, достоинства и недостатки.

2.8.4.1. Автоматные сети Петри

Автоматная сеть Петри — это сеть Петри, в которой каждый переход может иметь точно один выход и один вход.

Определение 1. Автоматная сеть Петри — это сеть Петри $S=(P, T, I, O)$ — такая, что для всех $t_j \in T$, $|I(t_j)| = 1$ и $|O(t_j)| = 1$.

Некоторые свойства автоматных сетей Петри очевидны. Прежде всего автоматные сети Петри — строго сохраняющие. Это означает, что число фишек в такой сети никогда не изменяется, и мы получаем таким образом конечную систему. Отсюда следует, что дерево достижимости для автоматной сети Петри является конечным, и, следовательно, все вопросы анализа для автоматных сетей Петри разрешимы. Фактически автоматные сети Петри эквивалентны автоматам, как они определяются в теории автоматов и формальных языков. Таким образом, эти модели имеют ограниченный интерес, несмотря на их мощность разрешения, из-за ограниченной мощности моделирования конечных автоматов.

2.8.4.2. Маркированные графы

Другим, часто упоминаемым в литературе подклассом сетей Петри является класс маркированных графов. Маркированный граф есть сеть Петри, в которой каждая позиция является входом для точно одного перехода и выходом точно одного перехода. Иначе говоря, мы можем сказать, что каждая позиция имеет точно один вход и один выход.

Определение 2. *Маркированный граф* есть сеть Петри $S = (P, T, I, O)$, такая, что для каждой $p_i \in P$:

$$|I(p_i)| = |\{t_j \mid p_i \in O(t_j)\}| = 1 \text{ и } |O(p_i)| = |\{t_j \mid p_i \in I(t_j)\}| = 1.$$

Маркированные графы двойственны автоматным сетям Петри в теоретико-графовом смысле, поскольку в автоматных сетях Петри переходы имеют один вход и один выход, в то время как в маркированных графах один вход и один выход имеют позиции. Они являются двойственными также и с точки зрения моделирования. В автоматных сетях Петри легко представить конфликтные ситуации с помощью позиции с несколькими выходами, но нельзя моделировать создание и уничтожение фишек, необходимых для моделирования параллельности, или ожидания, свойственные задачам синхронизации. С другой стороны, маркированные графы могут моделировать параллельность и синхронизацию, но не могут моделировать конфликты или принятие решений, зависящие от данных. Изучены такие свойства маркированных графов, как активность, безопасность и достижимость. Наиболее интересными структурными компонентами маркированного графа при изучении указанных свойств являются его *циклы*. Цикл в маркированном графе — это последовательность переходов $t_{j_1}, t_{j_2}, t_{j_3}, \dots, t_{j_k}$ — такая, что для каждой пары переходов t_{j_r} и $t_{j_{r+1}}$ из этой последовательности существует позиция p_{i_r} — такая, что $p_{i_r} \in O(t_{j_r})$ и $p_{i_r} \in I(t_{j_{r+1}})$.

Таким образом, цикл есть замкнутый путь из какого-либо перехода обратно в этот же переход.

Например, в маркированном графе на рис. 13 последовательность $t_1 t_2 t_1$ является циклом, как и последовательности $t_4 t_3 t_4$ и $t_2 t_4 t_3 t_1 t_2$.

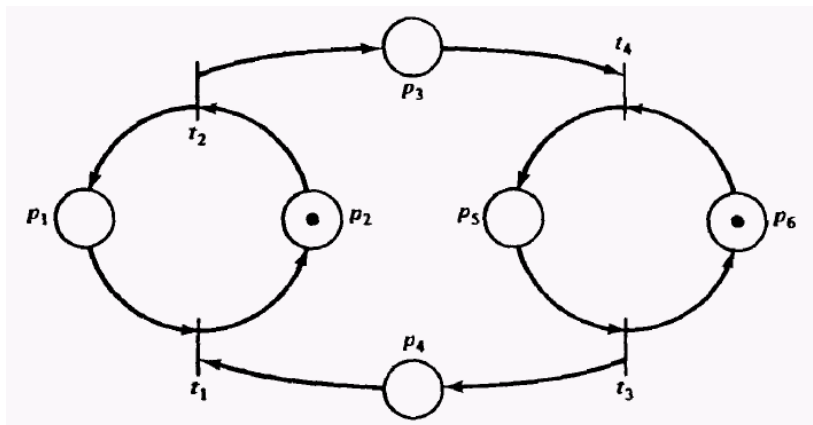


Рис. 7.13. Маркированный граф.

Важность циклов в маркированных графах вытекает из следующей теоремы.

Теорема 1. Число фишек в цикле маркированного графа не изменяется в результате запусков переходов.

Используя эту теорему, легко показать следующее.

Теорема 2. Маркировка является активной тогда и только тогда, когда в каждом цикле маркированного графа присутствует по меньшей мере одна фишка.

Теорема 3. Активная маркировка является безопасной тогда и только тогда, когда каждая позиция маркированного графа находится в цикле с числом фишек, равным единице.

Эти теоремы предоставляют простой и легкий путь исследования структуры маркированного графа и определения из его структуры и начальной маркировки, является ли маркированный граф активным или безопасным. Можно также показать, что задача достижимости маркировок для маркированных графов разрешима. Например, отметим следующее.

Теорема 4. Маркировка μ' достижима из активной маркировки μ в сильно связанном маркированном графе тогда и только тогда, когда общее число фишек в каждом цикле маркированного графа совпадает для маркировок μ и μ' .

Большая мощность разрешения маркированных графов очевидна из следующих теорем и работ по маркированным графам. Однако существует связь между мощностью разрешения и мощностью моделирования, и высокая мощность разрешения маркированных графов частично простирается из низкой мощности моделирования.

Поэтому исследователи пытались выделить другие подклассы сетей Петри, которые оставляли бы высокой мощность разрешения маркированных графов и увеличивали их мощность моделирования.

2.8.4.3. Сети Петри со свободным выбором

Хэк в своей диссертации на степень магистра в МТИ определил и исследовал один такой подкласс сетей Петри — сети Петри со свободным выбором. Этот подкласс допускает и конфликты автоматных сетей Петри, и параллельность маркированных графов, но в более ограниченном виде, чем в обычных сетях Петри.

Определение 3. Сеть Петри со свободным выбором есть сеть Петри $C = (P, T, I, O)$ — такая, что для всех $t_j \in T$ и $p_i \in I(t_j)$ либо $I(t_j) = \{p_i\}$, либо $O(p_i) = \{t_j\}$.

Важность этого определения заключается в том способе, которым оно допускает управляемые конфликты. Конфликт появляется только тогда, когда одна позиция является входом нескольких переходов. По определению сетей Петри со свободным выбором, если позиция является входом для нескольких переходов (потенциальный конфликт), то она является единственным входом всех этих переходов. Следовательно, либо все эти конфликтующие переходы одновременно являются разрешенными, либо ни один из них. Это позволяет свободно осуществлять выбор (разрешение конфликта) запускаемого перехода, присутствие фишек в других позициях не влияет на выбор запускаемого перехода.

Эта ограниченная форма конфликтов была допущена Хэком для доказательства необходимого и достаточного условия того, чтобы маркированная сеть Петри со свободным выбором являлась активной или безопасной. Условие активности связано с маркировками ловушек и тупиков в сети. *Ловушка* — это такое множество позиций, что каждый переход, входом для которого является одна из позиций множества, имеет выходом другую позицию того же множества. Это означает, что если в какой-либо позиции ловушки имеется фишка, то она будет в одной из позиций ловушки всегда. Запуск перехода может перемещать фишку между позициями, но удалить фишку из ловушки он не может. *Тупик* есть такое множество позиций, что каждый переход, который имеет в качестве выхода одну из позиций тупика, использует какую-либо позицию тупика в качестве входа. Это означает, что если все позиции тупика в какой-то момент станут пустыми, то все это множество позиций останется пустым всегда. Ни один переход не может поместить фишку в тупик потому, что в тупике

Кононюк А.Е. Графы

нет фишек, которые сделали бы разрешенным переход, выходом которого служит позиция из тупика.

Хэк доказал, что необходимым и достаточным условием активности маркированной сети Петри со свободным выбором является требование того, чтобы каждый тупик содержал ловушку с фишкой. Эта теорема основывается на работе Коммонера. Для определения необходимого и достаточного условия безопасности нужно показать, что сеть Петри со свободным выбором покрывается объединением автоматных сетей Петри.

К сожалению, дальнейшего развития работы по сетям Петри со свободным выбором не получили, и поэтому свойства сетей Петри со свободным выбором, связанные с достижимостью, эквивалентностью, включением, языками и т. д., рассмотрены не были.

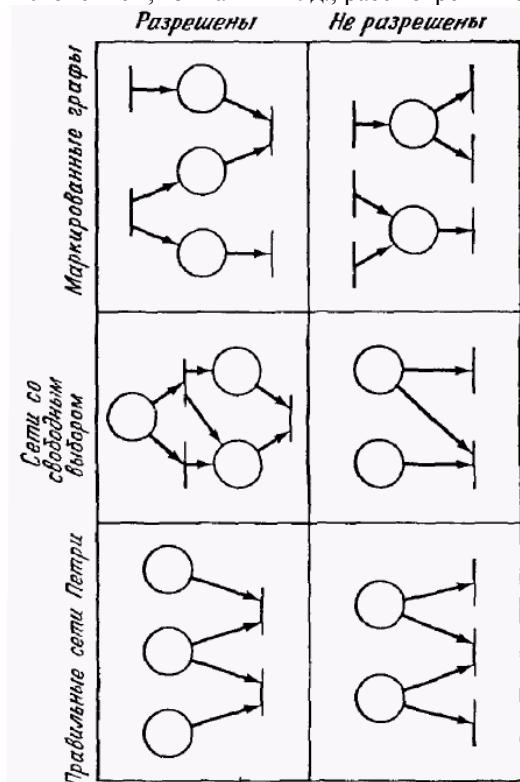


Рис. 14. Диаграмма осуществимости некоторых структурных конфигураций в различных сетях Петри.

2.8.4.4. Правильные сети Петри

Хэком также был определен другой подкласс сетей Петри, названный *правильными* сетями Петри. В правильных сетях требуется, чтобы каждый переход имел не более одной входной позиции, которая совместно используется с другим переходом и поэтому служит для ограничения возможностей возникновения конфликтов. Исследования свойств этого подкласса сетей Петри до сих пор не проводились.

Доказательство Патила того факта, что P/V-системы не могут решить всех задач синхронизации, и контрдоказательство Парнаса весьма кратки и интересны. Они привели к доказательству того, что сети Петри не могут моделировать все без исключения параллельные системы. Эти результаты привели Агервалу к исследованию вопроса о том, что должно присутствовать в модели, которая может описывать все системы.

2.9. Обобщения сетей Петри

Анализ языков, порождаемых сетями Петри, показывает, что сети Петри по выразительной мощности, т.е. по способности адекватно моделировать поведение дискретных систем, превосходят такие классы моделей, как конечные автоматы, но все же не обладают "универсальными" возможностями, так как некоторые классы языков не могут порождаться сетями Петри. Это заставило искать такие обобщения сетей, которые увеличивали бы их выразительную мощность.

В этом разделе будет рассмотрен ряд модификаций сетей Петри, направленных на усиление их моделирующих способностей, и проведено сравнение получаемых классов друг с другом и с классом сетей Петри.

2.9.1. Счетчиковые автоматы

Рассматриваемые ниже обобщения сетей Петри приводят к универсальным моделирующим системам, порождающим рекурсивно-перечислимые классы языков, т.е. к системам, равномошным машинам Тьюринга. Этот факт проще устанавливается сравнением обобщенных сетей со *счетчиковыми автоматами*, модификациями машин Минского, которые, как известно, равномошны машинам Тьюринга. Счетчиковый автомат состоит из конечного множества счетчиков

Кононюк А.Е. Графы

$\{x_i | 1 \leq i \leq n\}$, алфавита и программы автомата.

Программа автомата представляет собой связный ориентированный граф с одной начальной вершиной без входных дуг, с одной заключительной вершиной без выходных дуг; из остальных вершин выходит одна или две дуги. Вершинам приписаны операторы одного из шести типов (рис. 1):

- а) начальной вершине приписан оператор старт, единственный в программе,
- б) заключительной вершине приписан оператор стоп, единственный в программе,
- в) оператор прибавления единицы $x_i := x_i + 1$ может быть приписан вершине с одной выходной дугой,
- г) оператор печати символа из алфавита автомата: печать a также приписывается вершинами с одной выходной дугой,
- д) оператор условного вычитания единицы: если $x_i \neq 0$ то $x_i := x_i - 1$ может быть приписан вершинам с двумя выходными дугами,
- г) оператор недетерминированного перехода на также приписывается вершинам с двумя выходными дугами.

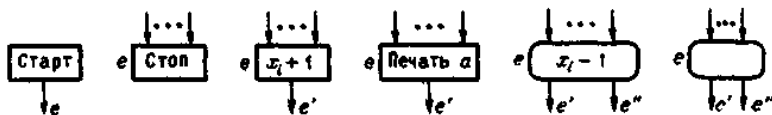


Рис. 1.

На рис. 1 операторы представлены в некоторой сокращенной форме. Метки у выходных дуг указывают вершины (метки операторов), на которые эти дуги заводятся.

Счетчиковый автомат порождает (печатает) слово в алфавите автомата, продвигаясь по программе, меняя значения счетчиков и печатая очередной символ. Движение начинается в начальной вершине. При прохождении оператора $x_i := x_i + 1$ значение счетчика x_i увеличивается на 1. При прохождении оператора если $x_i \neq 0$ то $x_i := x_i - 1$ осуществляется проверка равенства значения счетчика x_i нулю и, в зависимости от результата, осуществляется переход по одной из двух дуг, выходящих из данной вершины: по левой, если $x_i \neq 0$, и по правой, если $x_i = 0$, причем из x_i вычитается единица. Оператор недетерминированного перехода выбирает произвольным образом направление дальнейшего движения. При прохождении оператора печати автомат печатает символ на ленте автомата справа от уже на-

Кононюк А.Е. Графы

печатанных символов, формируя таким образом на ленте некоторое слово (в начале работы автомата на ленте пустое слово).

Если автомат достигает заключительную вершину, он останавливается, и результатом его работы является напечатанное им слово, в противном случае он закичивается. Поскольку в общем случае программа автомата может содержать операторы недетерминированного перехода, то автомат, остановившийся в заключительной вершине, печатает одно из множеств слов, которые он может напечатать. Это множество слов образует язык, порождаемый автоматом. Если программа не содержит операторов недетерминированного перехода и автомат останавливается, то язык состоит из единственного слова. Если автомат закичивается при любом возможном варианте выполнения его программы, то он порождает пустой язык.

Машина Минского не содержит алфавита, операторов печати и недетерминированного перехода. Поэтому она является "числовой" машиной, задающей некоторую функцию $f: \mathbf{N}^n \rightarrow \mathbf{N}^n$, где $f(m_1, \dots, m_n)$ — вектор значений счетчиков x_1, \dots, x_n при остановке машины, m_1, \dots, m_n — начальные значения тех же счетчиков. Функция f не определена тогда и только тогда, когда машина закичивается. Для любой частично-рекурсивной функции существует задающая ее машина Минского, поэтому машины Минского эквивалентны машинам Тьюринга по своим вычислительным возможностям.

Теорема 1. *Любой рекурсивно перечислимый язык может быть порожден некоторым счетчиковым автоматом.*

Доказательство. Известно, что любой рекурсивно перечислимый язык может быть взаимнооднозначно отображен в рекурсивно перечислимое множество целых неотрицательных чисел, причем многими способами. Выберем следующую кодировку $K: A^* \rightarrow \mathbf{N}$ слов из алфавита $A = \{a_1, \dots, a_n\}$ числами:

$$K(\lambda) = 0,$$

$$\forall a_i \in A, \forall \alpha \in A^*: K(a_i \alpha) = i + (n + 1) \cdot K(\alpha).$$

Например, кодом двоичного слова 1001 в алфавите $\{0,1\}$ служит число $(2 + 3 \cdot (1 + 3 \cdot (1 + 3 \cdot 2))) = 68$, а кодом строки цифр 123 в алфавите десятичных цифр $\{1, \dots, 9\}$ служит число $(1 + 10 \cdot (2 + 10 \cdot 3)) = 321$. Кодировочная функция K взаимнооднозначна, что видно из ее определения, поэтому существует обратная (декодировочная) функция $K^{-1}: \mathbf{N} \rightarrow A^*$, сопоставляющая числам слова в алфавите A . (Функция K^{-1} является частично определенной, так как функция K отображает множество A^* в \mathbf{N} , а не на \mathbf{N} .)

Счетчиковый автомат на рис. 2 реализует декодирующую функцию K^{-1} , которая является частично-рекурсивной.

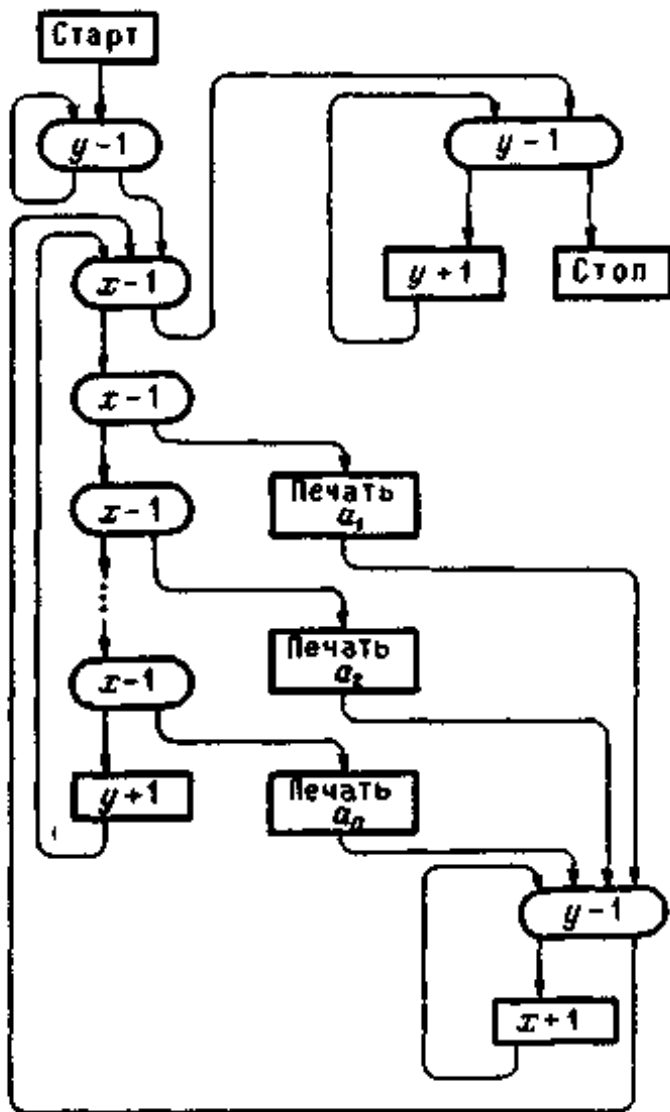


Рис. 2

Этот автомат имеет два счетчика x и y , он останавливается и печатает слово $K^{-1}(m)$, если m является начальным значением счетчика x и одновременно числом-кодом. В противном случае автомат закичивается.

Таким образом, любой рекурсивно перечислимый язык является образом некоторого рекурсивно перечислимого множества чисел при декодирующей функции K^{-1} , вычисляемой некоторым счетчиковым автоматом. В свою очередь, каждое рекурсивно перечислимое множество чисел является областью значений некоторой частично-рекурсивной функции f , которая может быть задана машиной Минского. Следовательно, любой рекурсивно перечислимый язык может быть областью значений функции, задаваемой счетчиковым автоматом, как показано на рис. 3,а.

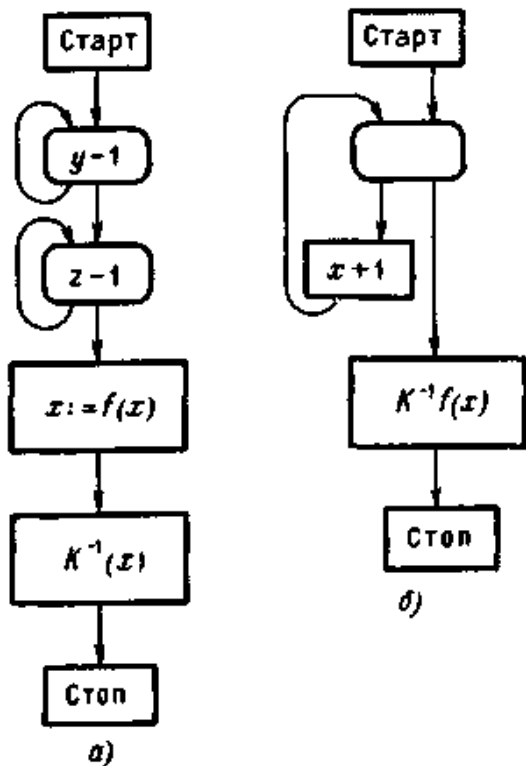


Рис. 3.

На этом рисунке фрагмент " $x = f(x)$ " соответствует подпрограмме, которая представляет собой программу машины Минского (без начального и заключительного операторов), задающей функцию f , а фрагмент "печать $K^{-1}(x)$ " является программой счетчикового автомата на рис. 2 (без начального и заключительного операторов).

Наконец, любой рекурсивно перечислимый язык может быть порожден счетчиковым автоматом, программа которого начинается (после оператора **старт**) недетерминированным переходом, позволяющим получить произвольное значение счетчика x (см. рис. 3, б), и затем совпадает с программой автомата на рис. 3, а. Язык, порождаемый автоматом на рис. 3,б, совпадает с областью значений суперпозиции функций $K^{-1}f$.

2.9.2. Ингибиторные сети и сети с приоритетами

Вышемы встречались с ситуациями, когда сети Петри моделировали некоторые арифметические операции. Сравнение счетчиковых автоматов с сетями Петри показывает, что пяти из шести операторов автомата можно сопоставить фрагменты сетей Петри, имитирующие их работу. Например, оператору **старт** можно поставить в соответствие начальное место без входных дуг и с единичной разметкой (рис. 4,а); оператору **стоп** - заключительное место без выходных дуг (рис. 4,б); оператору печати символа a — переход, помеченный символом a' , с одним входным и одним выходным местом (рис. 4,в). Оператор недетерминированного перехода моделируется двумя переходами с общим конфликтным входным местом и разными выходными местами (рис. 4, г). Оператор прибавления единицы $x_i := x_i + 1$ может быть представлен переходом с одним входным и одним выходным местом для моделирования управляющих связей оператора с другими операторами программы и еще одним выходным местом, которое соответствует счетчику x_i автомата (рис. 4, д).

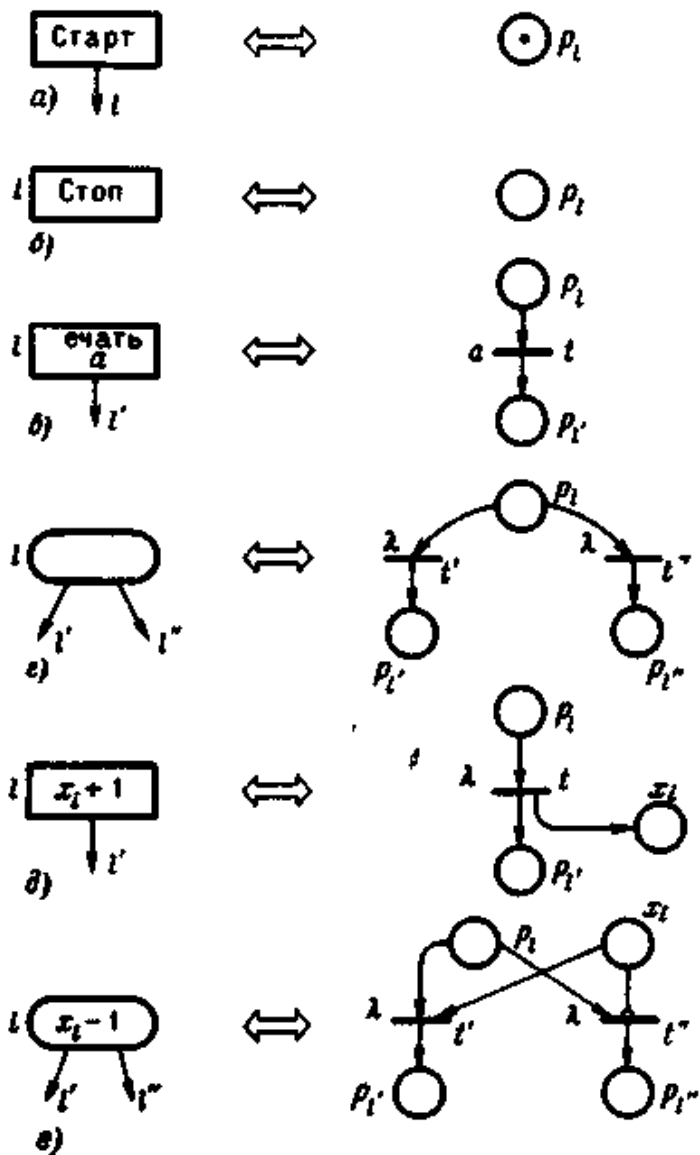


Рис.4

Кононюк А.Е. Графы

Однако оператор условного вычитания единицы не удастся промоделировать средствами сетей Петри. Причина этого состоит в том, что в сети Петри можно заметить (и отметить это срабатыванием некоторого перехода) тот факт, что место сети изменило разметку с нулевой на ненулевую, но нельзя отметить срабатыванием перехода факт изменения разметки с ненулевой на нулевую. Таким образом, из двух альтернатив ($x_i \neq 0$ и $x_i = 0$), содержащихся в операторе условного вычитания единицы, в сети Петри можно представить только одну, первую, но нельзя отразить проверку на ноль, так как сеть не может реагировать непосредственно на отсутствие фишки в месте (если это место неограниченно, в противном случае это можно сделать косвенным путем). Келлер и Косараю обратили на этот факт внимание и показали, что сети Петри не могут моделировать машины Минского и Тьюринга.

Естественным в этой ситуации был шаг, сделанный Флинном и Адджервалой, которые модифицировали сети Петри, введя в них специальные ингибиторные дуги, осуществляющие проверку на нулевую разметку, и показали, что получаемое обобщение дает класс сетей (ингибиторные сети), которые могут моделировать счетчиковые автоматы и, следовательно, порождать любые рекурсивно перечислимые языки.

Ингибиторная сеть представляет собой сеть Петри, дополненную специальной функцией инцидентности $F_I: P \times T \rightarrow \{0, 1\}$, которая вводит *ингибиторные дуги* для тех пар (p, t) , для которых $F_I(p, t) = 1$. Ингибиторные дуги связывают только места с переходами, на рисунках их изображают заканчивающимися не стрелками, а маленькими кружочками. Правило срабатывания переходов в ингибиторной сети модифицируется следующим образом. Переход t может сработать при разметке M , если

$$\forall p \in {}^*t: M(p) \geq F(p, t) \wedge M(p) \cdot F_I(p, t) = 0.$$

Другими словами, переход t может сработать, если каждое его входное место p , соединенное с t "обычной" дугой с кратностью $W(p, t)$, содержит не менее $W(p, t)$ фишек, а каждое входное место, соединенное с t ингибиторной дугой (ее кратность всегда равна 1), имеет нулевую разметку.

Используя ингибиторную дугу, можно промоделировать оператор условного вычитания если $x_i \neq 0$ то $x_i = x_i - 1$ фрагментом ингибиторной сети, показанной на рис. 4, в. Место x_i соответствует счетчику x_i , срабатывание перехода t' - альтернативе, связанной с выполнением условия $x_i \neq 0$ и последующим вычитанием единицы из x_i , срабатывание перехода t'' — альтернативе, связанной с выполнением условия $x_i = 0$.

Кононюк А.Е. Графы

Для построения ингибиторной сети, моделирующей счетчиковый автомат, достаточно каждому оператору программы автомата сопоставить моделирующий его фрагмент ингибиторной сети и затем связать эти фрагменты в единую сеть. С этой целью символы входных и выходных "управляющих" мест переходов в фрагментах сети выбираются в соответствии с метками операторов в автомате, как показано на рис. 4, а каждому счетчику x_i автомата сопоставляется место x_i . Переходы, соответствующие операторам печати, помечаются соответствующими символами, остальные переходы вводятся как λ -переходы. Фрагменты связываются в сеть путем слияния одинаково помеченных выходных и входных мест.

Теорема 2. *Любой рекурсивно перечислимый язык может быть порожден помеченной ингибиторной сетью как префиксный или терминальный язык сети.*

Доказательство. Способ, которым строится помеченная ингибиторная сеть, моделирующая счетчиковый автомат, непосредственно убеждает в эквивалентности терминальных языков, порождаемых обеими системами, если для сети фиксировать терминальный язык с такой разметкой M_β , при которой все места сети, кроме места, соответствующего оператору стоп, имеют нулевую разметку, а это место - единичную. Таким образом, класс $\mathcal{L}^{\lambda_0}(I)$ всех терминальных языков (помеченных) ингибиторных сетей является классом рекурсивно перечислимых языков. Класс $\mathcal{L}^{\lambda}(I)$ образован префиксами языков из класса $\mathcal{L}^{\lambda_0}(I)$ рекурсивно перечислимых языков и поэтому он также является классом рекурсивно перечислимых языков.

При описании функционирования сетей Петри отмечался недетерминизм следующего рода: если несколько переходов могут сработать, то срабатывает любой из них. В реальных дискретных системах имеют место ситуации, когда из двух готовых работать устройств требуется запустить сначала одно, например, первое, а затем второе. Другими словами, одно из устройств имеет приоритет на запуск перед другим в том случае, если оба готовы работать. Эти ситуации не моделируются в сетях Петри в силу принятого в них правила срабатывания нескольких готовых к срабатыванию переходов.

Модифицируем определение сети Петри следующим образом. Введем множество приоритетов PR , элементы которого частично упорядочены некоторым отношением \leq (меньше или равно). С каждым переходом t сети Петри свяжем его приоритет $pr(t) \in PR$. Правило срабатывания перехода модифицируем, дополнив его условием: переход t может сработать, если для любого другого перехода t' этой сети, который может сработать по стандартному условию, $pr(t') \leq pr(t)$. Другими словами, если несколько переходов готовы работать, то срабатывает любой

такой переход, приоритет которого не меньше приоритетов остальных готовых к срабатыванию переходов. Такую модификацию сети Петри называют *сетью с приоритетами*.

На рис. 5,а показан пример простой сети с приоритетами, для которой в качестве множества приоритетов выбрано множество натуральных чисел и отношение совпадает с отношением "больше или равно".

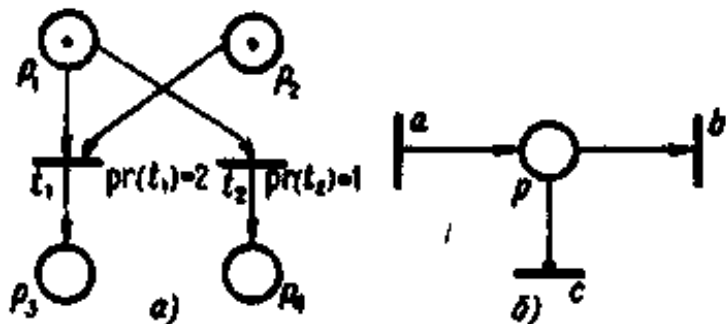


Рис. 5.

Пусть $pr(t_2) = 1 < pr(t_1) = 2$. Если $M_0(p_1) = 0$, то ни один из переходов не может сработать. Если $M_0(p_1) = 1$ и $M_0(p_2) = 0$, то срабатывает только переход t_2 , который перемещает фишку из места p_1 в место p_4 . Если $M_0(p_1) = M_0(p_2) = 1$, как показано на рис. 5, а, то по стандартным условиям оба перехода могут сработать, но поскольку $pr(t_2) < pr(t_1)$, то в этой сети с приоритетами разрешено сработать только переходу t_1 , который лишает оба своих входных места фишек и помещает фишку на место p_3 . Легко заметить, сравнив данную сеть с ингибиторной сетью на рис. 4, е, что обе они функционируют одинаково и что поэтому сеть на рис. 5, а моделирует оператор условного вычитания единицы счетчиков-врг автомата. Отсюда следует справедливость следующей теоремы, аналогичной теореме 2 как по формулировке, так и по доказательству.

Теорема 3. *Любой рекурсивно перечислимый язык может быть порожден помеченной сетью с приоритетами как ее префиксный или терминальный язык.*

Теорема 4. *Проблемы ограниченности, достижимости, эквивалентности по префиксным и терминальным языкам, пустоты этих языков неразрешимы для ингибиторных сетей и сетей с приоритетами.*

Доказательство. Для машин Минского и, следовательно, счетчиковых автоматов неразрешима проблема остановки (останавливается ли машина, начав работать с заданными начальными значениями счетчиков?).

Кононюк А.Е. Графы

При построении ингибиторной сети, моделирующей заданный счетчиковый автомат, счетчику x_i в сети соответствует место x_i , а текущему состоянию счетчика x_i - текущая разметка места x_i . Поэтому проблема ограниченности места в ингибиторной сети равносильна проблеме ограниченности счетчика в автомате. Последняя проблема неразрешима. Чтобы убедиться в этом, достаточно показать неразрешимость "более узкой" проблемы, а именно — не существует алгоритма, который для любого счетчика может установить, будет ли его начальное нулевое значение изменено хотя бы один раз, т.е. является ли счетчик 0-ограниченным или нет. Преобразуем произвольный заданный счетчиковый автомат следующим образом. Введем дополнительный вспомогательный счетчик x , запись в который делается только после того, как исходный автомат останавливается, заменив этот оператор на последовательность операторов $x := x + 1$ и **стоп**. Если бы была разрешима проблема 0-ограниченности, то мы могли бы решать проблему остановки счетчикового автомата, воспользовавшись описанным преобразованием. Таким образом, проблема 0-ограниченности не может быть разрешимой и, следовательно, неразрешима более общая проблема ограниченности счетчиков и проблема ограниченности ингибиторной сети.

Аналогично доказывается неразрешимость проблемы ограниченности в классе сетей с приоритетами и проблемы достижимости в обоих классах обобщенных сетей.

Неразрешимость проблем эквивалентности и пустоты языков следует из теоремы 3 и неразрешимости этих проблем для рекурсивно перечислимых языков.

Теорема 5. *Классы ингибиторных сетей и сетей с приоритетами строго мощнее класса сетей Петри и равномощны классам машин Тьюринга и Минского, т.е. являются "универсально мощными".*

Доказательство. Непосредственно следует из теорем 2 и 3. В частности, контекстно-свободный язык, относительно которого установлено, что он не порождается никакой из помеченных сетей Петри, порождается в качестве терминального (в том числе свободного) языка ингибиторной сети, показанной на рис.5, б.

2.9.3. Раскрашенные, синхронные и самомодифицируемые сети

Описываемые в этом параграфе модификации сетей Петри предлагались в первую очередь для того, чтобы более адекватно и удобно выра-

Кононюк А.Е. Графы

жать в терминах сетей особенности функционирования реальных дискретных систем. Эти модификации приводят к классам сетей, строго более мощным, чем класс сетей Петри.

Например, при моделировании сетями Петри дискретных систем фишки часто соответствуют объектам, передаваемым от компонента к компоненту системы (данным в информационных системах, деталям, ресурсам и т.п.). Зачастую эти объекты имеют дополнительные атрибуты, позволяющие различать их и использовать эти различия для управления функционированием системы. Однако фишки в сетях Петри "безлики" и не отражают такие различия. Пусть необходимо описать с помощью сетей Петри фрагмент операционной системы, управляющей обменими между тремя накопителями на магнитных дисках D_1 , D_2 , D_3 и центральным процессором ЦП через два канала A и B . При этом требуется, чтобы D_1 использовал канал A , D_2 - канал B , D_3 - оба канала A и B .

Для адекватного описания подобных ситуаций можно использовать модификацию сетей Петри, в которой фишкам приписаны некоторые признаки, например, различные цвета, а условия срабатываний переходов и правила изменения разметки сети задаются специальной таблицей, учитывающей цвета фишек. Пример такой *раскрашенной сети* показан на рис. 6.

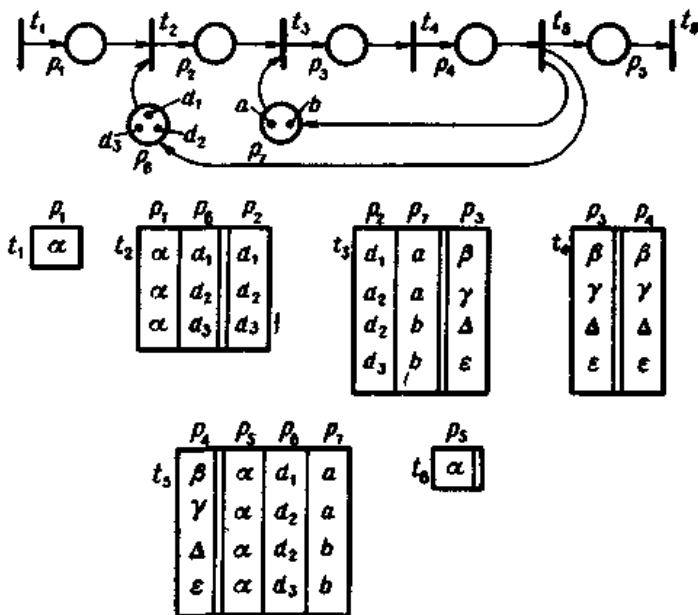


Рис. 6.

В этой сети использованы следующие фишки с признаками:
 d_1, d_2, d_3 — фишки, отмечающие возможность связи с дисковыми D_1, D_2, D_3 .
 a, b - фишки, отмечающие доступность каналов A и B ;
 $\epsilon, \beta, \gamma, \delta, \epsilon$ — вспомогательные фишки для запоминания предыстории функционирования системы.

Для каждого из шести переходов сети имеется индивидуальная таблица условий срабатывания. В ней столбцы, связанные с входными местами перехода, содержат сочетание конкретных признаков фишек, при которых переход может сработать. Столбцы, связанные с выходными местами перехода, указывают признаки, с которыми переход добавляет фишки в свои выходные места для каждого входного сочетания признаков. Признаки фишек указаны на рисунке рядом с ними. Эта сеть моделирует упоминавшийся выше фрагмент операционной системы.

Выразительная мощность раскрашенных сетей зависит от мощности множества признаков. Класс сетей с конечным множеством признаков эквивалентен классу сетей Петри, хотя при преобразовании рас-

Кононюк А.Е. Графы

крашенной сети в сеть Петри могут значительно .возрасти размеры сети.

Сети с бесконечным множеством признаков могут моделировать счетчиковые автоматы, и, следовательно, класс таких сетей равномошен классам ингибиторных сетей и сетей с приоритетами. Идея моделирования проста и состоит в следующем. Целые неотрицательные числа, и в том числе 0, кодируются признаками из бесконечного множества. В результате возникает возможность изобразить разметку $M(p)$ места p , в том числе нулевую разметку, фишкой с соответствующим признаком. Таблицы срабатываний задаются таким образом, что переход получает возможность реагировать на отсутствие "обычной" фишки в некотором месте за счет того, что в этом месте присутствует фишка с признаком, соответствующим нулю. Еще одна модификация сетей связана с изменением правила функционирования сетей - *синхронные сети*. В этом случае работа сети разбивается на такты. В начале каждого такта выясняется, какие переходы могут сработать (по стандартным условиям срабатываний) . из них выбирается некоторое максимальное множество T' взаимно неконфликтующих переходов, т.е. для которых справедливо следующее утверждение:

$$\forall p \in P: \sum_{t \in T'} F(p, t) \leq M(p).$$

Затем все такие переходы срабатывают в любом порядке; изменяя разметку сети обычным образом. На этом заканчивается такт, после чего начинается новый такт при новой разметке. Подчеркнем, что внутри такта выбранное множество срабатывающих переходов не меняется. Если после срабатывания части из выбранных переходов появится переход, который может сработать по стандартным условиям, он не добавляется к выбранным. (После срабатывания части переходов разметка не может измениться так, что для одного из выбранных, но не сработавших переходов перестанут выполняться условия срабатывания.) Другими словами, срабатывания выбранных переходов внутри такта не мешают друг другу и не оказывают влияния на ситуацию других переходов.

Такое изменение правил функционирования сетей приводит к увеличению их выразительной мощности до уровня машин Тьюринга. Чтобы убедиться в этом, достаточно показать, каким образом выражается средствами синхронных сетей оператор условного вычитания единицы

счетчикового автомата. На рис. 7 показан соответствующий фрагмент синхронной сети.

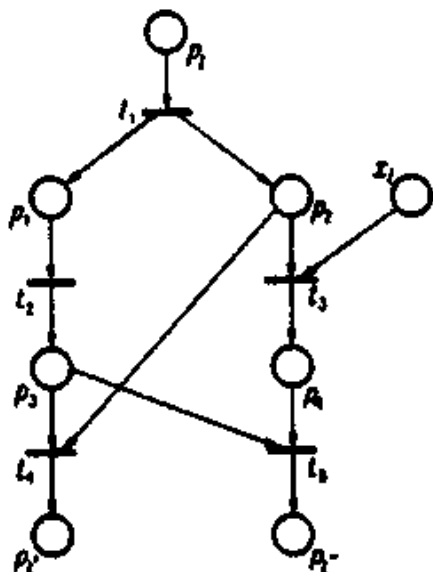


Рис. 7.

Если $M(p_1) = 1$, а $M(x_1) = 0$, то после срабатывания перехода t_1 места p_1 и p_2 имеют единичную разметку, но может сработать только переход t_2 . Затем срабатывает t_4 и окончательно фишка поступает в место p_1' . Если $M(p_1) = 1$ и $M(x_1) = 1$, то после срабатывания перехода t_1 могут сработать переходы t_2 и t_3 . В обычных сетях Петри можно разрешить срабатывание перехода t_2 и затем перехода t_4 (пока не сработал переход t_3 , в месте p_2 остается фишка). Но в синхронной сети вместе с t_2 должен сработать переход t_3 . Это приведет к тому, что переход t_4 не сможет затем сработать (место p_2 лишилось фишки), но сработает переход t_5 и поместит фишку в место p_1' .

В сетях с приоритетами и в синхронных сетях намечился отход от чисто локального принципа управления функционированием сети, принятого в сетях Петри, и переход к привлечению более глобальной информации о состоянии сети. В первом случае это информация о статических приоритетах тех переходов, которые могут сработать по стандартным условиям, во втором - о множестве всех переходов, которые готовы сработать при данной разметке. Именно в силу

Кононюк А.Е. Графы

привлечения данных о глобальной ситуации в системе и возрастает выразительная мощность сетей.

Аналогичный подход просматривается в *самомодифицируемых сетях* Фалька. В сетях Этого класса каждой дуге (p, t) и (t, p) приписана *модифицируемая кратность* $V(p, t)$ или $V(t, p)$. Если кратность дуги - число, то оно имеет тот же смысл, что и кратность в определении сетей Петри. Но в качестве кратности может выступить символ q некоторого места данной сети. В этом случае кратность дуги переменна и равна текущей разметке $M(q)$ места q . Поскольку разметка места q может динамически меняться в процессе работы сети, то и кратность тех дуг, которым сопоставлен символ q , динамически меняется. В результате число фишек, перемещаемых одним и тем же переходом от входных мест к выходным местам, динамически меняется при разных срабатываниях этого перехода. Правило срабатывания перехода t изменяется следующим образом:

$$\forall p \in T: M(p) > V(p, t),$$

где $V(p, t) = n$, если дуга (p, t) помечена числом $n \geq 1$, и $V(p, t) = M(q)$, если дуга (p, t) помечена символом места q .

Правило изменения разметки сети после срабатывания перехода t выглядит так:

$$\forall p \in P: M'(p) = M(p) - F_M(p, t) + F_M(t, p),$$

где F_M - модифицируемая функция инцидентности, определяемая следующим образом :

$$F_M(x, y) \begin{cases} n, & \text{если } xFy \wedge V(x, y) = n, \quad \text{где } n - \text{число,} \\ M(q), & \text{если } xFy \wedge V(x, y) = q, \quad \text{где } q - \text{символ места,} \\ 0, & \text{если } \neg(xFy). \end{cases}$$

Введение модифицируемой кратности позволяет достаточно просто моделировать ингибиторные сети и сети с приоритетами. На рис. 8 показано, каким образом ингибиторная дуга и фрагмент сети с приоритетами трансформируются во фрагменты самомодифицируемых сетей.

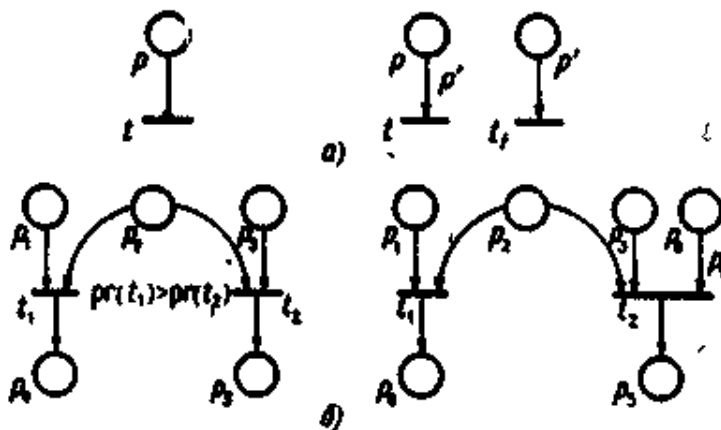


Рис. 8

В первом случае использован тот факт, что переход t может сработать в том случае, если его входное место p имеет нулевую разметку, но при текущей разметке M значение $F_M(p, t) = M(V(p, t)) = M(p') = 0$, где p' — добавленное в сеть место с "фиктивным" переходом t_f , без входящих в p' дуг и с разметкой, постоянно равной 0. Во втором случае место p_6 влияет или не влияет на готовность перехода t_2 к срабатыванию в зависимости от разметки места p_1 и входного места "конкурирующего" перехода t_1 .

В результате на самомодифицируемые сети распространяются утверждения из 2.9.2, т.е. эти сети порождают класс рекурсивно перечислимых языков, и основные сетевые проблемы для них неразрешимы.

2.10. Регулярные и иерархические сети

Сеть Петри определена как набор, состоящий из множеств мест и переходов, отношения инцидентности и функции начальной разметки. Такой способ задания сетей обычен для постановки и решения теоретических задач анализа сетей. Графовую форму представления сетей Петри удобно применять в иллюстративных целях. Однако оба эти способа представления сетей не подходят для прикладных задач моделирования дискретных систем, например, для задач, включающих автоматический анализ, синтез и преобразования сетевых моделей с помощью ЭВМ.

Кононюк А.Е. Графы

Более удобным оказывается или матричное, или аналитическое представление сетей. В первом случае сеть Петри $T = (P, T, F, M_0)$ задается двумя целочисленными матрицами: A , размерности $|P| \times |T|$, и B , размерности $|T| \times |P|$, и вектором M_0 длины $|P|$. Строка i , $1 \leq i \leq |P|$, матрицы A соответствует месту $p_i \in P$, столбец j , $1 \leq j \leq |T|$, матрицы A соответствует переходу $t_j \in T$ (множества P и T строго упорядочиваются). Строка i , $1 \leq i \leq |T|$, матрицы B соответствует переходу $t_i \in T$, столбец j , $1 \leq j \leq |P|$, матрицы B соответствует месту $p_j \in P$. Элемент $A(i, j)$ матрицы A равен $F(p_i, t_j)$, элемент $B(i, j)$ матрицы B равен $F(t_i, p_j)$. Вектор M_0 — это вектор начальной разметки. Аналитическое представление сети Петри задает ее с помощью формулы в некоторой алгебре сетей. Эта формула построена из символов, задающих некоторые элементарные сети, и сетевых операций, с помощью которых описываемая сеть может быть построена из элементарных сетей. Предложенная в работах алгебра сетей порождает класс регулярных сетей, для которых характерно топологическое регулярное строение, и дает возможность расчленить процесс анализа и конструирования сетей на совокупность этапов, на каждом из которых достаточно иметь дело с более простыми фрагментами сети.

Класс регулярных сетей формально является подклассом сетей Петри за счет топологических ограничений (ограничения на отношение инцидентности), которые возникают при аналитическом способе задания сетей. Однако будет показано, что этот класс эквивалентен по порождаемым языкам классу (ординарных) сетей Петри.

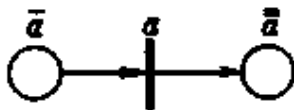
Во второй части раздела рассматривается обобщение регулярных сетей - иерархические сети, предназначенные для адекватного моделирования иерархических динамических систем. Будет показано, что иерархические сети обладают такой же выразительной мощностью, что и ингибиторные сети или сети с приоритетами.

Позже мы вернемся к алгебре регулярных и иерархических сетей как к средству описания семантики структур управления асинхронных параллельных программ, на основе которого разработаны программные механизмы управления вычислениями в языках параллельного программирования.

2.10.1. Алгебра регулярных сетей

Алгебра регулярных сетей строится с помощью операций над сетями и класса элементарных сетей. *Элементарная сеть* — это сеть вида

Кононюк А.Е. Графы



где a — символ-переход, \bar{a} — *головное место* элементарной сети, $\bar{\bar{a}}$ — ее *хвостовое место*. В формульном представлении элементарная сеть обозначается символом перехода a .

В классе регулярных сетей, который будет определен ниже, обобщено понятие разметки. Множество всех целых неотрицательных чисел расширено за счет введения элемента ω , который используется как разметка мест сети и обозначает "неограниченное число фишек" в данном месте. Такое обобщение является формальным, так как место с разметкой ω можно удалить из сети со всеми инцидентными дугами. Например, следующие сети эквивалентны и обладают одинаковыми свойствами:



Введение мест с "бесконечной разметкой" позволит топологически структурировать сети Петри, не сужая, по существу, класс рассматриваемых сетей.

Определим операции над сетями. Пусть $h(N)$ обозначает множество головных мест сети N , а $I(N)$ - множество ее хвостовых мест. (Ниже указываются правила изменения множеств $h(N)$ и $I(N)$, которые могут считаться индуктивными определениями множеств головных и хвостовых мест сети.)

Операция наложения " , ". Эта операция — обычное теоретико-множественное объединение графов, дополненное правилом формирования разметки; с помощью этой операции одна сеть накладывается на другую.

Если $N_1 = (P_1, T_1, F_1, M_{01})$ и $N_2 = (P_2, T_2, F_2, M_{02})$, то

$$N = (N_1, N_2) = (P_1 \cup P_2, T_1 \cup T_2, F_1 \cup F_2, M_0), \quad i \text{ где}$$

$$M_0(p) = \begin{cases} M_{0i}(p), & \text{если } p \in P_i, \text{ где } i = 1, 2, \text{ и } p \notin P_1 \cap P_2, \\ \min(M_{01}(p), M_{02}(p)), & \text{если } p \in P_1 \cap P_2. \end{cases}$$

Пример выполнения операции наложения показан на рис. 1,а.

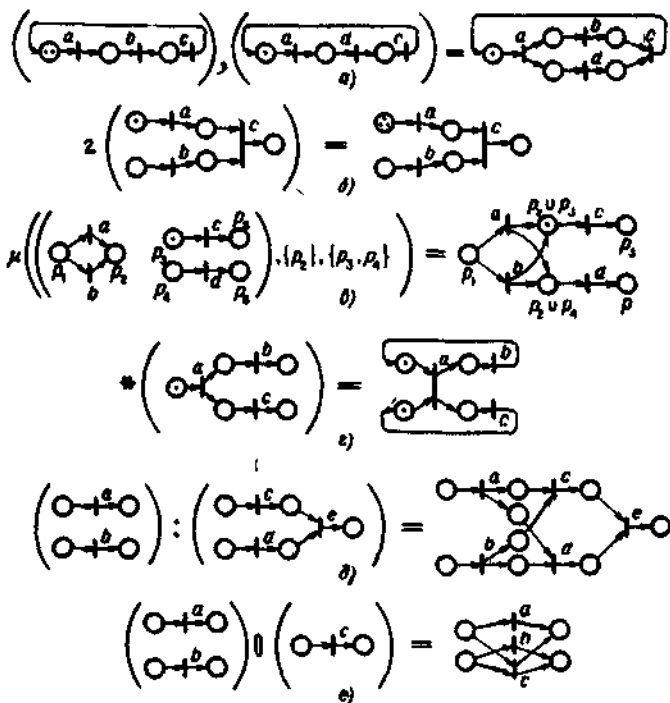


Рис. 1

По определению операции $h(N) = h(N_1) \cup h(N_2), l(N) = l(N_1) \cup l(N_2)$.

Операция разметки " n (\cdot)". Мы рассматриваем эту операцию как унарную. Она представляет собой, по существу, множество операций, задаваемых параметром n — целым неотрицательным числом или символом ω "бесконечного числа" фишек. Сеть $N' = n(N)$ совпадает как граф с сетью N , но имеет другую начальную разметку M'_0 , а именно:

$$M'_0(p) = \begin{cases} n + M_0(p), & \text{если } p \in h(N), \\ M_0(p) & \text{в противном случае.} \end{cases}$$

Пример выполнения операции разметки показан на рис. 1, б. Остальные операции над сетями определяются с помощью операции наложения и вспомогательной операции слияния мест в сети. Места, возникающие в результате слияния, представляют собой не символы, а множества символов, что требует некоторого простого обобщения

Кононюк А.Е. Графы

определения сети Петри: место идентифицируется не символом, а множеством символов (в том числе состоящим из единственного символа). Операция слияния мест состоит в свою очередь из двух подопераций: формирования слитых мест и замены сливаемых мест слитыми.

Мы ограничимся неформальным описанием этой операции. Пусть $X = (x_1, \dots, x_n)$ и $Y = (y_1, \dots, y_m)$ - два сливаемых множества мест сети $N = (P, T, F, M_0)$. Если X или Y — пустое множество, то результат слияния - исходная сеть N . Операция слияния μ множества мест X и Y в сети N строит новую сеть $N' = \mu(N, X, Y)$ в два этапа. Сначала каждое место $x_i \in X$ копируется в m экземплярах, где m - число мест в Y , а каждое место $y_j \in Y$ копируется в n экземплярах, где n - число мест в X . Места копируются вместе с их разметкой и инцидентными дугами. Затем каждая пара мест (x_i^j, y_j^i) заменяется новым местом $z = x_i \cup y_j$ с разметкой $M_0(z) = M_0(x_i) + M_0(y_j)$ и соответствующими инцидентными дугами. Пример выполнения операции слияния показан на рис. 1, в.

Продолжим определение операций над сетями.

Операция итерации ""*. Эта операция унарная, она сливает множество головных и множество хвостовых мест сети и применима к сети N , если множества головных и хвостовых мест не пересекаются:

$N' = *(N) = \mu(N, h(N), l(N))$, где $h(N) \cap l(N) = \emptyset$. По определению множества головных и хвостовых мест сети $N' = *(N)$ совпадают друг с другом и с множеством мест, полученных слиянием головных и хвостовых мест исходной сети. Пример выполнения операции итерации показан на рис. 1, г.

Операция присоединения ";". Эта операция соединяет две сети в одну, сливая множество хвостовых мест первой сети со множеством головных мест второй сети: $N = (N_1; N_2) = \mu(N_1, N_2, l(N_1), h(N_2))$. По определению головными местами сети являются "бывшие" головные места сети N_1 и те места, в образовании которых участвовали места из $h(N_1)$; аналогично назначаются хвостовые места сети N . Пример выполнения операции присоединения показан на рис. 1, д.

Операция исключения "||". Эта операция объединяет две сети N_1 и N_2 в одну сеть $N = (N_1 || N_2)$, сливая соответственно их головные и хвостовые места. По определению головные места сети N образованы местами, в которые вошли при слиянии мест головные места сетей N_1 и N_2 ; аналогично назначаются хвостовые места сети N . Пример выполнения операции исключения показан на рис. 1, е.

Пусть \mathcal{E} - класс элементарных сетей, т.е. класс символов-переходов. *Формула сети* в алгебре, порождаемой классом и введенными выше операциями, определяется следующим образом:

Кононюк А.Е. Графы

- (а) символ из E - формула,
 (б) если A - формула, то $n(A)$ и $*(A)$ - также формулы,
 (в) если A и B - формулы, то (A, B) , $\{A; B\}$ и $(A//B)$ - также формулы.
 Формулы в предлагаемой алгебре задают класс сетей Петри, которые мы будем называть *регулярными*. Например, формула

$$\{*(a; b) \} d\}, *(c; d) \} b\}$$

задает регулярную сеть, показанную на рис. 2.

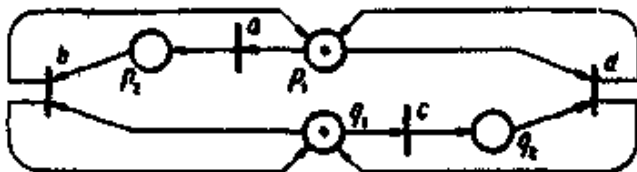


Рис 2

Одна и та же регулярная сеть может быть задана различными формулами, которые мы в этом случае считаем тождественными. Тожественные формулы могут быть трансформированы друг в друга с помощью конечной последовательности тождественных преобразований.

Пусть A, B, C — произвольные формулы, n_1 и n_2 — целые неотрицательные числа или символ ω .

(1) Коммутативность операций наложения и исключения:

П1. $(A, B) = (B, A)$.

П2. $(A \} B) = (B \} A)$.

(2) Ассоциативность бинарных операций:

П3. $((A, B), C) = (A, (B, C))$.

П4. $((A; B); C) = (A; (B; C))$.

П5. $((A \} B) \} C) = (A \} (B \} C))$

Для того чтобы сократить число скобок в формулах, будем учитывать ассоциативность операций, записывания формулы вида $((A, B), C)$ или $(A, (B, C))$ как (A, B, C) . Кроме того, будем опускать самые внешние скобки и введем следующее старшинство операций: унарные операции связывают сильнее, чем бинарные, а операции " ; " и " \} " старше чем операция " , ".

(3) Дистрибутивность операций:

Кононюк А.Е. Графы

П6. $(A, B); C = (A; C), (B; C).$

П7. $A; (B, C) = (A; B), (A; C).$

П8. $(A, B) \parallel C = (A \parallel C), (B \parallel C).$

П9. $*(A, B) = *A, *B, *(A; B), *(B; A).$

П10. $*(A \parallel B) = *A \parallel *B = *A \parallel B = A \parallel *B = (*A; *B).$

(4) Рефлексивность операции наложения.

П11. $(A, A) = A.$

(5) Свойства операции разметки:

П12. $n_1 (n_2 A) = n_2 (n_1 A) = (n_1 + n_2) A.$

П13. $n(*A) = *(nA).$

П14. $n(A, B) = (nA, nB).$

П15. $n(A; B) = (nA; B).$

П16. $n(A \parallel B) = (n_1 A \parallel n_2 B),$ где $n_1 + n_2 = n.$

Пример тождественных формул: сеть на рис. 2 задается как формулой, приведенной выше, так и следующими формулами:

$1 * (a, b) \parallel d, (c; d) \parallel b).$

$(* (1a; b) \parallel 1d, * (1c; d) \parallel * 1b).$

Формулу A из класса всех формул регулярных сетей назовем *расслоенной формулой*, если она имеет вид $(A_1, A_2, \dots, A_n), n \geq 2$, где A_i - формулы. Формулу A назовем *стандартно расслоенной*, если A_i - формулы не содержащие операции наложения ", ". Последние будем называть *примитивными формулами*, а задаваемые ими сети - также примитивными.

Теорема 1. *Класс формул регулярных сетей и его подклассы - класс расслоенных и класс стандартно расслоенных формул - тождественны в том смысле, что для любой формулы существуют тождественные ей расслоенная и стандартно расслоенная формулы.*

Доказательство. Если A - примитивная формула, то она тривиально преобразуется в расслоенную с помощью П11: $A = (A, A)$. Если A - не примитивная или расслоенная формула, то существует ее подформула, которая не является ни примитивной, ни расслоенной и имеет вид:

$((B, C); D), (B; (C, D)),$

$((B, C) \parallel D), (B \parallel (C, D)),$

$* (B, C), n(B, C).$

Применяя подходящее из преобразований П6, П7, П8, П9, П14, эту формулу можно трансформировать соответственно к одной из следующих форм:

Кононюк А.Е. Графы

$(B; D, C; D),$

$(B; C, B; D),$

$(B \parallel D, C \parallel D),$

$(B \parallel C, B \parallel D),$

$\{ * B, * C, * (B; C), * (C; B) \},$

$(nB, nC).$

Таким образом, подформула преобразована в тождественную расслоенную формулу. Поступая аналогично с теми подформулами новой формулы A' , которые не являются ни примитивными, ни расслоенными, и затем продолжая этот процесс до тех пор, пока не будет получена расслоенная формула или пока все подформулы не станут примитивными или стандартно расслоенными, можно получить расслоенную или стандартно расслоенную формулу, тождественную исходной формуле.

2.10.2. Некоторые свойства регулярных сетей

Теорема 1 говорит о том, что любая непримитивная регулярная сеть может быть получена из более простых сетей (в том числе — из примитивных) с помощью операции наложения. Такое расслоение регулярной сети позволяет расчленить процесс ее анализа или конструирования на отдельные этапы, на которых выступают сети с более простыми свойствами.

В частности, примитивные сети весьма похожи по своим свойствам на автоматные, но не совпадают с ними. В примитивной сети каждый переход имеет ровно одно входное и одно выходное место, сеть консервативна. Графовая структура примитивных сетей более ограничена, чем у автоматных сетей, за счет ее регуляризации, получаемой при алгебраическом способе конструирования с помощью операций присоединения, исключения и итерации.

С другой стороны, примитивная сеть может содержать места с разметкой ω , поэтому она, в отличие от автоматной сети, может иметь бесконечное множество достижимых разметок и, следовательно, не моделируется конечным автоматом. В силу этого примитивная сеть может быть неограниченной, причем только в том случае, если содержит места с начальной разметкой ω .

Примитивная сеть имеет единственное головное и единственное хвостовое место, она жива, если и только если:

- она представляет собой сильно связный граф и ее начальная разметка содержит хотя бы одну фишку, или
- начальная разметка головного места равна ω .

При поэтапном анализе регулярных сетей, последние удобно задавать с помощью (стандартно) расслоенных формул, затем выяснить свойства составляющих сетей, и зная, каким образом операция

Кононюк А.Е. Графы

наложения формирует свойства результирующей сети, по свойствам составляющих сетей устанавливать свойства анализируемой сети. Для этого рассмотрим вопрос о том, как связаны между собой свойства составляющих и результирующей сетей.

Пусть $N = (N_1, N_2)$. Если $P_1 \cap P_2 \neq \emptyset$, то преобразуем сеть N_2 в сеть N'_2 , заменив в сети N_2 каждое место $p \in P_1 \cap P_2$ на новое место \bar{p} ,

$\bar{p} \notin P_1 \cup P_2$ и сохранив для него начальную разметку места p . Сеть

$\bar{N} = (N'_1, N'_2)$ будем называть *макетом* сети N . Отметим, что формально сети N'_2 и N не являются регулярными, так как содержат места, которые не конструируются по правилам образования регулярных сетей. Более того, макет не удовлетворяет условию АЗ

определения сети вообще, так как может содержать пару мест p и \bar{p} , которые инцидентны одному и тому же множеству переходов. Но поскольку макет является вспомогательной конструкцией, мы допускаем такое отклонение от принятого определения сети. Свойства сетей N и \bar{N} весьма близки друг к другу. В частности, если

$P_1 \cap P_2 = \emptyset$, то $N = \bar{N}$.

На рис. 3 показан пример двух сетей N_1 и N_2 , сеть $N = (N_1, N_2)$ и ее макет \bar{N} .

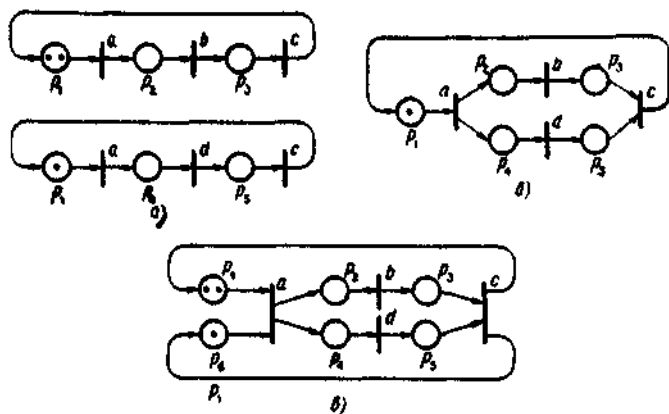


Рис. 3

Пусть множества мест P_1 и P_2 сетей N_1 и N_2 упорядочены каким-либо образом. При переименовании мест из $P_1 \cap P_2$ в сети N_2 новые места

Кононюк А.Е. Графы

получают порядковые номера заменяемых мест. Множество мест \bar{P} макета \bar{N} сети $N=(N_1, N_2)$ упорядочивается так, что сначала следуют места из P_1 согласно их порядку в P_1 , а затем места из P_2 , сохраняющие свой порядок. Пусть $M \in R(N), \bar{M} \in R(\bar{N})$, где $R(N)$ и $R(\bar{N})$ — множества достижимых разметок сетей N и \bar{N} ; обозначим через M' и \bar{M}' проекции разметок M и \bar{M} на множество мест P_i , где $i=1,2$. При векторном представлении разметок $\bar{M}_0 = M_{01} \circ M_{02}$, т.е. вектор начальной разметки сети \bar{N} — это конкатенация векторов начальных разметок сетей N_1 и N_2 .

Лемма 1. $R(\bar{N}) \subseteq R(N_1) \circ R(N_2)$.

Доказательство. Учитывая, что $\bar{M}_0 = M_{01} \circ M_{02}$, получим

$\bar{M} = M_1 \circ M_2$ и $M_1 \in R(N_1), M_2 \in R(N_2)$. Покажем, что

непосредственно следующая за \bar{M} в результате срабатывания некоторого перехода t разметка \bar{M}' также представляет собой конкатенацию разметок из $R(N_1)$ и $R(N_2)$. Действительно, если $t \in T_1 \setminus T_2$, то в сети N его входные и выходные места — это места из P_1 ,

поэтому $\bar{M}' = M'_1 \circ M_2$, где M'_1 — это разметка из $R(N_1)$ такая, что $M'_1[t > M_1$ в сети N_1 . Аналогично, при $t \in T_2 \setminus T_1$ новая разметка $\bar{M}' = M_1 \circ M'_2$, где $M'_2 \in R(N_2)$ и $M_2[t > M'_2$ в сети N_2 .

Наконец, если $t \in T_1 \cap T_2$, то его входные места делятся на две независимые группы: места из P_1 и места из P'_2 (т.е. из модифицированной сети N'_2). Поэтому $\bar{M}' = M'_1 \circ M'_2$, где $M'_1 \in R(N_1), M'_2 \in R(N'_2), M_1[t > M'_1$

в сети $N_1, M_2[t > M'_2$ в сети N_2 .

Таким образом, в каждом из рассмотренных случаев разметка \bar{M}' такая, что $\bar{M}'[t > \bar{M}$ в сети \bar{N} , представляет собой конкатенацию разметок из $R(N_1)$ и $R(N_2)$. Начиная от начальной разметки \bar{M}_0 , убеждаемся, что любая достижимая в N разметка представляет собой такую конкатенацию.

Лемма 2. Для сети $N=(N_1, N_2)$ множество достижимых разметок $R(N)$ состоит из всех тех и только тех разметок, которые образованы из разметок множества $R(\bar{N})$ по следующему правилу:

Кононюк А.Е. Графы

$$M(p) = \begin{cases} \bar{M}(p), & \text{если } p \in P_1 \setminus P_2 \cup P_2 \setminus P_1, \\ \min(\bar{M}(p), \bar{M}(\bar{p})), & \text{если } p \in P_1 \cap P_2. \end{cases}$$

$$\forall p \in P = P_1 \cup P_2;$$

Справедливость этой леммы непосредственно следует из определения операции наложения и определения макета \bar{N} сети N . Заметим, в частности, что $M^i_0 \leq \bar{M}^i_0 = M_{0i}$ где $i=1$ или 2 .

Хотя мы рассматривали сеть $N=(N_1, N_2)$, составленную наложением двух сетей, ясно, что понятие макета сети и леммы 1 и 2 естественным образом обобщаются на случай, когда $N=(N_1, N_2, \dots, N_n)$.

Теорема 2. Сеть $N=(N_1, N_2, \dots, N_n)$ ограничена, если ограничены сети N_1, N_2, \dots, N_n .

Доказательство. Если сети N_1, N_2, \dots, N_n ограничены, то множества $R(N_1), R(N_2), \dots, R(N_n)$ содержат только конечные разметки. В силу леммы 1 таким же будет множество $R(\bar{N})$, а в силу леммы 2 множество $R(N)$ содержит только конечные разметки, т.е. сеть N ограничена.

Лемма 3. В сети $N=(N_1, N_2)$ разметка M' достижима от разметки M и переход t достижим от M , если и только если разметка \bar{M}' , из которой образована M' , и переход t достижимы в \bar{N} от \bar{M} , из которой образована M .

Доказательство. Для любого перехода $t \in T = T_1 \cup T_2 = \bar{T}$ множество разметок $M_t^+ \subseteq R(N)$, при которых он может сработать в N , - это те и только те разметки, которые образованы (по правилу лемме 2) из аналогичного множества $M_t^+ \subseteq R(\bar{N})$. Действительно, если t может сработать при разметке $\bar{M} = M'_1 \circ M_2$ в макете \bar{N} , то

$$\forall p \in P = P_1 \cup P_2: M(p) - F(p, t) \geq 0.$$

Отсюда следует, что

$$\forall p \in P_i: M_i(p) - F_i(p, t) \geq 0,$$

где $i = 1, 2$. С учетом леммы 2 получаем:

$$\forall p \in P: M(p) - F(p, t) \geq 0.$$

С другой стороны, если t не может сработать при разметке \bar{M} в \bar{N} , то он не сможет сработать при M в N , так как $\exists p \in P: \bar{M}(p) - \bar{F}(p, t) < 0$ и, с учетом леммы 2, для того же p верно, что $M(p) - F(p, t) < 0$.

В результате любой последовательности

$$\bar{M}(t_1) \bar{M}(t_2) \dots \bar{M}(t_k)$$

Кононюк А.Е. Графы

при функционировании сети \bar{N} соответствует последовательность $M\langle t_1 \rangle M''\langle t_2 \rangle \dots \langle t_k \rangle M'$ при функционировании сети N , что гарантирует справедливость утверждения леммы.

Из лемм 2 и 3 непосредственно следует

Лемма 4.

(а) Сеть $N=(N_1, N_2)$ эквивалентна макету \bar{N} .

(б) Сеть N жива тогда и только тогда, когда жив макет \bar{N} .

Пусть $G(N_1), G(N_2)$ - графы разметок сетей $N_1, N_2, G(N)$ и $G(\bar{N})$ - графы разметок сетей $N=(N_1, N_2)$ и \bar{N} . Граф $G(\bar{N})$ связан с графами $G(N_1)$ и $G(N_2)$ следующим образом. Пусть вспомогательный граф $G^0(\bar{N})$ - это граф со множеством вершин $R(\bar{N}) = R(N_1) \circ R(N_2)$.

Две вершины \bar{M} и \bar{M}' соединены в этом графе i -дугой ($i = 1, 2$), помеченной символом перехода t , если и только если $\bar{M} = M_1 \circ M_2, \bar{M}' = M'_1 \circ M'_2$, вершины M_i и M'_i соединены в

$G(N_i)$ дугой с символом t . Если $t \in T_1 \setminus T_2$, в графе $G^0(\bar{N})$ из вершины \bar{M} в вершину \bar{M}' ведут две дуги с символом t : 1.-дуга и 2.-дуга (эти дуги будем называть парными). Граф $G(\bar{N})$ отличается от графа $G(N)$ тем, что в $G(\bar{N})$ отсутствуют:

(а) все непарные дуги, помеченные символами из $T_1 \cap T_2$,

(б) для каждой удаленной i -дуги - все вершины (вместе с исходящими дугами), на которые нельзя прийти из \bar{M} по i -путям, где i -путь — это путь по i -дугам и по тем i -дугам, которые соединяют две соседние вершины-разметки с одной и той же i -частью разметки ($\bar{i} = 1$, если $i = 2$, и $\bar{i} = 2$, если $i = 1$).

(Определение графа $G(\bar{N})$ неэффективно, так как все рассматриваемые графы разметок и граф $G^0(N)$ могут быть бесконечными.)

На рис. 4 показаны две (примитивные) сети $N_1 = * 1(a; b) [d]$ и $N_2 = * 1(c; d) [b]$.

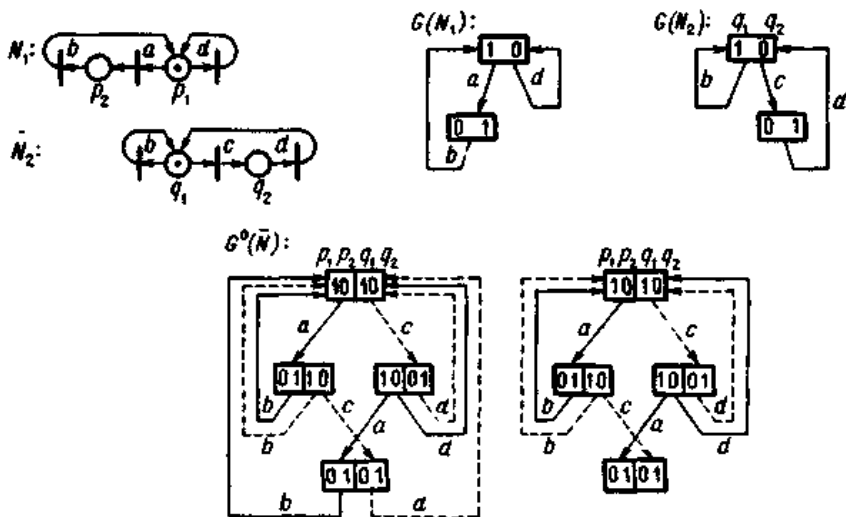


Рис. 4

Сеть $N = (N_1, N_2)$ — это сеть на рис. 2. Заметим, что N совпадает с \bar{N} , так как $P_1 \cap P_2 = \emptyset$. На этом же рисунке изображены графы $G(N_1)$, $G(N_2)$, $G^0(\bar{N})$, $G(\bar{N})$.

Лемма 5. Если разметка $\bar{M}' = M'_1 \circ M'_2$ достижима от разметки $\bar{M} = M_1 \circ M_2$ в сети \bar{N} , то разметка M'_i достижима от разметки M_i в сети N_i , $i=1,2$.

Доказательство. Если \bar{M}' достижима от \bar{M} в N , то в $G(N)$ есть путь $(\bar{M}, \bar{M}', \dots, \bar{M}')$. Каждое "звено" этого пути включает пару смежных вершин-разметок и соединяющую их дугу. По лемме 1 в $G(N_i)$ есть пара вершин, которые являются проекциями смежных вершин звена. Если в $G(\bar{N})$ соединяющие дуги — это i -дуга с символом t , то, по определению графа $G(\bar{N})$, в $G(N_i)$ соответствующие вершины соединены дугой с тем же символом. Если эта дуга — i -дуга, то проекции смежных вершин на N_i совпадают (см. доказательство леммы 1) и им соответствует одна вершина в $G(N_i)$. Поэтому указанному выше пути в $G(\bar{N})$ соответствует путь в $G(N_i)$ от M_i к M'_i , что и доказывает утверждение леммы. Следствием лемм 3 и 5 является

Кононюк А.Е. Графы

Теорема 3. Если в сети $N = (N_1, N_2)$ разметка M' достижима от разметки M и переход $t \in T$ достижим от M , то в сети $N_i, i=1,2$, разметка M^i и переход t достижимы от M_i .

Как в случае лемм 1 и 2, леммы 3, 4 и теорема 3 обобщаются на случай наложения нескольких сетей.

Базовым фрагментом V_p сети Петри N , где $p \in P$, называют подсеть, включающую место p , все дуги, инцидентные p , и все переходы из

$\cdot p \cup p \cdot$. Например, для сети, показанной на рис. 5, а, все ее базовые фрагменты изображены на рис. 5, б.

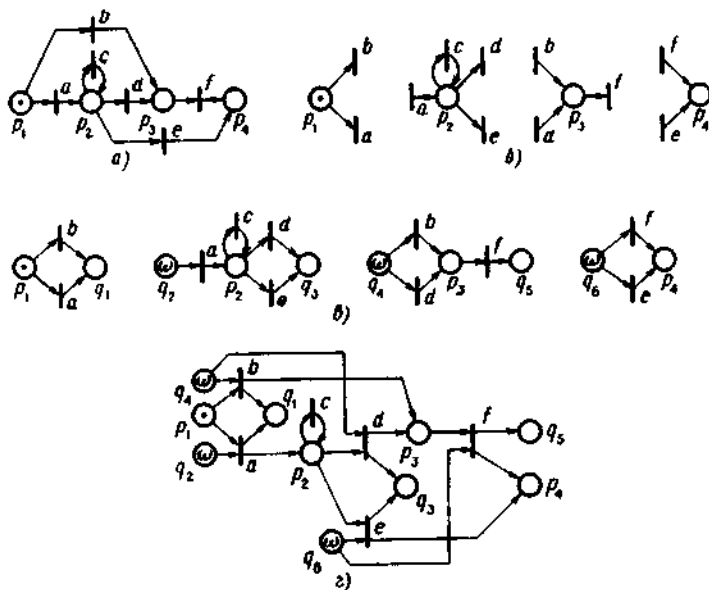


Рис. 5.

Регуляризованный базовый фрагмент (р.б.ф.) — это фрагмент, который получается из базового фрагмента следующим образом:

(а) если $\cdot p \setminus p \cdot \neq \emptyset$, то вводится новое место, которое объявляется головным местом р.б.ф., его разметка полагается равной ω и из него заводятся дуги на каждый переход из $\cdot p \setminus p \cdot$;

Кононюк А.Е. Графы

(б) если $p \setminus p \neq \emptyset$, то вводится новое место, которое объявляется хвостовым местом р.б.ф., его разметка полагается равной 0 и на это место заводятся дуги от каждого перехода из $p \setminus p$;

(в) если $p \setminus p = \emptyset$, то p - головное место р.б.ф.;

(г) если $p \setminus p = \emptyset$, то p — хвостовое место р.б.ф.

На рис. 5, в показаны регуляризованные базовые фрагменты, полученные из базовых фрагментов на рис. 5, б. Р.б.ф. является регулярной сетью и р.б.ф. эквивалентен базовому фрагменту, из которого он получен.

Заметим, что, если $V_{p_1}, V_{p_2}, \dots, V_{p_n}$ - базовые фрагменты сети N и

$(p_1, p_2, \dots, p_n) = P$, то $(V_{p_1}, V_{p_2}, \dots, V_{p_n}) = N$, т.е. сеть N - результат наложения друг на друга всех базовых фрагментов.

Теорема 4. *Класс сетей Петри и его подкласс регулярных сетей равномошен, т.е. любую сеть Петри можно преобразовать в эквивалентную регулярную сеть.*

Доказательство. Алгоритм преобразования (регуляризации) сети состоит из трех этапов:

(1) в заданной сети Петри выделяются все базовые фрагменты

$$V_{p_1}, V_{p_2}, \dots, V_{p_n};$$

(2) каждый базовый фрагмент V_p преобразуется в регуляризованный базовый фрагмент U_p ;

(3) формируется регулярная сеть наложением друг на друга всех полученных р.б.ф., т.е.

$$N' = (U_{p_1}, U_{p_2}, \dots, U_{p_n}).$$

На рис. 5, а показана сеть Петри, которая не является регулярной, так как не существует задающей ее формулы. На рис. 5, б - все ее базовые фрагменты, на рис. 5, в - соответствующие р.б.ф., на рис. 5, г - регулярная сеть, эквивалентная исходной сети.

Построенная сеть N' эквивалентна исходной сети N . Действительно, сеть N' отличается от сети N лишь тем, что в нее добавлены новые места и дуги, соединяющие их с переходами сети N . Ясно, что те места, которые введены как хвостовые места р.б.ф., не изменяют язык сети, так как из них не исходит ни одной дуги. Те места, которые введены как головные места р.б.ф., имеют разметку ω , и поэтому их добавление к входным местам переходов сети N также не меняет язык.

Алгебраизация сетей и особенно расслоение их на более простые сети позволяют более последовательно решать проблему живости, переходя от частных случаев сетей ко все более сложным.

Кононюк А.Е. Графы

Иллюстрацией сказанного могут служить следующие две простые теоремы.

Теорема 5. *Регулярная сеть $N=(N_1, \dots, N_n)$ не жива, если хотя бы одна из составляющих сетей N_1, \dots, N_n не жива.*

Доказательство. Предположим противное - сеть N жива, но одна из составляющих сетей N_i не жива, например, за счет того, что некоторый переход t не достижим от некоторой разметки $M^i \in R(N_i)$. Так как N жива, то переход t достижим от любой разметки $M \in R(N)$. На основании лемм 1, 2 и теоремы 3 этот переход должен быть достижимым в N_i от любой разметки $M^i \in R(N_i)$, что противоречит предположению. \square

Теорема 5 устанавливает только необходимые условия живости регулярных сетей. Из того факта, что все составляющие сети L_1, \dots, L_n живы,

не следует, что и результирующая сеть $N=(N_1, \dots, N_n)$ будет живой.

Например, сети N_1 и N_2 на рис. 4 живы, но не является живой сеть

$N=(N_1, N_2)$ (рис. 2). Легко устанавливается следующий факт.

Теорема 6. *Регулярная сеть $N=(N_1, N_2)$ жива, если N_1 и N_2 живы, и пересечение их множеств переходов содержит не более одного перехода.*

Регулярную сеть $N=(P, T, F, M_0)$ назовем *конечноразмеченной*, если

$$\forall p \in P, M_0(p) \neq \omega.$$

Теорема 7. *Конечноразмеченная регулярная сеть ограничена.*

Доказательство. Пусть конечноразмеченная регулярная сеть N представлена стандартно расслоенной формулой (N_1, \dots, N_n) , Возможны два случая.

Если ни одна из сетей N_1, \dots, N_n не содержит мест с ω , то каждая из них ограничена и, по теореме 2, сеть N также ограничена.

Если некоторая составляющая сеть N_i содержит место p с разметкой $M^i_0(p) = \omega$, то тогда имеется в другой составляющей сети N_j место p с разметкой $M^j_0(p) = k \neq \omega$. В противном случае разметка ω проникла бы в сеть N . Заменяем в сети N_i разметку M^i_0 на разметку $M^i'_0$ так, чтобы $M^i'_0(p) = k$.

Прделаем аналогичные изменения так, чтобы ни одна из составляющих сетей не содержала мест с разметкой ω . В силу определения операции наложения сеть окажется представленной новой стандартно расслоенной формулой (N'_1, \dots, N'_n) такой, что ни одна из составляющих сетей не содержит мест с разметкой ω . Эта формула эквивалентна предыдущей, а все составляющие сети ограничены. Следовательно, сеть N также ограничена.

Теорема 8. *Проблема живости разрешима в классе конечноразмеченных регулярных сетей.*

Кононюк А.Е. Графы

Доказательство. Пусть N - конечноразмеченная сеть, $G(N)$ - ее граф разметки. Для любой вершины M из графа $G(N)$ выпишем множество $T(M)$ переходов, метящих все дуги, исходящие из M . Множество $T(M)$ содержит те переходы, которые могут сработать в N при разметке $M \in R(N)$.

Сеть N жива тогда и только тогда, когда для любого перехода $t \in T$ и для любой вершины M в $G(N)$ выполнено следующее условие: если $t \in T(M)$, то в $G(N)$ существует путь из M в некоторую вершину M' такую, что $t \in T(M')$.

В силу теоремы 6 сеть N ограничена, и, следовательно, граф $G(N)$ конечен. Поэтому можно непосредственно убедиться, выполнено ли для N сформулированное выше необходимое и достаточное условие живости.

2.10.3. Иерархические сети

Иерархические сети являются обобщением регулярных сетей и служат для моделирования иерархических систем, которые, наряду с неделимыми, атомарными компонентами, содержат составные компоненты, сами представляющие собой системы (быть может, также иерархические).

Для определения иерархических сетей и формул класс элементарных формул, т.е. класс символов переходов, разбивается на два непересекающихся подкласса: *терминальные* символы и *нетерминальные* символы. Соответственно переходы разделяются на *простые* и *составные*.

Терминальные символы и простые переходы трактуется точно так же, как и в предыдущей главе. Нетерминальные символы обозначают "элементарную" сеть, состоящую из единственного составного перехода (с головным и хвостовым местом), который имеет внутреннюю структуру и является по существу самостоятельной (иерархической) сетью.

Иерархическая сеть - это сеть, задаваемая *структурной формулой*. Последняя представляет собой формулу сети, которая строится из терминальных и нетерминальных символов с помощью операций алгебры регулярных сетей и упорядоченного множества *определений нетерминальных символов*.

Каждое определение имеет вид $s: A$, где s - нетерминальный символ, A - формула сети. Следующая структурная формула определяет иерархическую сеть, показанную на рис. 6:

$(1 \circ (u; (v, w)), 1 \circ (f[h]))$

$u : (1(a, b); c)$

$v : (1 \circ h; k)$

$w : ((2x, 4d; e), (2f; e))$

$x : (2f; g)$.

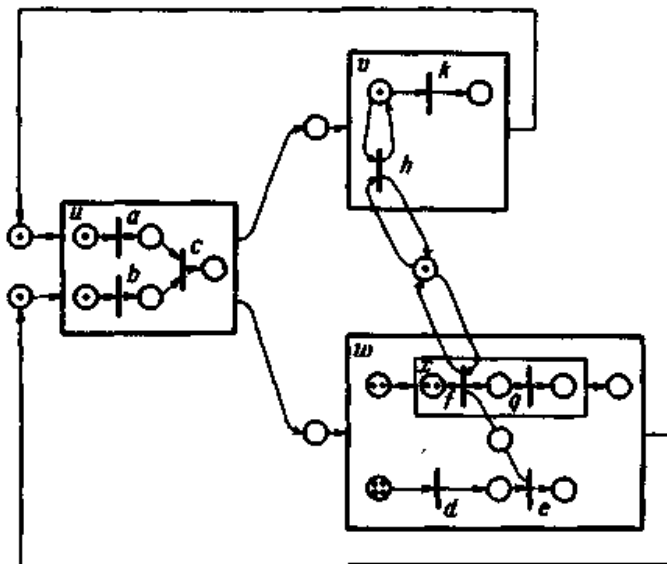


Рис. 6.

Дополнительные контекстные ограничения состоят в следующем.

(1) Любой символ, входящий в структурную формулу и не определяемый в ней, является терминальным. Он обозначает простой переход, как и символы a, b, c, d, e, f, k на рис. 6.

(2) Каждый нетерминальный символ определяется один раз и не может входить в формулу в правой части своего и последующих определений.

Нетерминальные символы u, v, w, x на рис. 6 обозначают составные переходы, изображаемые прямоугольниками, внутри которых содержатся сети, задаваемые формулами в правых частях определений. Переход t является *внутренним* переходом перехода t' , а переход t' является *объемлющим* переходом для перехода t , если:

(1) t содержится в правой части определения перехода t' ,

Кононюк А.Е. Графы

(2) существует цепочка переходов t_1, t_2, \dots, t_k такая, что $t_1 = t', t_k = t$ и для любого i , где $2 \leq i \leq k$, переход t_i является внутренним переходом перехода t_{i-1} .

К приведенным выше ограничениям добавляется еще одно:

(3) Любые два определения нетерминальных символом могут содержать в своих правых частях одинаковые символы, если только они задают переходы, один из которых является внутренним по отношению к другому. Это ограничение гарантирует "хорошую вложенность" составных переходов в иерархической сети.

Переход t' *охватывает* переход t , если t является внутренним переходом t' и не существует в сети перехода t'' такого, что t' является внутренним переходом t'' и объемлющим для t . Переход t — охватываемый переходом t' .

Переходы называются *соседями*, если они охватываются одним и тем же переходом или не имеют охватывающих переходов. В последнем случае они называются *переходами верхнего уровня*. Место p является *локальным* входным или выходным местом перехода t , если все переходы, связанные с p , являются соседями t или внутренними переходами соседей перехода t . В противном случае, место p является *внешним* входным или выходным местом перехода t .

Место p является *внутренним* местом t , если оно связано только с внутренними переходами перехода t .

Место p является *сторонним* для перехода t , если оно не является его внутренним местом, но в t есть внутренний переход, связанный с местом p .

На рис. 6 переход f является внутренним для переходов w и x переход x охватывает f . Переходы a , b и c являются соседями. Месте $\{\bar{f}\}$ - локальное входное место перехода f и внутреннее место в x , место $\{\bar{f}, \bar{e}\}$ является сторонним для x и внешним выходным местом перехода f .

Иерархическая сеть функционирует, переходя от разметки к разметке, как и регулярная сеть, но правила функционирования иерархической сети отличаются от соответствующих правил для регулярной сети. Эти различия вызваны наличием составных переходов, срабатывание которых является не мгновенным событием, как в сетях Петри, а составным действием. Поэтому целесообразно говорить не о срабатывании составного перехода, а о его работе. Если связать с функционированием сети (дискретное) время, то можно говорить о том, что составной переход может находиться в одном из двух состояний - *пассивном* и *активном*. Смена пассивного состояния на активное, или *активация* составного перехода, и смена активного

Кононюк А.Е. Графы

состояния на пассивное, или *завершение*, являются мгновенными событиями. Начальное состояние всех переходов - пассивное. Будем считать, что простые переходы также могут быть активны, но их активность мгновенна и активация совпадает с завершением.

Переход t может быть активирован, если:

- (1) он пассивен,
- (2) охватывающий его переход активен или t является переходом верхнего уровня,
- (3) $\forall p \in P, M(p) \geq F(p, t)$, где P множество всех мест в сети.

Когда составной переход t активируется, происходит смена текущей разметки M сети на разметку M' по правилу

$$\forall p \in P: M'(p) = M(p) - F(p, t).$$

Условием завершения составного перехода t служит тот факт, что все его внутренние переходы пассивны и ни один из них не может быть активирован. Когда переход t завершен, происходит смена текущей разметки M на разметку M' по правилу

$$\forall p \in P_t: M'(p) = M_0(p).$$

$$\forall p \in P \setminus P_t: M'(p) = M(p) + F(t, p),$$

где P_t - множество внутренних мест перехода t , P - множество всех мест сети.

Другими словами, при завершении составного перехода восстанавливается начальная разметка его внутренних мест, а его выходные места получают дополнительные фишки.

Пусть каждому составному переходу t сопоставлены два символа, символ \bar{t} активации перехода и символ $\bar{\bar{t}}$ завершения перехода. Пусть также в иерархической сети множество $\overline{TT} = T_s \cup \bar{T} \cup \bar{\bar{T}}$, где T_s — множество простых переходов, \bar{T} — множество символов активации, $\bar{\bar{T}}$ — множество символов завершений составных переходов.

В процессе функционирования иерархической сети разметка M может смениться на разметку M' в результате:

- (1) срабатывания простого перехода t , т.е. $M[\bar{t}] > M'$,
- (2) активирования составного перехода t , т.е. $M[\bar{\bar{t}}] > M'$,
- (3) завершения составного перехода t , т.е. $M[\bar{\bar{\bar{t}}}] > M'$.

Разметка M' *достижима* в иерархической сети от разметки M в результате последовательности срабатываний $\tau = t_1, \dots, t_k$, где $t_i \in \overline{TT}$, если существует последовательность разметок $M[t_1]M_1[t_2] \dots [t_k]M'$.

Определения достижимых переходов, ограниченности сети и ее живости переносятся естественным образом на иерархические сети.

При определении свободных языков иерархических сетей появляются варианты.

Полным свободным языком $L(N)$ иерархической сети N назовем множество всех последовательностей $t \in TT^*$, таких, что существует достижимая в N разметка M , причем $M_o \uparrow \tau > M$.

Свободным языком $L_i(N)$ иерархической сети N будем считать проекцию языка $L(N)$ на множество T_s т.е. множество последовательностей, полученных удалением символов активации и завершения составных переходов в последовательностях из $L(N)$.

Помеченная иерархическая сеть имеет дополнительно (частичную) помечающую функцию $\Sigma: T_s \rightarrow A$, которая метит только простые переходы. Точно так же, как ранее, определяются префиксный и терминальный языки помеченной иерархической сети.

2.10.4. Сравнение иерархических сетей с другими классами сетей

В этом параграфе будет проведено сравнение выразительной мощности класса иерархических сетей, класса сетей Петри и их обобщений, рассмотренных ранее. Будет показано, что иерархические сети обладают такими же моделирующими способностями, что и счетчиковый автомат, и, следовательно, иерархические сети являются существенным обобщением сетей Петри (и регулярных сетей).

Теорема 9. *Любой рекурсивно перечислимый язык может быть порожден помеченной иерархической сетью.*

Доказательство. Достаточно продемонстрировать, как в иерархических сетях можно промоделировать оператор условного вычитания единицы счетчикового автомата. На рис. 7 показан фрагмент сети, моделирующий оператор если $x_i \neq 0$ то $x_i := x_i - 1$.

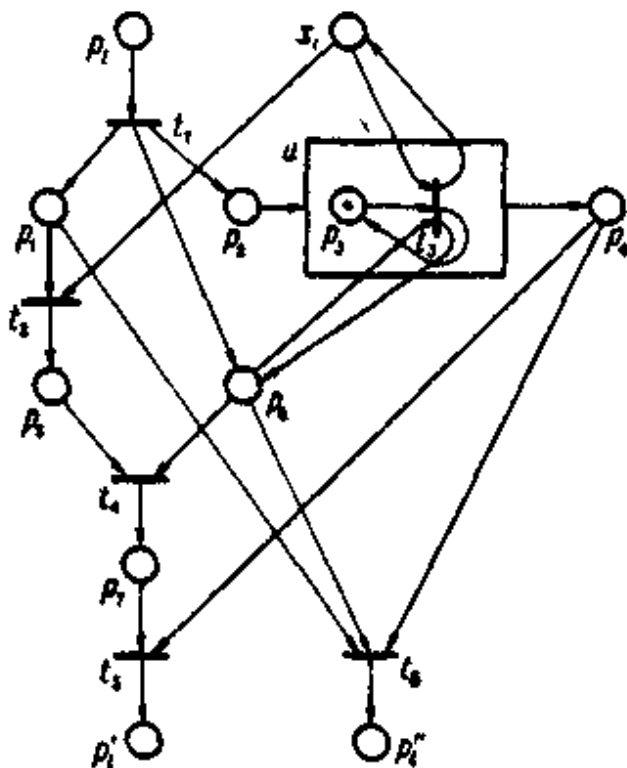


Рис. 7

Место x_i соответствует счетчику x_i , место p_1 принимает "управляющую" фишку от фрагмента, соответствующего предыдущему оператору, места p_1' и p_1'' соответствуют альтернативам $x_i \neq 0$ или, $x_i = 0$, ровно одно из них должно получить фишку в результате работы фрагмента. После того как в место p_i поступает фишка, срабатывает переход t_1 , посылая по фишке в три места p_1, p_2, p_6 . Если место x_i не содержит фишек, то переход t_2 не может сработать, в результате чего не смогут сработать ни t_4 , ни t_5 , т.е. p_1' не получит фишки. Составной переход u в этом случае является единственным переходом, который может сработать. После его активации начинает работать внутренняя сеть, содержащая единственный переход t_3 . Если x_i не содержит фишек, то переход t_3 не может сработать и составной

переход u завершается, посылая фишку в место p_4 . После этого может сработать только переход t_6 , который направляет фишку в место p_1 .

Если место x_i содержит фишки, то могут сработать как переход t_2 , так и составной переход u . Срабатывание t_2 посылает фишку в место p_5 , после чего может сработать переход t_4 , забирающий фишки из мест p_5 , p_6 и посылающий фишку в p_7 . Параллельно с последовательностью срабатываний переходов t_2 , t_4 активируется составной переход u . Если он начал работать раньше, чем переход t_2 , то его внутренняя сеть функционирует в результате циклического повторения срабатываний внутреннего перехода t_3 . Переход u закончит работу или после того, как t_2 заберет единственную фишку из места x_1 , или после того, как переход t_4 заберет фишку из места p_6 . После завершения составного перехода место p_4 получает фишку, которая позволяет сработать переходу t_5 , посылающему фишку в место p_1 . Переход t_6 не может сработать, так как в этом случае места p_1 , p_6 , p_4 никогда не могут иметь одновременно фишки. Таким образом, фрагмент иерархической сети на рис. 7 моделирует оператор условного вычитания единицы.

Из теоремы 9 следует, что класс иерархических сетей равномогчен классам ингибиторных сетей и сетей с приоритетами, введенным ранее. Более того, сети любого из этих трех классов эффективно преобразуются в эквивалентные сети из двух других классов. Покажем, например, что иерархическую сеть можно трансформировать в сеть с приоритетами, которая λ -эквивалентна исходной сети в следующем смысле. Сравниваются свободный язык иерархической сети и язык помеченной сети с приоритетами (или сети другого типа), причем помечающая функция для последней сети имеет вид $\sum(t) = t$ или $\sum(t) = \lambda$, т.е. помечающая функция является частичной. Другими словами, так помеченная сеть отличается от непомеченной лишь наличием дополнительных λ -переходов, символы которых не используются при порождении языка сети.

Теорема 10. *Для любой иерархической сети можно построить λ -эквивалентную ей сеть с приоритетами.*

Доказательство. Алгоритм преобразования иерархической сети в сеть с приоритетами состоит в последовательной замене составных переходов, начиная с таких, у которых все внутренние переходы простые, на фрагменты сети с приоритетами.

Рассмотрим сначала частный случай иерархической сети, которая представляет собой единственный составной переход, все внутренние переходы которого - простые. Пусть T_t - множество внутренних переходов перехода t , P_t — множество его внутренних мест, M_0 — начальная разметка сети. Преобразование будет состоять в "раскрытии" составного перехода.

Кононюк А.Е. Графы

Для построения λ -эквивалентного фрагмента сети с приоритетами копируется "внутренняя сеть" перехода t и его входные и выходные места. К копии добавляются следующие места и λ -переходы (см. рис. 8).

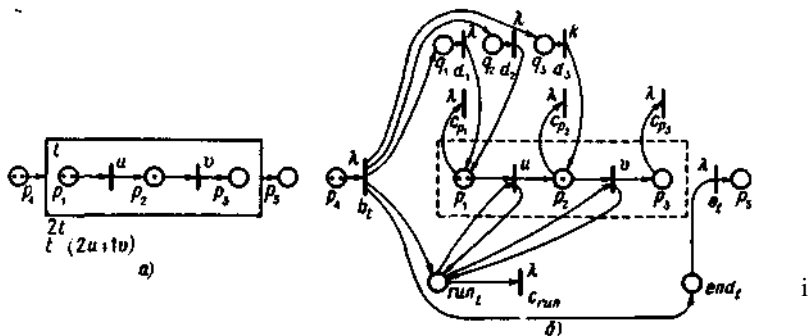


Рис. 8

- 1) λ -переход b_t с двумя выходными местами run_t и end_t . Этот переход в сети с приоритетами будет менять разметку входных мест перехода t и активировать внутреннюю сеть.
- 2) Множество λ -переходов $C_t = \{c_{run}, c_{p_1}, \dots, c_{p_n}\}$, где p_1, \dots, p_n — места из P_t . Эти переходы начнут работать тогда, когда ни один из внутренних переходов составного перехода t не сможет сработать, и "очистят" внутренние места от фишек.
- 3) Множество λ -переходов $D_t = \{d_1, d_2, \dots, d_m\}$ с входными местами q_1, q_2, \dots, q_m . Место q_i получает фишки от перехода b_t , с которым оно связано дугой. Эти λ -переходы будут восстанавливать начальную разметку внутренних мест после завершения работы внутренней сети.
- 4) λ -переход e_t , входным местом которого служит место end_t . Этот переход будет менять разметку выходных мест перехода t и инициировать восстановление разметки во внутренней сети.

Для введенных переходов и простых переходов в t вводятся приоритеты следующим образом:

$$\begin{aligned} pr(t_1) = \dots = pr(t_k) &> pr(c_{run}) > pr(c_{p_1}) = \dots = pr(c_{p_n}) > \\ &> pr(d_1) = \dots = pr(d_m) > pr(e_t) > pr(b_t), \end{aligned}$$

где $t_1, \dots, t_k \in T_t$.

Введенные X-переходы и места связываются дугами с простыми переходами и местами иерархической сети следующим образом.

- (1) Каждое входное место из t соединяется дугой с λ -переходом b_t .

Кононюк А.Е. Графы

- (2) Место гип_i становится входным и одновременно выходным местом каждого внутреннего перехода из T_i .
- (3) Место гип_i и каждое внутреннее место p_i из P_i становится входным местом соответственно λ -перехода $c_{\text{гип}}$ и λ -перехода c_{pi} .
- (4) Каждое внутреннее место p_i из P_i становится выходным местом l λ -переходов из D_l , где $l = M_o(p_i)$.
- (5) Каждое выходное место из t' становится выходным местом λ -перехода e_i .

На рис. 8, *з* показан пример составного перехода, на рис. 8,б — построенная по описанному алгоритму сеть с приоритетами.

Анализ функционирования исходной иерархической сети с единственным составным переходом и построенной сети с приоритетами непосредственно убеждает, что эти сети эквивалентны. Действительно, пока не сработает переход b_i , ни один переход из T_i , не может сработать, так как место гип_i не содержит фишек. Переход b_i может сработать только при тех условиях, при которых активируется составной переход t . После срабатывания b_i место гип_i получает фишку, и может функционировать подсеть, соответствующая внутренней сети перехода t . Переход b_i не может повторно сработать, пока не сработает переход e_i , соответствующий завершению составного перехода t , так как $\text{pr}(e_i) > \text{pr}(b_i)$.

Поскольку приоритет внутренних переходов перехода t — наивысший, то внутренняя сеть будет функционировать до тех пор, пока ни один из этих переходов не сможет сработать. Лишь после этого может сработать переход $c_{\text{гип}}$, который "выключит" внутреннюю сеть, убрав фишку из гип_i . Затем могут сработать остальные переходы из C_i , удаляющие все фишки из внутренних мест. Переходы из D_i могут сработать и передать начальную разметку, которая хранится в местах q_1, \dots, q_m , в соответствующие внутренние места.

Наконец, может сработать переход e_i , после чего функционирование фрагмента сети с приоритетами, соответствующего составному переходу t , заканчивается. Префиксные и терминальные языки, порождаемые в обоих случаях, совпадают.

В случае произвольной иерархической сети преобразование в сеть с приоритетами состоит в последовательной замене составных переходов на фрагменты сети с приоритетами. Сначала заменяются составные переходы с простыми внутренними переходами. Затем заменяются составные переходы второго уровня. При этом процедура замены составных переходов второго уровня повторяет описанную выше процедуру замены составных переходов первого уровня. Однако она не затрагивает те переходы и места, которые входят в уже

Кононюк А.Е. Графы

построенные по переходам первого уровня фрагменты сети с приоритетами. Исключение составляют λ -переходы b , которые соединяются входными и выходными дугами с новым местом run^*

Приоритеты переходов для второго уровня ниже приоритетов переходов в фрагментах для первого уровня. Приоритет λ -переходов b , вставляемых вместо составных переходов второго уровня, равен приоритету, назначаемому переходам, охватываемым этими составными переходами. \square

Теорема 11. *Класс иерархических сетей равномогнен классам ингибиторных сетей и сетей с приоритетами и строго мощнее классов (ординарных) сетей Петри и регулярных сетей.*

Доказательство. Справедливость утверждения теоремы немедленно следует из теорем 9 5.5. На рис. 9 показаны примеры иерархических сетей, которые порождают языки, не принадлежащие классам языков сетей Петри.

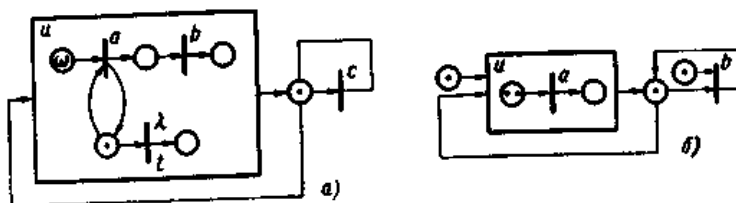


Рис. 9

Помеченная иерархическая сеть на рис. 9,а задается структурированной формулой

$$1 * (u [c])$$

$$u: (\omega s; b), (1 * s; t)$$

и помечающей функцией Σ такой, что $\Sigma(s) = s, \Sigma(b) = b, \Sigma(c) = c, \Sigma(t) = \lambda$, т.е. переход t является λ -переходом. Эта сеть порождает язык, который, как было установлено ранее, не порождается никакой помеченной сетью Петри. Сеть на рис. 9,б порождает свободный язык $\{a, aa, aab, b, ba, baa\}$, который не принадлежит классу свободных языков сетей Петри.

2.10.5. Сети с ожиданием и строго иерархические сети

Определение правил функционирования иерархической сети, данное в 2.10.3, предполагает, что локальные и внешние входные места внутренних переходов в составных переходах равноправны при выяснении возможности срабатывания переходов. В результате составной переход завершается, если ни один из его внутренних переходов не может сработать, так как хотя бы одно из его входных мест, безразлично — локальное или внешнее, не имеет фишек.

Рассмотрим, например, иерархическую сеть на рис 10.

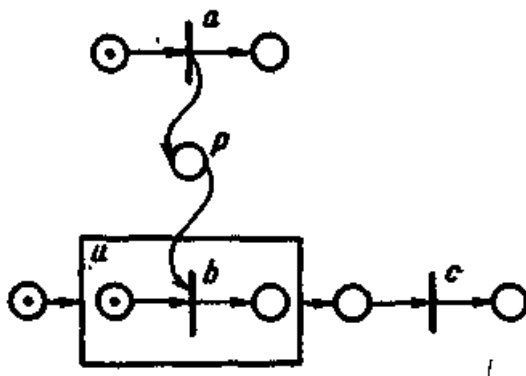


Рис. 10

Если составной переход u перешел в активное состояние, а переход a еще не сработал, то переход b не может сработать и тогда переход u завершается и становится пассивным. Здесь место p , внешнее место для перехода b и стороннее место для составного перехода u , служит как бы "внешним выключателем" для внутренней сети перехода u : если p содержит фишку, то внутренняя сеть функционирует (срабатывает переход b), а отсутствие фишек в p немедленно завершает u (переход b не срабатывает при этом). Можно изменить семантику функционирования иерархической сети таким образом, чтобы сторонние места служили не выключателями, а "внешними задержками": в случае сети на рис. 10 и описанной выше ситуации (переход a не сработал), хотя переход b не может сработать, переход u не завершается, а ждет появления фишек в месте p . Условием завершения перехода u будет служить тот факт, что локальное входное место перехода b не содержит фишек.

Таким образом, можно ввести новый тип иерархических сетей — *сети с ожиданием*, синтаксис которых полностью совпадает с синтаксисом

Кононюк А.Е. Графы

иерархических сетей из 2.10.3, но семантика изменяется за счет изменения условия завершения составного перехода.

В иерархической сети с ожиданием активный составной переход u может завершиться, если для любого его внутреннего перехода t верно, что

(а) переход t пассивен,

(б) не выполняется *локальное условие* активации перехода t :

$$\forall p \in P_{it}: M(p) - F(p, t) \geq 0,$$

где P_{it} — множество локальных входных мест перехода t .

Сети с ожиданием более адекватно, чем "простые" иерархические сети, описывают и моделируют, например, структуры операционных систем, для которых типичны параллельные процессы с ожиданием: процесс ждет информацию, вырабатываемую другими процессами, или процесс ждет освобождения ресурса, захваченного другими процессами.

Обозначим через S_1 класс иерархических сетей, введенных в 2.10.3, а через S_2 — класс иерархических сетей с ожиданием. Пару иерархических сетей - одна сеть N_1 из класса S_1 и вторая сеть N_2 из класса S_2 - назовем *подобными*, если они задаются одной и той же формулой сети.

Теорема 12. Для любой пары подобных сетей $N_1 \in S_1$ и $N_2 \in S_2$ верно: $L(N_1) \supseteq L(N_2)$, где $L(N)$ - полный свободный язык иерархической сети N .

Доказательство. Любому переходу t , который может стать активным в L/Γ при начальной разметке N_2 , может стать активным и в N_1 , при M_0 , так как различие между сетями из S_1 и из S_2 связано только с условиями завершения составных переходов. Начальная разметка M_0 сменится на одну и ту же разметку M_1 в обеих сетях, потому что топология сетей N_1 и N_2 одинакова. Следовательно, для любой последовательности срабатываний $\tau = t_1, t_2, \dots, t_n$ из $L(N_2)$ верно, что $t_1 \in TT$ является и первым символом некоторой последовательности $\tau' \in L(N_1)$. Индукцией по длине последовательностей из $L(N_2)$ легко устанавливается, что в $L(N_1)$ существует $\tau' = \tau$.

В то же время в $L(N_1)$ могут существовать последовательности срабатываний, которые не входят в $L(N_2)$. Они возникают в том случае, когда некоторый составной переход может завершиться в N_1 , послать фишки в свои выходные места и тем самым активировать другие переходы, а в N_2 этот переход не может завершиться, так как ожидает появления фишек в сторонних местах. Например, для подобных сетей N_1 и N_2 со структурой сети, изображенной на рис. 10, соответствующие языки имеют вид:

$$L(W_1) = \{a, \bar{u}, a\bar{u}, \bar{u}a, \bar{u}ab, a\bar{u}b, \bar{u}ab\bar{u}, a\bar{u}b\bar{u}, \bar{u}ab\bar{u}c, a\bar{u}b\bar{u}c, \bar{u}\bar{u}, \bar{u}\bar{u}a, \bar{u}\bar{u}c, \bar{u}\bar{u}ac, \bar{u}\bar{u}ca\},$$

$$L(W_2) = \{a, \bar{u}, a\bar{u}, \bar{u}a, \bar{u}ab, a\bar{u}b, \bar{u}ab\bar{u}, a\bar{u}b\bar{u}, \bar{u}ab\bar{u}c, a\bar{u}b\bar{u}c\}.$$

Здесь \bar{u} обозначает активирование составного перехода u , а $\bar{u}\bar{u}$ - его завершение.

Теорема 13. Любую иерархическую сеть с ожиданием из S_2 можно преобразовать в λ -эквивалентную сеть из класса S_1 .

Доказательство. Достаточно в каждый составной переход u добавить λ -переходы d_1, d_2, \dots, d_n по одному на каждый охватываемый переходом u переход t_1, \dots, t_n имеющий хотя бы одно внешнее входное место. Входными и одновременно выходными местами λ -перехода d_i назначаются локальные входные места перехода $t_i, 1 \leq i \leq n$. Переходы d_1, \dots, d_n не позволяют завершиться составному переходу u в построенной сети, пока для всех них не окажутся ложными условия активации, которые совпадают с локальными условиями активаций переходов t_1, \dots, t_n .

На рис. 11, а показан пример составного перехода сети с ожиданием, а на рис. 11, б — фрагмент λ -эквивалентной сети из класса S_1 .

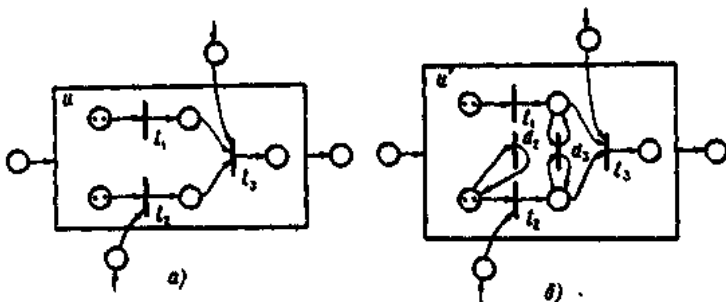


Рис. 11.

Теорема 14. Произвольную ингибиторную сеть можно преобразовать в λ -эквивалентную иерархическую сеть с ожиданием из класса S_2 .

Доказательство. Преобразование осуществляется в три этапа. На первом этапе ингибиторная сеть преобразуется в λ -эквивалентную ингибиторную сеть, обладающую следующим свойством: любой переход сети содержит среди своих входных и выходных мест не более одного места, из которого исходит ингибиторная дуга. На втором этапе полученная ингибиторная сеть представляется как результат наложения базовых фрагментов $V_{p_1}, V_{p_2}, \dots, V_{p_n}$, где $\{p_1, p_2, \dots, p_n\}$ -

Кононюк А.Е. Графы

множество всех мест сети. Для каждого фрагмента ингибиторной сети строится λ -эквивалентный фрагмент сети с ожиданием. Наконец, третий этап состоит в сборке искомой сети из класса S_2 с помощью операции наложения.

1-й этап. Исходная сеть преобразуется в λ -эквивалентную ингибиторную сеть таким образом, чтобы никакая пара базовых фрагментов второй сети не содержала один и тот же переход. Если ни один переход исходной сети не содержит среди своих входных и выходных мест более одного места, инцидентного ингибиторной дуге, то она остается без изменений. В противном случае для каждого перехода a , не удовлетворяющего вышеприведенному требованию, добавляется λ -переход t_a , дополнительные места с некоторой разметкой и дуги, соединяющие эти места с переходами сети. Число дополнительных мест и конфигурация новых связей зависят от того, как связан переход с местами, инцидентными ингибиторным дугам (назовем такие места ингибиторными).

а) Пусть переход a имеет два входных ингибиторных места (см. пример на рис. 12, а).

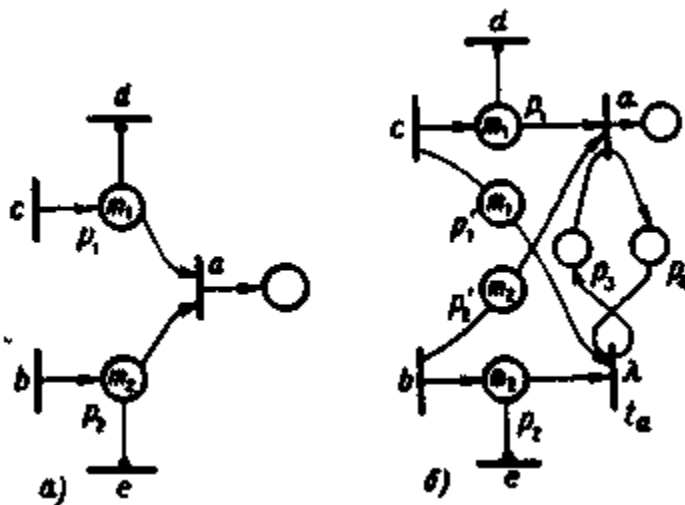


Рис. 12

Добавляются четыре места (места p'_1, p'_2, p_3, p_4 на рис. 12, б). Первые два места являются копиями двух ингибиторных мест, имеют ту же разметку и в них заводятся новые дуги от тех переходов, от которых ведут дуги в ингибиторные места. Из одного нового места заводятся

Кононюк А.Е. Графы

дуга на переход a , из другого - t_a . Два других места связываются дугами с переходами a и t_a так, как показано на рис. 12, б, и в место p_3 помещается фишка. Эти два места разрешают переходам a и t_a работать только поочередно. Можно непосредственно убедиться, что такое преобразование приводит к λ -эквивалентной сети. Преобразование легко обобщается на случай более двух входных ингибиторных мест.

б) Пусть переход a имеет входное p_1 и выходное p_2 ингибиторные места (см. пример на рис. 13, а).

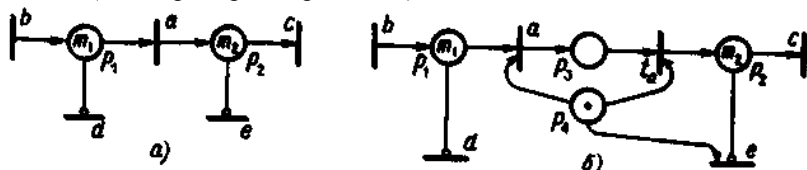


Рис. 13.

В этом случае добавляются два дополнительных места p_3 и p_4 и дополнительные дуги. Места p_3 и p_4 играют ту же роль, что и одноименные места в предыдущем случае. Кроме того, из места p_4 заводится дуга на переходы, связанные ингибиторными дугами с местом p_2 . Наличие этой дуги не разрешает этим переходам сработать в том случае, когда место p_2 имеет нулевую разметку, переход a сработал, а λ -переход t_a еще не сработал и не поместил фишку в место p_2 .

в) В случае, если a имеет два выходных ингибиторных места, добавляются места p_3 , p_4 и дуги заводятся так же, как и в предыдущем случае. Дуга, ведущая из a на одно из мест, "передается" λ -переходу t_a . Остальные случаи наличия нескольких ингибиторных мест у перехода a представляются как комбинации рассмотренных случаев, и последовательно применяются описанные преобразования. Во всех случаях, если переход a имеет среди своих входных мест такое место p , на которое не ведет ни одна дуга, то к этому месту присоединяется λ -переход с одним входным местом p' . Начальная разметка места p становится нулевой, а место p' получает разметку места p . Это необходимо для того, чтобы на последующих этапах место p стало внешним местом для составного перехода, содержащего переход a .

2-й этап. Построенная на первом этапе ингибиторная сеть представляется как результат наложения ее базовых фрагментов. Если базовый фрагмент не содержит ингибиторных дуг, то этот фрагмент объявляется без изменений фрагментом сети из класса S_2 . В противном случае базовый фрагмент V_p (рис. 14, а) преобразуется во фрагмент W_p ,

иерархической сети с ожиданием так, как показано на рис. 14, а (число переходов, связанных с местом p дугами одного типа, несущественно).

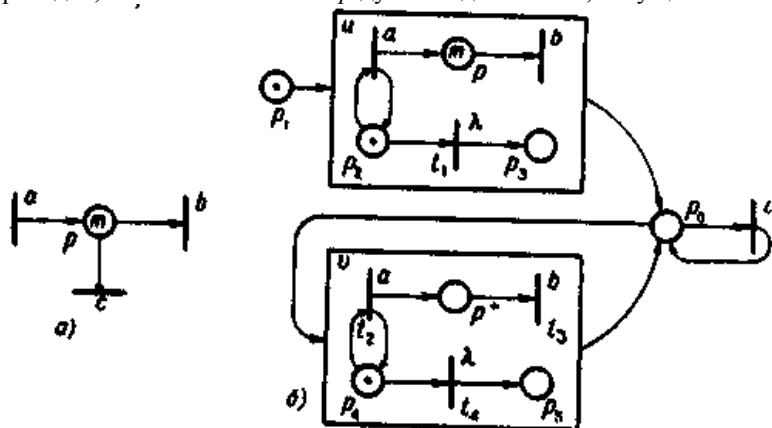


Рис. 14.

Фрагмент W_p состоит из двух составных переходов, отличающихся разметкой внутренних мест p и p^* . места p^0 и перехода c . Месту p в V_p соответствуют три места p , p^* и p^0 в W_p . В месте p^0 фишка может появиться, только если разметка мест p или p^* становится нулевой.

Наличие двух составных переходов одинаковой структуры вызвано тем фактом, что в составном переходе начальная разметка его внутренних мест восстанавливается после завершения перехода. Если место p в V_p имеет ненулевую разметку m , то после завершения перехода u , т.е. после того как место p лишится всех фишек, разметка m в нем восстановится. Поэтому вместо перехода u следующий раз начнет работать переход v , в котором место p^* , аналогичное месту p , не содержит фишек. (Заметим, что если p в V_p имеет нулевую разметку, составной переход v можно элиминировать в W_p .)

3-й этап. Осуществляется наложение фрагментов сети из класса S_2 , построенных по всем базовым фрагментам промежуточной ингибиторной сети. Затем для каждого фрагмента W_p из любого внешнего места, связанного дугами с переходами a и b в составном переходе u , заводятся такие же дуги для переходов t_2 и t_3 в составном переходе v . Одинаковые метки переходов элиминируются известным методом и проводится регуляризация сети. Преобразование исходной ингибиторной сети в иерархическую сеть с ожиданием завершено.

Теорема 15. *Класс S_2 иерархических сетей с ожиданием равномогчен классу S_1 иерархических сетей, классу ингибиторных сетей и классу сетей с приоритетами. Он строго мощнее класса сетей Петри и класса регулярных сетей.*

Доказательство. Из теоремы 13 следует, что класс S_1 мощнее класса S_2 . Ранее было показано (теорема 11), что класс S_1 равномогчен классу ингибиторных сетей и сетей с приоритетами. Теорема 14 говорит о том, что класс S_2 мощнее класса ингибиторных сетей. Отсюда следует, что все эти классы сетей равномогчны.

Тот факт, что класс сетей с ожиданием строго мощнее классов сетей Петри и регулярных сетей, непосредственно следует из теоремы 11.

Запретив использование в иерархических сетях мест, которые являются внешними для внутренних переходов составных переходов, можно выделить класс иерархических сетей S_0 , который образует собственный подкласс как класса S_1 , так и класса S_2 . Назовем сети из класса S_0 *строго иерархическими*. Из определений всех трех классов следует, что $S_0 = S_1 \cap S_2$, так как в классе S_0 условия завершения составных переходов связаны только с локальными местами этих переходов. Возникает вопрос, каковы моделирующие возможности строго иерархических сетей?

Теорема 16. *Класс строго иерархических сетей строго мощнее классов сетей Петри и регулярных сетей.*

Доказательство. Сети на рис. 9 являются строго иерархическими, так как не содержат мест, сторонних для составных переходов. Как установлено в теореме 11, эти сети порождают языки, не принадлежащие классам языков сетей Петри.

В то же время класс строго иерархических сетей оказывается не "универсально мощным". В ряде работ показано, что класс ингибиторных сетей строго мощнее класса S_0 , так как существуют ингибиторные сети, для которых нет λ -эквивалентных сетей из класса S_0 . Примером такой сети может служить ингибиторная сеть на рис. 12, а.

В целом сравнительный анализ выразительной мощности иерархических сетей и сетей Петри показывает, что введение иерархии в сетевые модели существенно улучшает моделирующие способности моделей. В свою очередь наличие межуровневых связей в иерархических системах увеличивает сложность их функционирования, и такие системы не моделируются "хорошо структурированными" сетями такими, как строго иерархические сети.

2.11. Сети-системы и сети-процессы

Сеть Петри является абстракцией динамической системы в том смысле, что ее переходы соответствуют событиям в системе, а места - условиям наступления событий. Событие — это некоторый факт в системе, обычно трактуемый как потенциальное действие компонента системы, которое может осуществляться один, несколько раз или ни разу. Функционирование системы в целом (или некоторой ее части — подсистемы) порождает *процесс* — аранжированную во времени или некоторым другим способом совокупность *реализаций событий (действий) и изменений условий*. В общем случае система может порождать разные процессы, а множество всех процессов, порождаемых системой, полностью характеризует динамику поведения системы.

Одним из способов изучения множества всех процессов, порождаемых системой, служит изучение языков соответствующей сети Петри. Однако языки сети Петри недостаточно полно отражают поведение моделируемой системы.

Для описания процессов, исследования их свойств и изучения связей между свойствами нужна некоторая синтаксическая форма представления процессов (протоколы процессов). В этом разделе рассматриваются методы сетевого представления процессов и устанавливаются взаимосвязи между свойствами сетей, моделирующих системы, и свойствами сетей, моделирующих процессы.

Идея разработки общей теории систем и процессов в терминах сетей восходит к ранним работам К. Петри, конкретизируется в статье и развивается в ряде последующих работ Петри и других авторов. В данном разделе, понятие параллельного процесса обобщается за счет допущения в процессах наряду с параллелизмом действий (и условий) альтернативы, возникающей в тех случаях, когда разрешено в процессе указывать набор взаимоисключающих действий, а также конкуренции, исключающей строгий параллелизм (одновременность), но допускающей произвольный порядок следования действий.

Изучение взаимосвязи между сетями-системами и сетями-процессами нацелено прежде всего на выявление требований к осмысленным, "хорошо устроенным" сетям, моделирующим "реалистичные" системы и процессы. С этой целью вводится ряд структурных свойств для сетей-процессов, с помощью которых выделяются хорошо устроенные сети-процессы. Затем формулируются требования к сетям-системам, которые могут порождать только такие, хорошо устроенные процессы.

Этот круг исследований еще далеко не завершен, но в силу его важности для целенаправленного развития общей теории сетей, опирающейся на строгие базовые семантические понятия, мы приводим в этом разделе некоторые начальные результаты, говорящие о продвижении в данном направлении.

2.11.1. Процессы

При определении функционирования сети Петри были использованы такие понятия, как граф разметок, множество достижимых разметок, множество последовательностей срабатываний. Путь в графе разметок от корневой вершины или последовательность срабатываний можно рассматривать как представление некоторого процесса, порождаемого сетью Петри, в форме линейно упорядоченной последовательности символов переходов и разметок мест или просто символов переходов. В последнем случае такая последовательность - это слово в алфавите T , а все множество возможных в сети процессов характеризуется свободным языком сети Петри. Этих понятий достаточно было для формулировки и анализа математических свойств сетей, но они не всегда адекватно отражают структурные и поведенческие аспекты сетей, в частности, при сопоставлении их с поведением моделируемых систем, особенно если речь идет о моделировании параллельных асинхронных систем. В этих случаях представление процессов в виде последовательностей реализаций событий отодвигает на второй план такие отношения между ними, как параллелизм или конфликты из-за ресурсов. Например, простая сеть на рис. 1, а порождает для $M_f = \bar{0}$ свободный терминальный язык $\{t_1 t_2, t_2 t_1\}$.

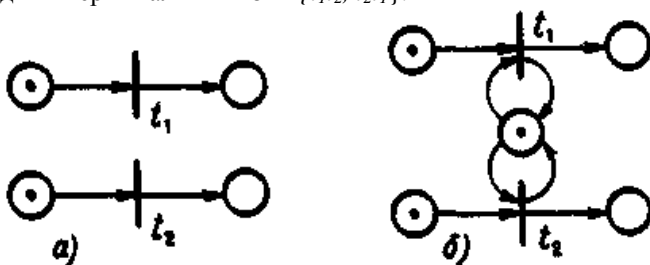


Рис. 1.

Этот язык описывает множество возможных строгих упорядочений действий в процессах, порождаемых данной сетью. При этом оказывается, что такой же свободный терминальный язык порождается

Кононюк А.Е. Графы

показанной на рис. 1, б сетью Петри другой структуры и с процессами другого характера. Действительно, в системе, моделируемой первой сетью, события t_1 и t_2 могут произойти одновременно (т.е. в множестве возможных процессов существует процесс, в котором события t_1 и t_2 параллельны). С другой стороны, среди процессов, порождаемых второй сетью, такого процесса нет: события t_1 и t_2 могут реализоваться в любом последовательном порядке, но только не одновременно.

Можно протоколировать процессы с помощью временных диаграмм, в которых по одной оси откладывается дискретное время, а по другой - выполнившиеся действия. Этот способ описания процессов, во-первых, громоздок, а во-вторых, обладает теми недостатками временного представления процессов в асинхронных системах, о которых говорили ранее.

Поиск новых форм представления процессов, которые были бы избавлены от указанных недостатков, приводит к идее задания процессов в виде структур сетевого типа, подобных структурам порождающих эти процессы систем. При таком определении процесс выглядит как совокупность реализаций событий и изменений условий системы, связанных отношениями разного типа, и задает, по существу, не единственный временной протокол функционирования системы, а некоторое множество временных протоколов, различающихся конкретными привязками действий ко времени. Другими словами, процесс представляет собой класс эквивалентности для временных протоколов, в котором эквивалентные протоколы характеризуются одним и тем же набором причинно-следственных и других отношений между действиями и условиями.

Базовыми, неопределенными понятиями, из которых строится процесс, служат два типа *элементов* процесса: *действия* и *изменения условий*. Действие в процессе - это реализация события системы, порождающей процесс. Изменение условия - это разовая реализация факта изменения некоторого условия в системе. Процесс представляет собой множество элементов процесса, содержащее хотя бы одно действие, хотя бы одно изменение условия и некоторую совокупность отношений, определенных на этом множестве элементов.

Все элементы процесса уникальны (различны). В понятии процесса, как оно введено в работах Петри, каждый элемент реализуется в процессе ровно один раз. Однако ниже мы будем рассматривать обобщение понятия процесса - *процессы с альтернативами*. В представлении такого процесса могут содержаться альтернативные действия и изменения условий, которые взаимно исключают друг друга: если выполняется, например, одно действие, то другое не

Кононюк А.Е. Графы

реализуется, и наоборот, но роено одно из этих действий обязательно происходит.

Типы отношений, связывающих элементы процесса, определяют тип процесса. Все они вводятся через базовое *отношение предшествования* элементов процесса, которое обозначается символом $<$.

Пусть x и y - элементы некоторого процесса. Запись $x < y$ трактуется таким образом, что элемент x входит в процесс "раньше, чем" элемент y , т.е. действие или изменение условия x завершится до того, как начинается действие или изменение условия y . При причинно-следственной трактовке процесса эта запись означает, что появление элемента y в этом процессе является следствием появления в нем элемента x . Постулируется, что отношение предшествования не рефлексивно и транзитивно. (Более точно:

$$x < y \iff \exists \pi \in \Pi: (t_x < t_y) \wedge \forall \pi \in \Pi: (t_x < t_y \text{ or } t_y < t_x).$$

где Π - множество всех временных протоколов процесса, t_x - момент завершения реализации элементах, t_y - момент начала реализации элемента y , *or* - логическая операция "исключительное или"

$$(A \text{ or } B \iff (A \wedge \neg B) \vee (\neg A \wedge B)).$$

Отношение следствия Π между элементами безальтернативного процесса определяется следующим образом:

$$x \Pi y \iff (x < y \text{ or } y < x) \vee (x = y)$$

Если элементы x и y связаны отношением Π в процессе, то возможен только один из двух вариантов: либо x всегда реализуется в процессе раньше, чем y , либо y реализуется в процессе раньше, чем x . Отношение следования рефлексивно ($x \Pi x$), симметрично ($x \Pi y \iff y \Pi x$) и нетранзитивно (из $x \Pi y \wedge y \Pi z$ не следует, что $x \Pi z$).

Отношение параллелизма со определяется следующим образом для процессов без альтернатив:

$$x \text{ со } y \iff (\neg(x < y) \wedge \neg(y < x)) \vee (x = y).$$

Это отношение не накладывает никаких ограничений на порядок следования элементов и не устанавливает никаких причинно-следственных связей между ними. Оно рефлексивно, симметрично и нетранзитивно.

Отношение конкуренции со_п определяется следующим образом:

$$x \text{ со}_p y \iff (x < y \wedge y < x) \vee (x = y).$$

Это отношение, в отличие от отношения следования Π , разрешает элементам x и y реализоваться в процессе в любом порядке: x может

Кононюк А.Е. Графы

как предшествовать y , так и следовать за ним. Однако x и y не могут реализоваться "одновременно". Отношение con рефлексивно, симметрично и нетранзитивно.

Для того чтобы определить отношение альтернативы, введем специальный фиктивный элемент Ω в процесс, который по определению реализуется "позже всех других элементов процесса". Тогда высказывание $x < \Omega$ выполнимо для элемента x , реализовавшегося в процессе, и только для него. Запись $\neg(x < \Omega)$ означает, что элемент x не реализовался в процессе.

Отношение альтернативы al определяется следующим образом:

$$x \text{ al } y \iff (x < \Omega \Rightarrow \neg(y < \Omega)) \vee (x = y).$$

Если процесс содержит элементы x и y , находящиеся в отношении al , то реализация одного из этих элементов исключает возможность реализации другого. Это отношение также рефлексивно, симметрично и нетранзитивно.

Данные выше определения отношений следования и параллелизма относились к элементам процессов без альтернативы, когда каждый элемент реализуется в процессе ровно один раз. Обобщим эти определения на случай процессов с альтернативами, в которых некоторые из альтернативных элементов могут не реализовываться:

$$x \text{ li } y \iff ((x < \Omega \wedge y < \Omega) \Rightarrow (x < y \text{ or } y < x)) \vee (x = y),$$

$$x \text{ co } y \iff ((x < \Omega \wedge y < \Omega) \Rightarrow (\neg(x < y) \wedge \neg(y < x))) \vee (x = y).$$

Обобщение состоит в том, что дополнительно выделены случаи, когда один из элементов x и y или оба элемента могут не реализоваться за счет того, что реализованы их альтернативы.

Процесс называется *последовательным*, если все его элементы связаны отношением следования, т.е. $\forall (x, y) \in X \times X: x \text{ li } y$, где X — множество элементов процесса. В последовательном процессе отношение следования является отношением полного упорядочения.

Процесс называется *параллельным*, если любая пара его различных элементов связана либо отношением li , либо отношением co .

Процесс называется *последовательно-альтернативным*, если любая пара его различных элементов связана либо отношением li , либо отношением al .

Процесс называется *параллельно-альтернативным*, если любая пара его различных элементов связана одним из отношений li , co , al .

Процесс называется *параллельным с конкуренцией*, если любая пара его различных элементов связана одним из отношений li , co , con .

Ниже мы рассмотрим сетевые представления для введенных типов процессов.

2.11.2. Сетевое представление параллельных процессов

Ранее было введено общее определение сети, на основе которого строятся различные классы динамических сетей, в том числе сети Петри и сети, представляющие введенные выше типы процессов. Напомним, что сеть - это тройка (P, T, F) , где P — непустое множество мест, T — непустое множество переходов, F — отношение инцидентности. Кроме того, для сетей выполнены следующие условия.

A1. $P \cap T = \emptyset$.

A2. $(F \neq \emptyset) \wedge (\forall x \in P \cup T, \exists y \in P \cup T; xFy \vee yFx)$ (любой элемент сети инцидентен хотя бы одному элементу другого типа).

A3. $\forall p_1, p_2 \in P: ({}^{\bullet}p_1 = {}^{\bullet}p_2) \wedge (p_1^{\bullet} = p_2^{\bullet}) \Rightarrow (p_1 = p_2)$.

Для всех рассматриваемых ниже типов сетей, представляющих процессы, понадобятся общие дополнения ограничения.

Введем вспомогательные обозначения:

$D(x)$ — путь в сети, начинающийся элементом x , т.е. конечная или бесконечная последовательность (x_1, x_2, \dots) такая, что $x = x_1$, и

$\forall i > 1: x_i F x_{i+1}$;

$D^{-1}(x)$ — обратный путь в сети, заканчивающийся элементом x , т.е. конечная или бесконечная последовательность (x_1, x_2, \dots) такая, что $x = x_n$, и $\forall i > 1: x_i F^{-1} x_{i+1}$, где F^{-1} - обращение отношения F ;

$D(x, y)$ - отрезок пути, начинающийся элементом x и заканчивающийся элементом y ;

$D^{-1}(x, y)$ - отрезок обратного пути, заканчивающийся элементом x и начинающийся элементом y ;

$H(N)$ - множество головных мест сети, не имеющих входных переходов, т.е. $H(N) = \{p \mid {}^{\bullet}p = \emptyset\}$.

A4. $\forall x, y \in X = P \cup T: xF^+y \Rightarrow \neg(yF^+x)$, если $x \neq y$, т.е. отношение F^+ (транзитивное замыкание отношения F) не симметрично, сеть не содержит циклов.

A5. $H(N) \neq \emptyset \wedge \forall x \in X, \forall D^{-1}(x): D^{-1}(x)$ - конечен. Это ограничение требует, чтобы любая сеть представляющая процесс, имела непустое множество головных мест и не содержала "бесконечных влево" путей.

A6. $\forall t \in T: {}^{\bullet}t \neq \emptyset \wedge t^{\bullet} \neq \emptyset$, т.е. любой переход имеет хотя бы одно входное и одно выходное место.

A7. $\forall p \in P: M_0(p) = \begin{cases} 1, & \text{если } p \in H(N), \\ 0 & \text{в противном случае.} \end{cases}$

Кононюк А.Е. Графы

Сети-процессы имеют начальную разметку, причем стандартную: только головные места содержат по одной фишке.

Введем сети, представляющие параллельные процессы. *Параллельной сетью* действий или *О-сетью* (от английского термина "occurrence net") будем называть сеть (со стандартной начальной разметкой), которая наряду с перечисленными выше условиями А1-А7 удовлетворяет еще одному условию.

АВ. $\forall p \in P: |p| \leq 1 \wedge |p^*| \leq 1$, т.е. каждое место сети имеет не более одного входного и не более одного выходного перехода. Места, не имеющие выходных переходов, будем называть хвостовыми местами сети. Все места, не входящие в множество головных или множество хвостовых мест, имеют ровно по одному входному и одному выходному переходу.

В общем случае *О*-сети могут быть бесконечными, а среди бесконечных сетей могут быть сети, не имеющие хвостовых мест. Легко видеть из определения *О*-сети, что в случае, если сеть конечна, она представляет собой частный случай сети Петри, а с учетом структурных ограничений, она является некоторым вариантом ациклического синхронизационного графа. В отличие от последнего, где требуется, чтобы $\forall p \in P: |p| = |p^*| = 1$, в *О*-сети каждое место инцидентно не более чем одной входной и выходной дуге. Если на сети действий перенести определение безопасной сети, то ясно, что любая *О*-сеть безопасна за счет стандартной начальной разметки и ограничений на топологию сети, задаваемых условиями А4, А5, А6.

Для того чтобы продемонстрировать, что *О*-сети могут использоваться в качестве сетевых представлений параллельных процессов, свяжем понятия, с помощью которых определялись процессы, с сетевыми понятиями. Будем считать, что элементы процесса представлены в *О*-сети ее элементами, при этом действия представлены переходами, а изменения условий - местами. Отношение предшествования представлено в сети отношением F^+ . Тогда отношение следования Π выражается через отношение F^+ следующим образом:

$$x \Pi y \iff (x F^+ y \text{ or } y F^+ x) \vee (x = y),$$

где $x, y \in X = P \cup T$.

С учетом того, что *О*-сеть ациклична ($(x F^+ y) \implies \neg (y F^+ x)$) отношение Π можно следующим образом выразить через отношение F^+ :

$$x \Pi y \iff (x F^+ y \vee y F^+ x) \vee (x = y),$$

где $x, y \in X = P \cup T$.

Отношение параллелизма переносится на элементы *О*-сети следующим образом:

$$x \text{ соу } y \iff \neg (x \Pi y) \vee (x = y) \iff \neg (x F^+ y \vee y F^+ x) \vee (x = y).$$

Кононюк А.Е. Графы

Например, переходы O -сети, показанной на рис. 2, находятся между собой в следующих отношениях:

	t_1	t_2	t_3	t_4
t_1			∞	∞
t_2		∞		∞
t_3	∞		∞	
t_4	∞	∞		∞

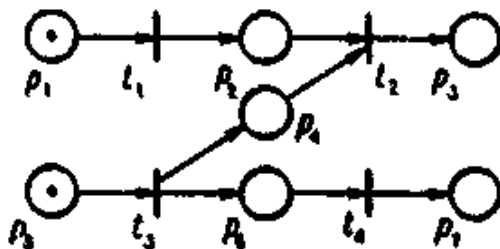


Рис. 2

В силу рефлексивности отношений Π и со любой элемент следует сам за собой и параллелен сам себе.

Естественным образом отношения Π и со между парами элементов обобщаются на произвольные наборы. Пусть $Y \subseteq X$ - непустое подмножество множества элементов сети. Множество Y называется *линией*, если $\forall x, y \in Y: x \Pi y$, т.е. любые два элемента из Y находятся в отношении следования. Множество Y называется *разрезом*, если

$\forall x, y \in Y: x \text{ со } y$, т.е. любые два элемента из Y параллельны. В графическом представлении O -сети любые ее элементы, принадлежащие некоторой линии, лежат на некотором пути в графе сети. Наоборот, все элементы, образующие разрез, попарно не соединены никаким путем. В множестве всех линий O -сети выделим максимальные, т.е. такие, которые не содержатся ни в каких других линиях. Их будем называть *Π -сечениями*. В множестве всех разрезов также выделим максимальные и их будем называть *со-сечениями*. В

Кононюк А.Е. Графы

сети на рис. 2 имеются следующие II-сечения $\{p_1, t_1, p_2, t_2, p_3\}, \{p_4, t_3, p_4, t_2, p_3\}, \{p_5, t_3, p_6, t_4, p_7\}$. Линия $\{t_3, p_4, t_2\}$ содержится в линии $\{p_5, t_3, p_4, t_2, p_3\}$ и поэтому не является li-сечением. В этой же сети много со-сечений, поэтому мы не перечисляем всех, только некоторые:

$\{p_1, p_8\}, \{p_1, p_4, p_6\}, \{p_2, p_4, t_4\}, \{t_3, t_4\}$.

Теорема 1. В O -сети $N=(P, T, F, M_0)$ любое II-сечение и любое со-сечение пересекаются не более чем по одному элементу. **Доказательство.** Предположим противное: существует II-сечение L и со-сечение C такие, что $|L \cap C| > 1$. В этом случае найдутся два элемента $x, y \in X$ такие, что $x, y \in L$ и $x, y \in C$. По определению сечений $x \in \Pi$ и одновременно $x \in \Pi$, но это может быть только в том случае, если $x = y$, т.е. L и C содержат единственный общий элемент.

Поскольку O -сети объявляются представлениями параллельных процессов, возникает вопрос, насколько корректно они представляют процессы этого типа. Сравнение определений параллельного процесса и O -сети убеждает, что все понятия, связанные с процессами, хорошо интерпретируются в терминах сетей. Чтобы установить обратное, следует выяснить, для любой ли O -сети можно найти подтверждающий ее реальный параллельный процесс. Трудность здесь состоит в том, что формальное понятие сети нужно сравнивать с неформализованным, интуитивным понятием параллельного процесса. Однако уже предварительные исследования показали, что существуют "синтаксически корректные" O -сети, которые, если их проинтерпретировать как процессы, оказываются неприемлемыми в качестве реальных процессов. В связи с этим Петри ввел дополнительное условие K -плотности для O -сетей, соблюдение которого позволяет избежать описанного выше несоответствия между синтаксическим и семантическим аспектами сетевого представления процессов.

O -сеть называется K -плотной, если в ней любое II-сечение и любое со-сечение пересекаются ровно по одному элементу. С учетом теоремы 1 это означает, что K -плотная сеть не содержит ни одной пары непересекающихся li-сечений и со-сечений. Легко убедиться непосредственно, что сеть на рис. 2 K -плотна.

Пример не- K -плотной сети приведен на рис. 3. В этой сети существует бесконечное II-сечение $\{p_1, t_1, p_4, t_1, p_7, t_3, \dots\}$, которое не пересекается с бесконечными со-сечениями $\{p_2, p_5, p_8, \dots\}, \{t_2, t_4, t_6, \dots\}, \{p_3, p_6, p_9, \dots\}$ (на рис. 3 эти

Кононюк А.Е. Графы

со-сечения показаны пунктирными линиями). Пример O -сети на рис. 3 хорошо иллюстрирует, в чем состоит несоответствие между синтаксисом и семантикой в не- K -плотной сети.

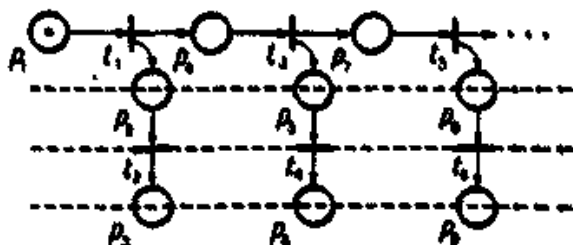


Рис. 3.

В соответствии со структурой сети в ней существуют указанные бесконечные со-сечения. Это означает, например, что все переходы сечения $\{t_2, t_4, t_6, \dots\}$ взаимно параллельны и существует конкретная временная реализация задаваемого сетью процесса, в которой все действия t_2, t_4, t_6, \dots могут произойти одновременно. С другой стороны, фишка распространяется вдоль бесконечного i_1 -сечения $\{p_1, t_1, p_4, t_3, \dots\}$ последовательно, и последовательно поступают фишки в места p_2, p_5, p_8, \dots . Этот процесс заполнения всех мест p_2, p_5, p_8, \dots фишками никогда не может завершиться, и поэтому потенциальный, "топологический" параллелизм действий t_2, t_4, t_6, \dots не может в действительности реализоваться. Таким образом, O -сеть на рис. 3 не описывает реальный параллельный процесс. Еще один пример O -сети, которая не является K -плотной, показан на рис. 4, а.

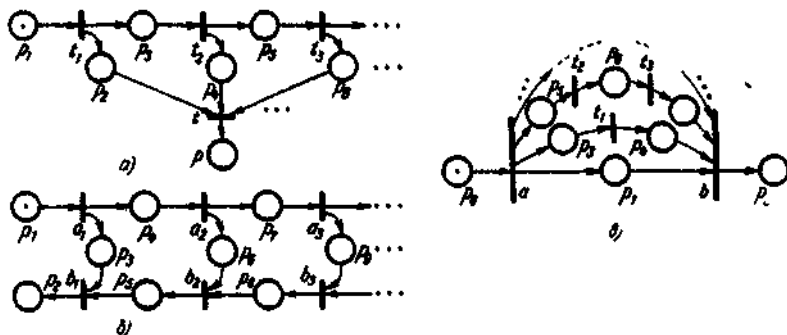


Рис. 4

Кононюк А.Е. Графы

В этой сети результат несоответствия топологического сетевого параллелизма отношению параллелизма в процессах приводит к тому, что переходу t не соответствует никакого действия. Действительно, поскольку li-сечение $\{p_1, t_1, p_3, t_2, p_5, t_3, \dots\}$ не пересекается с со-сечением $\{p_2, p_4, p_6, \dots\}$, то никогда не может случиться, чтобы все входные места перехода t имели разметку, отличную от 0. Поэтому переход t мертв.

В приведенных примерах не- K -плотные O -сети бесконечны. На рис.

4, в приведен пример также бесконечной сети, но эта сеть K -плотна. Следующая теорема говорит о том, что не K -плотной сетью может быть только бесконечная O -сеть.

Теорема 2. *Любая конечная O -сеть K -плотна.*

Доказательство. Предположим, что существует конечное со-сечение C и конечное li-сечение L такие, что $C \cap L = \emptyset$. Любой элемент li-сечения L не может быть параллелен всем элементам со-сечения C , так как в противном случае существовало бы со-сечение, содержащее C . Разобьем L на две линии L_1 и L_2 так, чтобы в L_1 входили все те элементы $x \in L$, для каждого из которых существует в C элемент y такой, что xF^+y , а в L_2 входили бы все те элементы x , для каждого из которых в C существует элемент y такой, что yF^+x . Другими словами, в L_1 входят элементы, "предшествующие" сечению C , а в L_2 входят элементы, "следующие за" сечением C . Заметим, что $L_1 \cup L_2 \neq \emptyset$. В L_1 и L_2 элементы можно частично упорядочить в соответствии с отношением F^+ . Так как L_1 и L_2 конечны, то в L_1 существует "максимальный" в этом упорядочении элемент x' , а в L_2 существует "минимальный" в этом упорядочении элемент x'' . Пусть $L_1 \neq \emptyset$, а $L_2 = \emptyset$. Тогда множество $L_1 \cup \{y'\} = L \cup \{y'\}$, где $y' \in C$, и $x'F^+y'$ образует линию, содержащую L_1 и, следовательно, L не является li-сечением, что противоречит предположению. Аналогичные рассуждения приводят к противоречию в случае $L_1 = \emptyset$, $L_2 \neq \emptyset$. Если $L_1 \neq \emptyset$ и $L_2 \neq \emptyset$, то в C существуют элементы y' и y'' такие, что $x'F^+y'$ и $y''F^+x''$. При $y' = y'' = y$ линия $L_1 \cup \{y\} \cup L_2 = L \cup \{y\}$ содержит L , поэтому L не является li-сечением, что приводит к противоречию. Если же $y' \neq y''$, то в L должен существовать элемент x такой, что $x'F^+x$ и xF^+x'' . Он не принадлежит ни L_1 , ни L_2 , и вновь образуется линия, содержащая L , что противоречит условию предположения.

Теорема 3. *В K -плотной O -сети, каждый переход которой имеет конечное множество входов, любой переход срабатывает ровно один раз.*

Кононюк А.Е. Графы

Доказательство. Любое li -сечение в O -сети содержит единственную фишку при любой достижимой разметке. Действительно, в силу условия А5 любое li -сечение начинается некоторым головным местом, содержащим фишку. При срабатывании некоторого перехода t в сети возможны два случая: t принадлежит данному li -сечению L или t не принадлежит ему. В первом случае t обязательно переместит фишку из своего входного места, принадлежащего L , в свое выходное место, также принадлежащее L . Во втором случае фишка остается в прежнем месте сечения L . Поэтому O -сеть безопасна.

Разобьем множество T переходов сети на "ярусы" следующим образом:

$$T_0 = \{t \mid t \subseteq H(N)\},$$

$$T_k = \{t \mid (\forall t' \in \cdot(t) \quad t' \in T_j \wedge j < k) \wedge (\exists t' \in \cdot(t) \quad t' \in T_{k-1})\}.$$

Ярус T_0 образуют те переходы, все входные места которых являются одновременно головными местами сети. Ярус T_k образуют те переходы, входные места которых являются выходными местами переходов из ярусов с меньшими номерами, причем хотя бы одно входное место является выходным местом некоторого перехода из яруса T_{k-1} . Заметим, что

$$T = \bigcup_{k=0}^{\infty} T_k,$$

так как для любого перехода $t \in T$ существует номер k такой, что $t \in T_k$. Этот номер можно определить следующим образом. Пусть $\{L_1, \dots, L_i\}$ — множество всех путей, ведущих из некоторого головного места сети к переходу t . Каждый из этих путей конечен, если O -сеть K -плотна. Последний факт является следствием теоремы о дискретности K -плотной O -сети. Дискретной названа сеть, не содержащая бесконечных отрезков li -сечений, соединяющих два элемента x и y сети, т.е. не содержащая бесконечных последовательностей x, x_1, x_2, \dots, z, y , где $x \in \cdot x_1, x_1 \in \cdot x_2, \dots, z \in \cdot y$. Например, сеть на рис. 4, а не дискретна, так как содержит бесконечный отрезок $\{p_1, t_1, p_2, t_2, p_3, t_3, \dots, t\}$. Тогда $k = \max(|L_1|, \dots, |L_i|) / 2 - 1$.

Покажем теперь индукцией по номеру яруса, что любой переход из любого яруса срабатывает ровно один раз. Легко видеть, что это утверждение справедливо для всех $t \in T_0$. Пусть оно справедливо для

всех переходов из ярусов T_j , где $j < k$. Тогда для любого $p \in \cdot t$, где t — переход из яруса T_k , возможны следующие два случая. В первом $p \in H(N)$ и, следовательно, для любой достижимой разметки M такой, что $M_0[p] > M$ и τ не содержит t , имеет место $M(p) = 1$. Во втором

случае $\exists t' \in T, j < k$ и $p \in (t')$. Так как по предположению t' может сработать, то существуют τ и M' такие, что $M_0[\tau > M', \tau'$ заканчивается переходом t' и $M'(p) = 1$. Заметим, что изменить разметку входных мест перехода с единичной на нулевую не может никакой другой переходу кроме самого t , в силу условия А8. Так как число k конечно, то по предположению достижима разметка M такая, при которой все входные места перехода получают по одной фишке и он может сработать, причем только один раз, так как его входные места не смогут больше получить фишки в силу условий А4 и А8.

Проиллюстрируем на примерах необходимость условий теоремы 3. На рис. 4, б приведем пример сети, каждый переход которой имеет конечное множество входных мест. В данном случае причиной того, что переходы b_1, b_2, b_3, \dots не могут сработать, является не- K -плотность исходной сети. Любой переход b_i может сработать только после выполнения последовательности переходов a_1, a_2, a_3, \dots , но эта последовательность бесконечна. В K -плотной сети на рис. 4, в переход b имеет бесконечное множество входов. Переходы a и b в данной сети соединены бесконечным множеством конечных путей. Длина каждого отдельного пути ограничена, но не существует такого натурального числа k , которое одновременно ограничивало бы длины всех данных путей. В силу этого процесс заполнения фишками входных мест перехода b никогда не может быть закончен, и переход b не может сработать в данной сети.

2.11.3. Сетевое представление последовательно-альтернативных процессов

Дополним условия А1-А7 в общем определении сетей-процессов следующими ограничениями.

А9. $|H(N)| = 1$, сеть имеет единственное головное место.

А10. $\forall t \in T: |t^-| = 1 \wedge |t^+| = 1$, т.е. любой переход сети имеет ровно одно входное и одно выходное место.

Сети, удовлетворяющие перечисленным условиям, назовем *последовательно-альтернативными сетями действий*, или S -сетями. В общем случае S -сети могут быть бесконечными, а конечные сети представляют собой ациклические автоматные сети с единичной разметкой в единственном головном месте. Из условий А9 и А10 следует, что S -сети представляют связный граф.

Точно так же, как и в случае O -сетей, S -сети являются некоторой синтаксической формой представления процессов, а именно -

Кононюк А.Е. Графы

последовательно-альтернативных процессов. Отношение следования Π выражается через отношение F^+ в S-сетях так же, как и в случае O-сетей. Отношение альтернативы al выражается для элементов S-сети через отношение F^+ следующим образом:

$$x \text{ al } y \iff \neg(x \Pi y) \vee (x = y) \iff \neg(x F^+ y \vee y F^+ x) \vee (x = y).$$

Для S-сетей можно определить понятие Π -сечения, и это определение полностью совпадает с определением Π -сечения для O-сетей. Множество A элементов S-сети назовем *альтернативным разрезом*, если $\forall x, y \in A: x \text{ al } y$, т.е. любые два элемента из A альтернативны. В графическом представлении S-сети все элементы, образующие альтернативный разрез, попарно не соединены никаким путем. В множестве всех разрезов S-сети выделим максимальные, т.е. такие, которые не содержатся ни в каких других, и будем называть их *al-сечениями*. На рис. 5,в показана S-сеть, в которой имеются следующие Π -сечения: $\{p_1, t_1, p_2, t_2, p_3, t_4, p_4, t_5, p_5, \dots\}$,

$\{p_1, t_3, p_3, t_6, p_5, \dots\}$

и следующие al-сечения: $\{t_1, t_3\}$, $\{p_2,$

$t_3\}$, $\{t_2, t_3\}$, $\{t_4, t_6\}$, $\{p_4, t_6\}$,

$\{t_5, t_6\}, \dots$

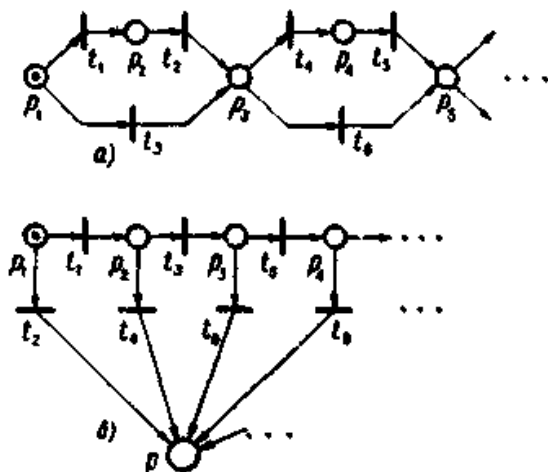


Рис. 5

Теорема 4. В S-сети любое Π -сечение и любое al-сечение пересекаются не более чем по одному элементу.

Доказательство. Предположим противное: существуют Π -сечение L и al-сечение S такие, что $|L \cap S| > 1$. В этом случае найдутся два элемента x и y сети такие, что $x, y \in L$ и $x, y \in S$. По определению

сечений $x \text{ li } y$ и одновременно $x \text{ al } y$, но это может быть только в том случае, если $x = y$, (т.е. L и S содержат единственный общий элемент. Так же, как и в случае O -сети, возникает проблема неадекватных S -сетей, (которые, если их интерпретировать как последовательно-альтернативные процессы, оказываются неприемлемыми как реальные процессы. Среди O -сетей выделялись K -плотные O -сети, для которых синтаксические (сетевые) и семантические представления соответствовали друг другу. Аналогично для S -сетей вводится понятие L -плотности, которое определяется следующим образом: S -сеть L -плотна, если в ней любое li -сечение и любое al -сечение пересекаются ровно по одному элементу. С учетом теоремы 4 это означает, что L -плотная сеть не содержит ни одной пары непересекающихся li -сечений и al -сечений. S -сеть на рис. 5, *a* является L -плотной, в то время как S -сеть на рис. 5, *б* не является L -плотной. В последней сети имеется бесконечное li -сечение $\{p_1, t_1, p_2, t_3, p_3, t_5, \dots\}$, которое не пересекается с бесконечным al -сечением $\{t_2, t_4, t_6, \dots\}$, показанным пунктирной линией. На примере этой сети можно увидеть, в чем выражается несоответствие между синтаксисом и семантикой не- L -плотной сети. Наличие в ней бесконечных al -сечений означает, что существуют бесконечные совокупности альтернативных действий (или изменений условий). По определению последовательно-альтернативного процесса в каждой полной альтернативе должно реализоваться ровно одно действие. С другой стороны, в сети на рис. 5, *б* возможна ситуация, когда фишка проходит вдоль бесконечного li -сечения $\{p_1, t_1, p_2, t_3, p_3, t_5, \dots\}$ и в результате ни одно из действий t_2, t_4, t_6, \dots не реализуется. Таким образом, указанные "топологические" альтернативы могут в действительности не иметь места.

Теорема 5. Любая конечная S -сеть L -плотна.

Опускаем доказательство этой теоремы, поскольку оно весьма похоже на доказательство теоремы 2 о K -плотности конечных O -сетей.

2.11.4. Сетевое представление параллельно-альтернативных процессов

Параллельно-альтернативные процессы мы будем представлять с помощью ациклических сетей, или A -сетей, удовлетворяющих условиям $A1$ — $A7$ и дополнительному условию $A11$, обеспечивающему безопасность сетей из данного класса (аксиома $A11$ будет сформулирована ниже).

Конюк А.Е. Графы

В А-сети переход может иметь более одного входного и/или выходного места, а место может быть инцидентно нескольким переходам. Примеры А-сетей показаны на рис. 6, а, 7, а, 8. О-сети и S-сети являются частными подклассами А-сетей.

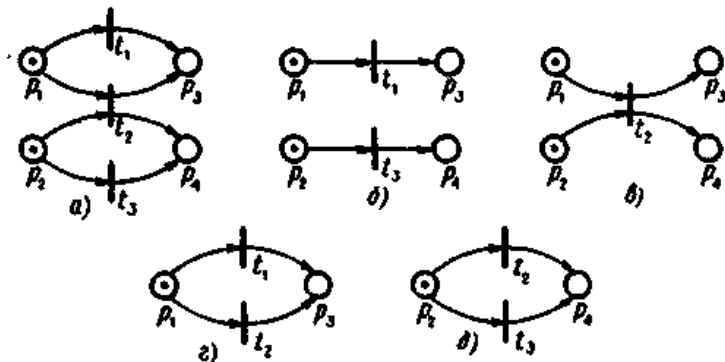


Рис. 6

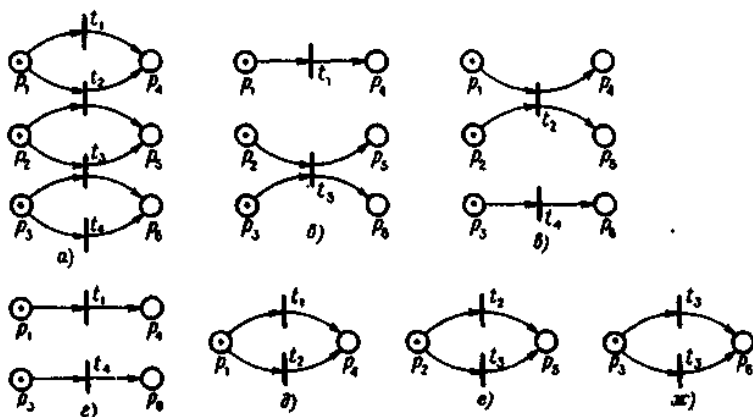


Рис 7.

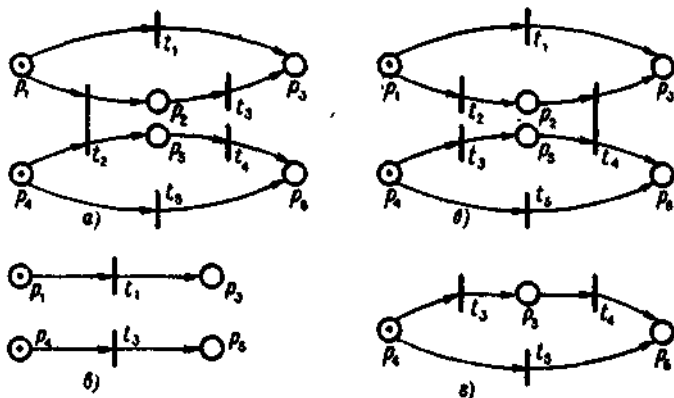


Рис. 8.

Покажем, каким образом отношения следования, параллелизма и альтернативы выражаются в терминах A -сетей. Отношение \bar{li} выражается через сетевое отношение F^+ точно так же, как и в O - и S -сетях.

Отношение альтернативности нельзя определить топологически единообразным способом для переходов и мест сети из-за разных требований к реализации условия и действия в системе и соответственно появлению фишки в месте и срабатыванию перехода в сети. Условие может реализоваться (измениться), если хотя бы одно непосредственно предшествующее ему действие реализовалось, т.е. место получит фишку, если хотя бы один переход, для которого это место является входным, сработал. Действие может реализоваться, если все непосредственно предшествующие ему условия реализовались, т.е. переход может сработать, если все его входные места имеют фишки.

Два перехода t_1 и t_2 альтернативны в A -сети в следующем случае:

$$t_1 \text{ ал } t_2 \iff \neg(t_1 \bar{li} t_2) \wedge \\ ((t_1 \cap t_2 \neq \emptyset) \vee (\exists p_1 \in \tau_1: (\forall t'_1 \in \rho_1: t'_1 \text{ ал } t_2)) \vee \\ (\exists p_2 \in \tau_2: (\forall t'_2 \in \rho_2: t_1 \text{ ал } t'_2))) \vee (t_1 = t_2).$$

т.е. разные переходы t_1 и t_2 альтернативны, если

- 1) они не находятся в отношении следования,
- 2) имеют общее входное место или непосредственный предшественник одного из переходов альтернативен другому.

Два места p_1 и p_2 альтернативны, если

$$p_1 \text{ ал } p_2 \iff (\forall t_1 \in \rho_1, \forall t_2 \in \rho_2: (t_1 \text{ ал } t_2) \wedge (t_1 \neq t_2)) \vee (p_1 = p_2).$$

Кононюк А.Е. Графы

т.е. два различных места p_1 и p_2 альтернативны, если любой переход, для которого выходным местом служит p_1 , альтернативен каждому переходу, выходным местом которого является p_2 , и все эти переходы различны.

Место p и переход t *альтернативны*, если

$$p \text{ al } t \iff (\forall t' \in \mathcal{P}: t' \text{ al } t) \wedge \neg(t \text{ li } p),$$

- т.е. место p и переход t альтернативны, если они не связаны друг с другом никаким путем и любой переход t' , для которого p является выходным местом, альтернативен переходу t . Два элемента x и y в A -сети *параллельны*, если они не связаны отношениями следования и альтернативны (или $x = y$)

$$x \text{ со } y \iff (x = y) \vee \neg(x \text{ li } y \vee x \text{ al } y).$$

Например, в сети на рис. 6, a переходы t_1 и t_2 связаны отношением al , переходы t_2 и t_3 также связаны отношением al , а переходы t_1 и t_3 связаны отношением со .

Дополнительное условие A11 формулируется следующим образом:

$$\text{A11. } \forall p \in P, \forall t_1, t_2 \in T: ((t_1, t_2 \in \mathcal{P}) \Rightarrow (t_1 \text{ al } t_2)).$$

Определения li-сечений, со-сечений, данные выше для O-сетей и S-сетей, переносятся на случай A-сетей. Al-сечение в A-сети определяется несколько иначе. Об этом будет сказано ниже.

Если рассматривать отношения li, со, al как "координатные оси" некоторого трехмерного пространства, то O-сети лежат в плоскости, образованной координатными осями li и со (нет альтернативных элементов), а S-сети лежат в плоскости, образованной осями li и al (нет параллельных элементов). Для O-сетей структурные ограничения, гарантирующие адекватное сетевое задание процессов, формулировались с помощью понятий li- и со-сечений (все они должны попарно пересекаться). Аналогично для адекватности представления последовательно-альтернативных процессов требовалось, чтобы попарно пересекались все li- и al-сечения. Для A-сетей, адекватно представляющих параллельно-альтернативные процессы, должны, во-первых, выполняться требования K-плотности и L-плотности их подсетей, вводимых ниже, и, во-вторых, требуется выполнение еще одного свойства (*M-плотности*), которое формулируется в терминах пересечения плоскостей, образованных, с одной стороны, li- и со-сечениями, а с другой — li- и al-сечениями.

Сеть $N' = (P', T', F')$ является *подсетью* сети $N = (P, T, F)$, если $P' \subseteq P$, $T' \subseteq T, F' \subseteq F \cap (P' \times T' \cup T' \times P')$.

Сеть N' называется *O-подсетью* A-сети N , если:

1) N' является подсетью N ,

Кононюк А.Е. Графы

2) для N' выполнены все условия, справедливые для О-сетей,

$$3) \forall t \in T': (\rho \in P' | \rho Ft) \subseteq P' \text{ и } \forall \rho \in P': F'(\rho, t) = F(\rho, t),$$

т.е. переход t в О-подсети N' имеет то же самое множество входных мест и все дуги, связывающие его с этими местами, что и в А-сети N .

Замечание. При определении О-подсети мы не требуем выполнения условия А3, поскольку разные в исходной А-сети места в О-подсети могут иметь одно и то же множество инцидентных переходов. Несмотря на то, что в подсети эти места совпадают, нам важно сохранить информацию, что и исходной А-сети они были инцидентны различным множествам переходов и по существу были разными.

О-подсеть N' сети N назовем *максимальной*, если:

1) для любой О-подсети N_2 сети N такой, что N_1 является О-подсетью сети N_2 , верно, что $N_1 = N_2$;

2) все головные места N' являются головными местами А-сети N .

Множество всех максимальных О-подсетей А-сети N образуют ее проекцию на координатную плоскость (li, со), а объединение свободных языков всех О-подсетей совпадает со свободным языком сети N . Для А-сети на рис. 6, а множество ее максимальных О-подсетей состоит из подсетей, показанных на рис. 6, б, в, а для А-сети на рис. 7, а - из подсетей, показанных на рис. 7, б, в, г.

Сеть $N' = (P', T', F')$ называется *S-подсетью* А-сети $N = (P, T, F)$, если:

1) N' является подсетью N ,

2) для N' выполнены все условия, справедливые для S-сетей,

$$3) \forall t_1, t_2 \in T, \forall \rho_1, \rho_2 \in P': (t_1 F \rho_1 \wedge t_2 F \rho_2 \wedge \neg(\rho_1 \parallel \rho_2)) \Rightarrow (\exists S\text{-подсеть } N'' = (P'', T'', F''): t_1, t_2 \in T'').$$

Фактически условие 3) обеспечивает, что в S-подсеть входят те и только те элементы, которые в исходной А-сети были альтернативны или находились в отношении следования.

Замечание. Для S-подсетей мы не требуем выполнения второй части условия А6, т.е. $\neg(\forall t \in T': t \neq \emptyset)$, так как из условия t_1 ал t_2 в А-сети N следует существование альтернативных входных мест (или наличие общего входного места), но альтернативность переходов не гарантирует альтернативности их выходных мест.

S-подсеть N' А-сети N назовем *максимальной S-подсетью*, если:

1) для любой S-подсети N'' сети N такой, что N' является S-подсетью сети N'' верно, что $N' = N''$;

2) головное место $H(N')$ входит во множество головных мест сети N .

Множество всех максимальных S-подсетей А-сети N образует ее проекцию на координатную плоскость (li.al). Для сети на рис. 6, а множество

Кононюк А.Е. Графы

ее максимальных S-подсетей состоит из подсетей, показанных на рис. 6, з, д, для сети на рис. 7, а - из подсетей, показанных на рис. 7, д, в, ж. Множество A элементов A-сети называется *альтернативным разрезом*, если $\forall x, y \in A: x \text{ ал } y \text{ в } N$ и существует максимальная S-подсеть N' , в которой множество A является альтернативным разрезом. Множество A элементов A-сети N называется *al-сечением*, если множество A образует максимальный альтернативный разрез в N , т.е. для любого альтернативного разреза A' в A-сети N из условия $A' \supseteq A$ следует, что $A'=A$.

A-сеть назовем *K-плотной*, если все ее максимальные O-подсети K-плотны, и A-сеть назовем *L-плотной*, если все ее максимальные S-подсети L-плотны.

A-сеть N назовем *M-плотной*, если результатом пересечения любой максимальной O-подсети сети N с любой максимальной S-подсетью сети N является некоторое (единственное) li-сечение сети N . Заметим, что если для определения понятий K- и L-плотности в "двумерных" O- и S-сетях требовалась единственность элемента, по которому пересекаются сечения, то в слугае "трехмерной" A-сети понятие M-плотности определяется с помощью пересечения "плоскостей" и это пересечение должно быть сечением. A-сеть на рис. 6, в является M-плотной. Действительно, ее максимальная O-подсеть на рис. 6, б пересекается с ее максимальными S-подсетями на рис. 6, з и а по li-сечениям (p_i, t_2, p_3) и соответственно (p_2, t_3, p_4) , а максимальная O-подсеть на рис. 6, в пересекается с теми же максимальными S-подсетями по li-сечениям (p_1, t_2, p^3) и соответственно (p_2, t_2, p_4) . С другой стороны, A-сеть на рис. 7, а не является M-плотной, так как пересечение ее максимальной O-подсети на рис. 7, з с ее максимальной S-подсетью на рис. 7, е пусто.

На рис. 8 показаны еще две A-сети, из которых одна является M-плотной (сеть на рис. 8, а), другая - не-M-плотной (сеть на рис. 8, б). Одна из максимальных O-подсетей второй сети (рис. 8, в) пересекается с одной из максимальных S-подсетей этой же сети (рис. 8, г) по линии $(p_4, t_3 p_5)$, которая не является li-сечением сети, так как образует лишь начальный фрагмент li-сечения $(p_4, t_3, p_5, , t_4, p_6)$.

A-сеть назовем *плотной*, если она K-плотна, L-плотна, M-плотна. На рис. 9, а показан пример плотной сети, а на рис. 9, б -сети, которая не является ни K-, ни L-, ни M-плотной.

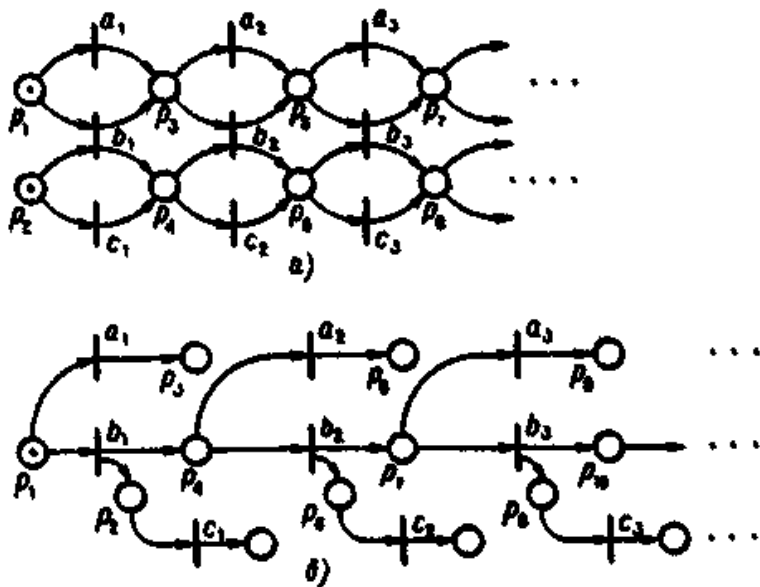


Рис. 9

Возможны различные варианты неплотности, возникающие за счет разных сочетаний нарушений структурных ограничений. Заметим, что если конечность O - или S -сети гарантировала K - или L -плотность этой сети, то конечная A -сеть может оказаться не плотной, но только за счет того, что она не является M -плотной.

Следующие утверждения иллюстрируют адекватность A -сетей как сетевых представлений параллельно-альтернативных процессов.

Назовем сеть *правильной*, если для любой достижимой тупиковой разметки M и для любого места p , не принадлежащего множеству хвостовых мест сети, $M(p) = 0$. (Заметим, что бесконечные A -сети могут в общем случае не иметь хвостовых мест и/или тупиковых разметок. В этих случаях они считаются правильными.)

Лемма 1. *Любая ациклическая сеть безопасна.*

Доказательство. Начальная разметка A -сети безопасна по определению. В силу условия A11 любое место может иметь в качестве входных переходов только альтернативные переходы. Поэтому исключена ситуация, когда некоторое место сети может получить более одной фишки (от паг параллельных переходов).

Лемма 2. *Любая S -сеть и любая O -сеть являются правильными.*

Кононюк А.Е. Графы

Доказательство. В S -сети любая достижимая разметка содержит ровно одну фишку. В силу связности S -сети любая тупиковая разметка достижима только в том случае, если единственная фишка сети попадает в хвостовое место.

В конечной O -сети любой переход сработает ровно один раз, перемещая фишки из всех своих входных мест в выходные. Поэтому при достижении тупиковой разметки ни одно из мест, не входящих во множество хвостовых мест сети, не может содержать фишек. В бесконечной O -сети нет достижимых тупиковых разметок, и она по определению — правильная.

Теорема 6. *Плотная A -сеть является правильной.*

Доказательство. По определению M -плотной сети результатом пересечения любой ее максимальной O -подсети с любой ее максимальной S -подсетью является li -сечение. Это означает, что любое li -сечение максимальной O -подсети N' является одновременно и li -сечением в исходной сети N . Отсюда следует, что любое хвостовое место O -подсети N' является хвостовым местом сети N , а множество хвостовых мест сети N образовано объединением множеств хвостовых мест всех ее максимальных O -подсетей. Заметим, что для любой достижимой разметки M в сети N множество мест, содержащих фишки, является со-сечением и существует максимальная O -подсеть, содержащая это сечение.

Поскольку любая K -плотная O -подсеть является правильной сетью (лемма 2), то при любой достижимой тупиковой разметке в плотной сети N любая ее максимальная O -подсеть содержит фишки только в своих хвостовых местах (для этой подсети такая разметка также будет тупиковой). Следовательно и сеть N содержит при тупиковой разметке фишки только в своих хвостовых местах.

Теорема 7. *В плотной A -сети любой переход может сработать не более одного раза и в любом ее a -сечении, состоящем из переходов, срабатывает ровно один переход.*

Доказательство. Первая часть утверждения теоремы справедлива для любой A -сети, что следует из ацикличности A -сети (условие $A4$) и ее безопасности (лемма 1).

Пусть A — некоторое a -сечение плотной A -сети N , для которой не выполнена вторая часть утверждения, т.е.

- (1) по крайней мере два перехода t_1 и t_2 из A срабатывают при функционировании сети N , или
- (2) все переходы из A мертвы.

В силу ацикличности сети и ее безопасности первый случай возможен, только если переходы t_1 и t_2 находятся в отношении li , но тогда они не могут оба входить в A .

Кононюк А.Е. Графы

Второй случай рассмотрим сначала для частного случая A -сетей, а именно: для S -сетей. Если S -сеть N L -плотна, то a l-сечение A пересекается со всеми ее li -сечениями. Поэтому в силу правильности L -плотной S -сети должно существовать li -сечение, все переходы которого срабатывают, в том числе и переход, входящий в a l-сечение A .

В общем случае плотной A -сети, если сеть N содержит a l-сечение A , ни один из переходов которого не срабатывает, то в N существует максимальная S -подсеть N' , содержащая это a l-сечение. Пусть

$\{N_1, N_2, \dots\}$ - множество всех максимальных O -подсетей сети N . Поскольку сеть N плотна и, следовательно, M -плотна, пересечение любой ее максимальной O -подсети N_i с подсетью N' образует некоторое li -сечение L_i в N и одновременно то же li -сечение L_i в N_i и N' . Поскольку сеть N плотна и, следовательно, L -плотна, в S -подсети N' a l-сечение A пересекается с li -сечением L_i по некоторому переходу $t_i \in L_i \cap A$. Поскольку сеть N плотна и, следовательно, K -плотна, то любой переход t_i срабатывает в максимальной K -плотной O -подсети N_i (теорема 3). Свободный язык, порождаемый A -сетью N , является объединением свободных языков всех максимальных O -подсетей N_1, N_2, \dots , поэтому один из переходов $t_i \in A$ обязательно сработает при функционировании сети N .

2.11.5. Развертка сетей Петри в сети-процессы

В предыдущих параграфах этого раздела введены сетевые представления для трех типов процессов — параллельных, последовательно-альтернативных и параллельно-альтернативных - и выделены свойства сетей, адекватно представляющих процессы. Рассмотрим, каким образом можно установить связь между сетями Петри как моделями систем и сетями, описывающими процессы их функционирования. После установления такой связи можно выделить среди сетей Петри сети, адекватно описывающие "осмысленные" системы, порождающие "осмысленные" процессы.

Введение обобщенных процессов, в которых действия и изменения условий могут быть связаны не только отношениями следования и параллелизма, но и отношениями альтернативы и конкуренции, позволяет ставить вопрос об установлении взаимно-однозначного соответствия между сетями Петри и порождаемыми ими сетями-процессами. Когда связь установлена, можно следующим образом перенести на сети Петри определения тех свойств, которые были введены для сетей-процессов.

Кононюк А.Е. Графы

Будем говорить, что сеть Петри является *плотной* (K -, L -, M -плотной), если порождаемая ею сеть-процесс является плотной (K -, L -, M -плотной).

Сопоставление сети Петри порождаемой сети-процесса будет рассматриваться как результат некоторого преобразования развертки исходной сети. Поскольку сеть, задающая процесс, может быть в общем случае бесконечной, то процедур развертки сводится к построению некоторого префикса сети-процесса, заканчивающегося указанием не бесконечное повторение ее периодического фрагмента. Мы ограничимся примерами преобразований развертки для подклассов сетей Петри, порождающих сети-процессы, рассмотренные выше.

Заметим, прежде всего, что для сети Петри, являющейся конечной сетью, удовлетворяющей ограничения $A1-A7$, преобразование развертки тривиально: исходная сеть и порождаемая ею сеть-процесс совпадают.

В качестве первого примера преобразования развертки рассмотрим развертку примитивных сетей, которые принадлежат пересечению класса автоматных сетей и регулярных сетей. Напомним, что простой путь в сети представляет собой последовательность элементов сети (x_1, x_2, \dots, x_n) такую, что $x_i F x_{i+1}$ для всех $1 \leq i \leq n$ и $x_i \neq x_j$ для любых двух элементов сети, кроме, быть может, x_1 и x_n . Простой путь называется простым циклом, если $x_1 = x_n$. Простой цикл назовем *регулярным*, если x_1 и x_n - головные и хвостовые места в циклической подсети примитивной сети, т.е. в подсети, для всех элементов x, y которой верно, что $x F^* y \wedge y F^* x$.

Назовем *циклическим компонентом* примитивной сети N ее максимальную циклическую подсеть N' такую, что

$$\forall y \in X \setminus X', \exists x \in X': \neg (x F^* y \wedge y F^* x).$$

Любой циклический компонент представим в виде формулы сети, начинающейся операцией итерации $*$.

Максимальная подсеть сети N , не содержащая внутри ни одного циклического компонента, образует *ациклический компонент*.

Легко видеть, что любую примитивную сеть N можно представить как конечную чередующуюся последовательность циклических и ациклических компонентов: $N - (N_1; N_2; \dots; N_n)$, где $n \geq 1$, и для любого $1 \leq i < n$, если N_i - циклический (ациклический) компонент, то N_{i+1} - ациклический (циклический) компонент.

Кононюк А.Е. Графы

Пусть $N = (P, T, F, M_0)$ - исходная сеть Петри; условимся в порождаемой сети-процессе $\tilde{N} = (\tilde{P}, \tilde{T}, \tilde{F}, \tilde{M}_0)$ использовать в качестве элементов множества $X = P \cup T$ символы элементов из $X = P \cup T$ с дополнительными верхними индексами.

Рассмотрим развертку примитивной сети со стандартной начальной разметкой M_0 , при которой только единственное головное место p сети имеет разметку $M_0(p) = 1$, а остальные места имеют нулевую разметку. На рис. 10, а показана примитивная сеть $N = 1a; *(b; *c; d); e$, на рис. 10, б сеть представлена как комбинация компонентов $(N_1; N_2; N_3)$, где N_1 и N_2 - ациклические компоненты, $N_3 = *(b; *c; d)$ - циклический компонент.

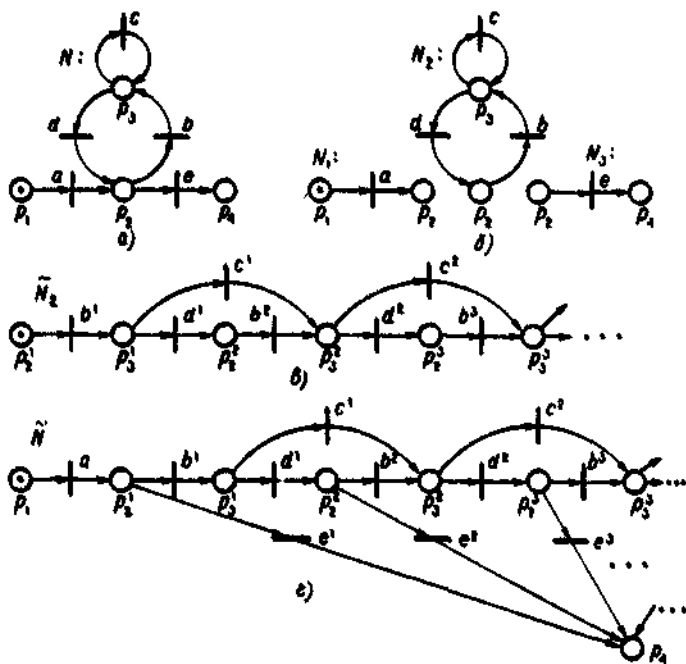


Рис. 10

При развертке такой примитивной сети сначала отдельно разворачиваются все ее циклические компоненты. В каждом циклическом компоненте N_i имеется единственное головное (и хвостовое) место p_i (в компоненте N_2 на рис. 10, б такое место - место p_2).

Головному месту p_i компонента N_i сопоставляется головное место p^i_i

Кононюк А.Е. Графы

развернутой сети \tilde{N}_i . Далее развертка осуществляется индуктивным способом.

Пусть p - некоторое место в компоненте N_i и ему соответствует место p^j в построенной части сети \tilde{N}_i , причем j - максимальный индекс среди верхних индексов копий места p в построенной части \tilde{N}_i . Если

$p = \{t_1, \dots, t_n\}$ в N_i , то в \tilde{N}_i добавляются вершины-переходы t^1, \dots, t^n (если их еще нет в \tilde{N}_i) и из места p^j заводятся дуги на каждый из новых добавленных переходов.

Пусть t - некоторый переход в циклическом компоненте N_i и ему соответствует в построенной части сети \tilde{N}_i переход t^j , где j - максимальный верхний индекс среди верхних индексов копий перехода t в построенной части \tilde{N}_i . Здесь возможны два продолжения развертки.

1) Если место $q = t$ не является головным местом некоторого регулярного цикла в N_i , то просматривается построенная часть \tilde{N}_i в поисках места q^j . Если такое место найдено, то оно объявляется выходным местом перехода t^j , в противном случае добавляется новое место q^j с верхним индексом j и на него заводится дуга от перехода t^j .

2) Если место $q = t$ является головным местом некоторого регулярного цикла в N_i , содержащего t , то просматривается построенная часть \tilde{N}_i в поисках места q^{j+1} . Если такое место найдено, то оно объявляется выходным местом перехода t^j , в противном случае добавляется новое место q^{j+1} и на него заводится дуга от перехода t^j .

На рис. 10, *б* показана развертка \tilde{N}_2 циклического компонента N_2 сети N , изображенной на рис. 10, *а*.

После развертки отдельных циклических компонентов осуществляется их стыковка с ациклическими компонентами, если таковые имеются в исходной сети. Пусть $N = (N_1; N_2; \dots; N_n)$. Для стыковки N_i и \tilde{N}_{i+1} , в том случае, когда N_i - ациклический компонент, а \tilde{N}_{i+1} - развертка циклического компонента, выделяется единственное общее "стыковочное" место p , которое является хвостовым местом в N_i , и головным в N_{i+1} , и затем p в N_i и p^1 в \tilde{N}_{i+1} сливаются.

Кононюк А.Е. Графы

Для стыковки \tilde{N}_i с N_{i+1} в том случае, когда \tilde{N}_i - развертка циклического компонента, а N_{i+1} - ациклический компонент, выделяется стыковочное место p , которое является хвостовым местом в N_i и головным в N_{i+1} . После развертки циклического компонента N_i в ациклическую сеть \tilde{N}_i этому месту p в \tilde{N}_i соответствует бесконечная последовательность хвостовых мест p^1, p^2, \dots . При стыковке \tilde{N}_i с N_{i+1} головное место p из N_{i+1} раскопируется в бесконечную последовательность p^1, p^2, \dots вместе с выходными дугами, переходами, на которые эти дуги ведут, и выходными дугами этих переходов, после чего сливаются одинаковые копии мест p^1, p^2, \dots из \tilde{N}_i и из \tilde{N}_{i+1} , где \tilde{N}_{i+1} - это ациклический фрагмент после копирования его головного места.

На рис. 10, з показана S-сеть \tilde{N} , представляющая процесс функционирования примитивной сети N , изображенной на рис. 10, а. Легко видеть, что S-сеть \tilde{N} не является L-плотной, так как в ней имеется li-сечение $\{p_1, a, p_1^1, b^1, p_1^2, a^1, p_1^3, b^2, p_1^4, a^2, p_1^5, \dots\}$, которое не пересекается с al-сечением $\{e_1, e_2, \dots\}$. Поэтому исходная примитивная сеть $N = 1a; *1b; *c; a^1; a$ не является L-плотной.

Процедура развертки примитивной сети со стандартной разметкой сводится, как легко видеть, к выделению всех возможных путей в графе исходной сети и склеиванию путей в тех местах, после которых эти пути совпадают. Поэтому справедливо следующее утверждение:

Теорема 8. *Результатом развертки примитивной сети со стандартной начальной разметкой является S-сеть.*

Пусть \tilde{N} - S-сеть, полученная разверткой примитивной сети N со стандартной начальной разметкой. Если для \tilde{N} ввести пометчающую функцию Σ такую, что $\Sigma(p^j) = t$ для любого j , где $p^j \in \tilde{T}, t \in T$, то можно сформулировать следующую теорему.

Теорема 9. $L(N) = L(\tilde{N}, \Sigma)$, т.е. свободный префиксный язык сети N и префиксный язык помеченной сети (\tilde{N}, Σ) совпадают.

Развертка примитивной сети с произвольной начальной разметкой осуществляется сведением к рассмотренному выше случаю сети со стандартной начальной разметкой. С этой целью производится предварительное преобразование расщепления исходной сети на совокупность примитивных сетей со стандартной разметкой. Расщепление осуществляется следующим образом. Пусть m — сумма

Кононюк А.Е. Графы

фишек в начальной разметке исходной сети. Если $m = 0$, то сеть порождает "пустой" процесс, никак не изображаемый сетями. В противном случае эта сеть копируется в m экземплярах, и каждый из экземпляров сети получает новую начальную разметку по следующему правилу:

- 1) в каждой копии единственное место имеет ненулевую разметку, равную 1;
- 2) если место p в некоторой копии имеет фишку, то $M_o(p) > 0$ в исходной сети;
- 3) общее число копий, имеющих фишку в месте p , в точности равно $M_o(p)$ в исходной сети.

Следующий шаг состоит в индексации переходов и мест в каждом из m экземпляров. Эти экземпляры упорядочиваются произвольным образом и каждый элемент i -го экземпляра ($1 \leq i \leq m$) получает верхний индекс i . На рис. 11, а показана примитивная сеть $2a; *(b; c)$. Начальная разметка этой сети $(2, 0, 1)$ отличается от стандартной. На рис. 11, б исходная сеть преобразована в совокупность трех примитивных сетей, две из которых имеют по фишке в головном месте, а третья содержит фишку во "внутреннем" месте p_3 .

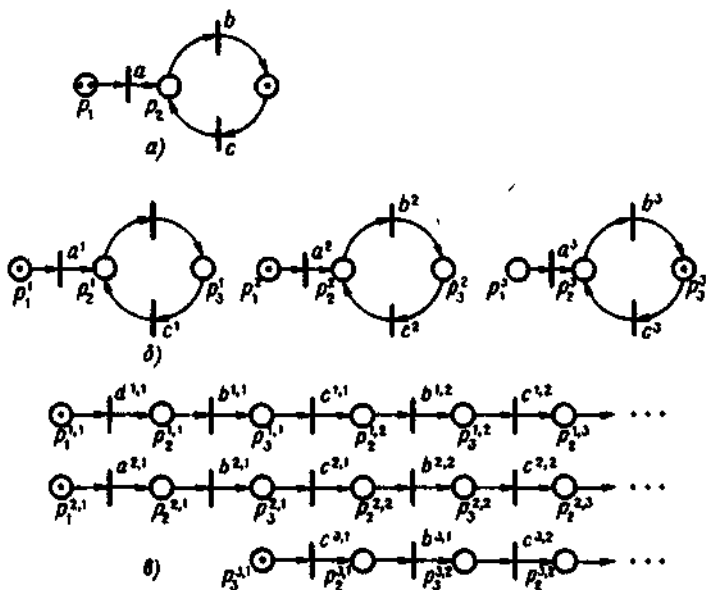


Рис. 11

Каждая из копий разворачивается по правилам развертки примитивной сети со стандартной разметкой, причем переходы и места получают второй верхний индекс. В случае, если в некоторой копии фишка содержится не в головном месте сети, как, например, в третьей копии сети на рис. 11, б, развертка осуществляется, как будто фишка находится в головном месте. Затем фишка помещается в первое вхождение $p^{i,1}$ (в развертку копии) того места p^i , которое содержало фишку в соответствующей i -й копии исходной сети. Наконец, в развертке i -й копии удаляются все те элементы x (и связывающие их дуги), для которых не выполняется отношение $p^{i,1}F^+x$, т.е. те элементы, которые не достижимы в развернутой сети при движении от места $p^{i,1}$ вдоль дуг сети. При этом необходима корректировка вторых верхних индексов оставшихся элементов. Для элемента $p^{i,j}$ берется новый второй индекс $(j-k)$, где k — максимальный индекс удаленного элемента $p^{i,k}$. На рис. 11, в показана сеть, полученная в результате развертки сети на рис. 11, а.

Заметим, что развертка примитивной сети с произвольной начальной разметкой может привести к сети, не являющейся S-сетью, т.е. теорема 8 не верна для этого случая. Например, сеть на рис. 11, в является A-сетью. Для получаемых сетей-процессов выполняется ограничение A10, т.е. каждый переход имеет ровно одну входную и ровно одну выходную дуги, но нарушается требование A9 к S-сети иметь ровно одно головное место. В то же время теорема 9 остается верной и для случая примитивной сети с произвольной разметкой.

Рассмотрим развертку подкласса регулярных сетей, которые мы назовем *параллельными сетями*. Параллельная сеть N представляет собой наложение последовательных сетей $N_1; N_2; \dots; N_n$, каждая из которых является простым путем или простым (регулярным) циклом. Примеры параллельных сетей показаны на рис. 12, в и 13, а.

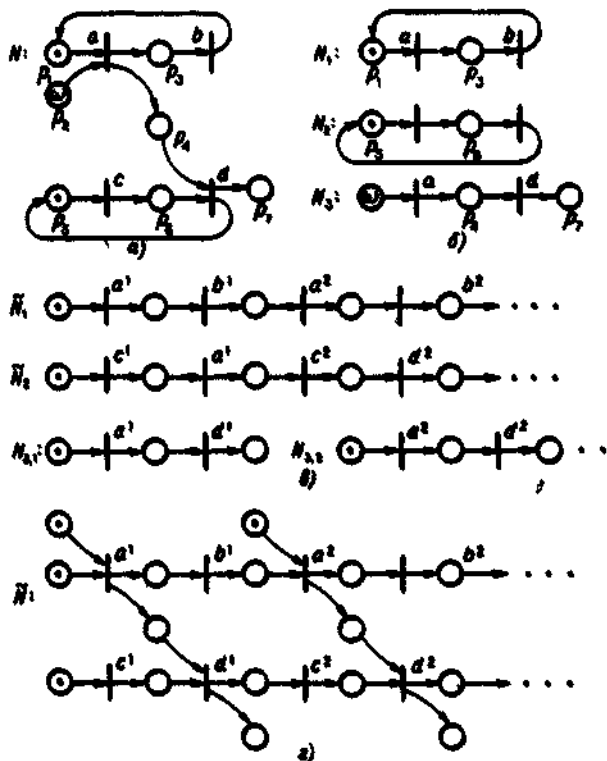


Рис. 13.

Процедура развертки параллельных сетей включает три этапа. На первом этапе в исходной сети $N = (N_1; N_2; \dots; N_n)$ выделяются все ее последовательные составляющие, и каждая из них разворачивается по правилам развертки примитивных сетей с произвольной разметкой (последовательная сеть является частным случаем примитивной сети). На втором этапе осуществляется специальная переиндексация развернутой сети \tilde{N}_i для обеспечения последующего наложения развернутых сетей. На третьем этапе развернутые сети накладываются друг на друга с учетом индексации и образуют развернутую сеть

Кононюк А.Е. Графы

($\tilde{N}_1; \tilde{N}_2; \dots; \tilde{N}_n$), которая затем приводится к "каноническому" виду сети-процесса путем удаления мертвых переходов и связанных с ними дуг и мест.

Пусть N_i - последовательная составляющая параллельной сети и начальная разметка сети N_i помещает в ее места m фишек. После расщепления N_i , на m экземпляров последовательных сетей $N_{i,k}$

$1 \leq k \leq m$, со стандартной разметкой (единственная фишка в каждой из копий) и их развертки в S-сети $\tilde{N}_{i,k}$ производится переиндексация переходов данных сетей следующим образом. На первом шаге новый индекс получают первые переходы в каждой из сетей. Первый переход в первой сети ($\tilde{N}_{i,1}$) получает индекс 1. Если первый переход в k -й сети ($\tilde{N}_{i,k}$) не имеет одноименных среди первых переходов предыдущих $k - 1$ сетей ($\tilde{N}_{i,1}, \dots, \tilde{N}_{i,k-1}$), то он также получает индекс 1, в противном случае данный переход получает индекс, на единицу больший максимального индекса среди одноименных первых переходов в предыдущих сетях $\tilde{N}_{i,1}, \dots, \tilde{N}_{i,k-1}$.

На k -м шаге индексируются k -е переходы в каждой из копий. Аналогично индексированию на первом шаге, переиндексация ведется "сверху вниз" с учетом перехода копий; k -й переход в сети $\tilde{N}_{i,j}$ получает индекс 1, если среди всех ранее переиндексированных переходов нет ему одноименных, в противном случае данный переход получает индекс, на единицу больший максимального индекса среди всех ранее занумерованных одноименных переходов.

После переиндексации все переходы из множества сетей $\{\tilde{N}_{i,k} \mid 1 \leq k \leq m\}$ различны: или переходы имеют разные имена (но могут иметь одинаковый индекс), или если переходы имеют одну и ту же метку, то их индексы (по построению) различны. Все сети $\tilde{N}_{i,k}$

$1 \leq k \leq m$, объединяются в одну сеть \tilde{N}_i путем наложения. При этом ни одна из пар переходов фактически не накладывается, так как все переходы различны.

В случае, если число экземпляров m конечно (сеть N_i не содержит разметки с ω) и N_i не является циклом, процесс переиндексации заканчивается за конечное число шагов. В противном случае процедура переиндексации применяется к конечным префиксам разверток циклических сетей.

На рис. 12, а показана параллельная сеть N , представляющая собой наложение двух последовательных сетей N_1 и N_2 (рис. 12, б). Начальная

Кононюк А.Е. Графы

разметка сети N_1 помещает в ее места 3 фишки, сети N_2 - 2 фишки, поэтому сеть N_1 расщепляется на три последовательные сети $N_{1,1}$, $N_{1,2}$, $N_{1,3}$ со стандартной разметкой, сеть N_2 - на две сети $N_{2,1}$, $N_{2,2}$. На рис. 12, в и з показаны сети \tilde{N}_1 и \tilde{N}_2 после переиндексации переходов и объединения разверток расщеплений в одну сеть.

После того как все составляющие сети расщеплены, развернуты и переиндексированы, производится "выравнивание" числа накладываемых переходов. Если в сети \tilde{N}_i максимальный индекс у перехода t равен l_1 , а в сети \tilde{N}_j максимальный индекс у того же перехода равен l_2 и $l_2 < l_1$, то в сеть \tilde{N}_j добавляются $l_1 - l_2$ копии сети N_j с нулевой разметкой всех мест. При этом все вхождения перехода t индексируются описанным выше способом, начиная с индекса $(l_2 + 1)$. Такое выравнивание числа вхождений переходов производится для всех переходов, входящих одновременно в сети N_i и N_j , и для всех пар сетей, составляющих сеть N

На рис. 12, д показана сеть \tilde{N}_2 после выравнивания числа вхождений перехода c . В сеть \tilde{N}_2 добавлена "пустая" копия сети N_2 , и переход c получил верхний индекс 3.

Предпоследний этап состоит в построении сети $\tilde{N} = (\tilde{N}_1, \tilde{N}_2, \dots, \tilde{N}_n)$ путем объединения выровненных разверток составляющих сетей N_1, N_1, \dots, N_n в одну сеть с помощью операции наложения. При выполнении операции наложения вхождения одинаковых символов переходов с одинаковыми верхними индексами совмещаются по общим правилам операции наложения. (Заметим, что здесь операция наложения обобщена на случай бесконечных сетей.)

На рис. 12, в показана сеть $\tilde{N} = (\tilde{N}_1, \tilde{N}_2)$, полученная наложением сетей \tilde{N}_1 и \tilde{N}_2 .

Последний этап состоит в приведении полученной в результате сети к форме, удовлетворяющей требованиям; предъявляемым к сетям-процессам. Дело в том, что в результате наложения в результирующей сети могут появиться головные места без фишек. "Чистка" полученной сети состоит в удалении из сети таких головных мест и всех тех элементов сети, к которым можно прийти из головных мест по путям вдоль дуг. На рис. 12, ж показана сеть $\tilde{N} = \tilde{N}_1, \tilde{N}_2$) после чистки. Эта сеть является О-сетью и представляет параллельный процесс функционирования параллельной сети, изображенной на рис. 12, а. На рис. 13 показана параллельная сеть с циклами, О-сеть,

Кононюк А.Е. Графы

являющаяся результатом развертки предыдущей сети, и некоторые промежуточные этапы развертки. O -сеть на рис. 13, z не является K -плотной и поэтому не K -плотна и параллельная сеть на рис. 13, a (она является регуляризованной версией синхронизационного графа). Причиной не- K -плотности этой сети служит неограниченность мест p_4 и p_7 .

Анализ описанного преобразования развертки параллельных сетей позволяет сформулировать следующие два утверждения:

Теорема 10. *Результатом развертки параллельной сети является O -сеть.*

Теорема 11. $L(N) = L(\tilde{N}, \Sigma)$, где UN - свободный язык помеченной сети-развертки, в которой $\forall t^j: \Sigma(t^j) = t$.

Процедура развертки произвольной регулярной сети здесь не описывается. Она состоит из тех же этапов, что и процедура развертки параллельной сети, но этап переиндексации усложняется и требует топологических преобразований разверток составляющих примитивных сетей. На рис. 14 приведены примеры регулярных (и легко регуляризуемых) сетей, которые разворачиваются в A -сети, рассмотренные в предыдущих параграфах. Например, сеть на рис. 14, a не является K -плотной, так как разворачивается в не- K -плотную O -сеть, структура которой показана на рис. 3 (с точностью до именованых переходов); сеть на рис. 14, b разворачивается в L -плотную S -сеть на рис. 5, a ; сеть на рис. 14, b разворачивается в не- L -плотную S -сеть на рис. 5, b ; что же касается сети, изображенной на рис. 14, d , то она не является ни K -, ни L -, ни M -плотной, так как такими свойствами обладает A -сеть на рис. 9, b и эта A -сеть является разверткой предыдущей сети.

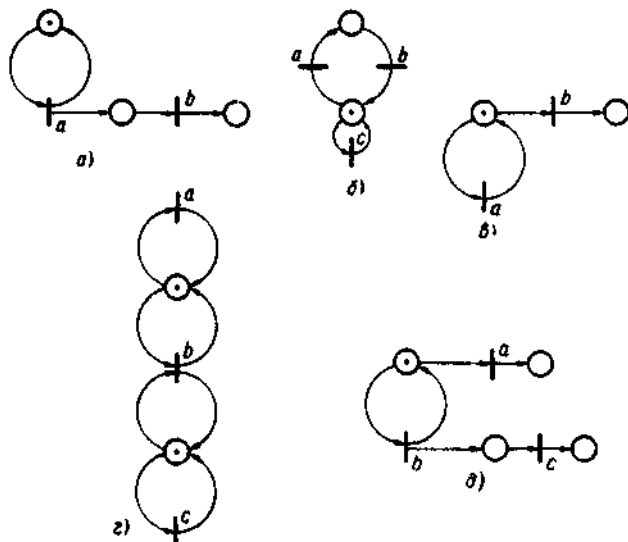


Рис.14

2.11.6. Сетевое представление параллельных процессов с конкуренцией

Сети, предназначенные для описания параллельных процессов с конкуренцией, строятся на основе О-сетей с добавлением в них специальных мест, называемых ресурсами, и дуг, связывающих эти места с переходами особым способом. В сети $N = (P, T, F)$, удовлетворяющей условиям А1— А8, назовем *интервалом* подсеть $I = (P_I, T_I, F_I)$, для которой выполнены следующие условия:

- 1) имеется переход $t_b \in T$, который служит началом интервала;
- 2) имеется переход $t_e \in T$, который служит концом интервала, причем

$$t_b F^* t_e \vee t_b = t_e;$$

$$3) \tilde{T}_I = \{t \in T \mid t_b F^* t \wedge t F^* t_e\},$$

т.е. множество переходов интервала включает все переходы сети, в которые имеется путь от перехода t_b и одновременно от которых имеется путь в сети до перехода t_e ;

$$4) P_I = \{p \in P \mid t_b F^* p \wedge p F^* t_e\},$$

т.е. множество мест интервала включает все места сети, в которые есть путь пографу сети от перехода t_b и от которых есть путь до перехода t_e ;

$$5) F_I = F \cap (P_I \times T_I \cup \tilde{T}_I \times P_I);$$

Кононюк А.Е. Графы

б) для любого $x \in (P \cup T) \setminus (P_l \cup T_l)$ не существует пути в сети N от x до $y \in (P_l \cup T_l)$, не содержащего перехода t_b , и не существует пути от y до x , не содержащего перехода t_e .

Таким образом, интервал образует подсеть, которую в теории графа называют гамаком, причем при движении по графу внутрь гамака нельзя попасть, минуя его начало, а из гамака нельзя выйти, минуя его конец. Если $t_b = t_e$, то интервал состоит из единственного перехода. Интервал I сети N с началом t_b и концом t_e будем обозначать $I(t_b, t_e)$. Тот факт, что I является интервалом сети N , будем обозначать $I \subseteq N$. Ясно, что в любой сети имеются тривиальные интервалы, состоящие из единственного перехода. Сеть на рис. 4,а имеет единственный нетривиальный интервал с началом t_l и концом t . Этот интервал является бесконечной подсетью. В сетях на рис. 4, б, 2 и 3 нет нетривиальных интервалов. Можно показать, что в K -плотной O -сети любой интервал конечен.

Сетью действий с конкуренцией (или *OC-сетью*) назовем сеть вида $N = (P_d \cup P_c, T, F_d \cup F_c, M_0)$, в которой:

- 1) P_c — множество мест, называемых ресурсами, и $P_d \cap P_c = \emptyset$;
- 2) $F_d \subseteq (P_d \times T \cup T \times P_d)$, $F_c \subseteq (P_c \times T \cup T \times P_c)$;
- 3) подсеть $N_d = (P_d, T_d, F_d, M_0(P_d))$ представляет собой O -сеть;
- 4) $\forall p \in P_c: M_0(p) = 1$, т.е. все места-ресурсы из P_c имеют единичную разметку;
- 5) $\forall p \in P_c: |p| = |p'| > 1$, т.е. из каждого места-ресурса выходит столько же дуг, сколько и входит, но всегда ресурс имеет минимум по две входной и по две выходной дуги;
- 6) $\forall p \in P_c: (\forall t_1 \in p, \exists t_2 \in p' : I(t_1, t_2))$ - интервал в подсети действий N_d и $\forall t_2 \in p, \exists t_1 \in p' : I(t_1, t_2)$ — интервал в подсети действий N_d .

На рис. 15,а показан пример OC -сети.

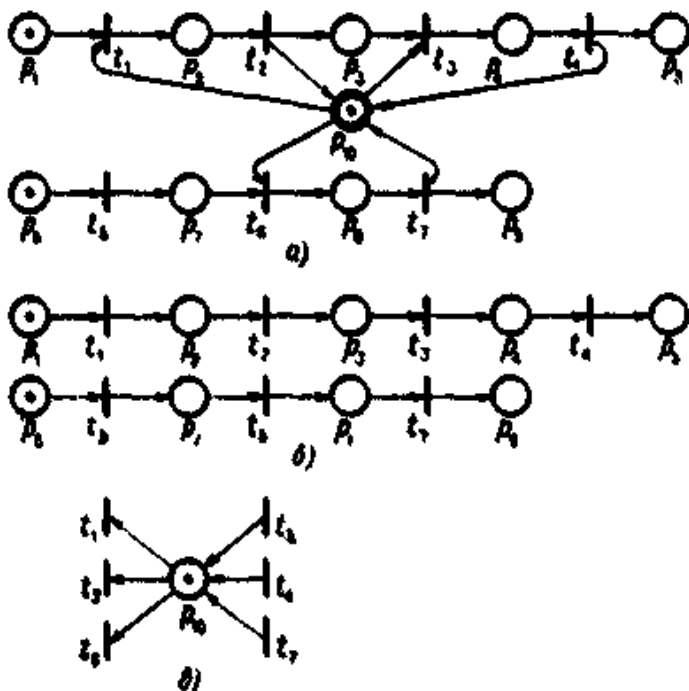


Рис. 15

Из определения ОС-сети видно, что она представляет собой наложение на О-сеть "синхронизирующей" сети специального вида, в которой множество мест совпадает с множеством P_c , каждый выходной переход места-ресурса p и P_c является началом некоторого интервала в О-сети, для этого перехода имеется парный входной для p переход, который служит концом этого интервала. На рис. 15, б и в показано разложение ОС-сети на составляющую ОС-сеть и синхронизирующую сеть. (Операция наложения была введена ранее для конечных сетей. Мы уже отмечали, что она естественным образом может быть расширена на случай бесконечных сетей.)

Если в разных интервалах их начала являются выходными переходами для некоторого ресурса, а их концы - входными для того же ресурса, то эти интервалы будем называть критическими интервалами по данному ресурсу. Если $\{I_1, \dots, I_n\}$ - множество интервалов, каждая пара которого является парой критических интервалов по ресурсу p , то будем обозначать этот факт следующим образом: $\|I_1, \dots, I_n\|p$.

Кононюк А.Е. Графы

Единственным ресурсом в сети N на рис. 1, a служит место p_{10} . Интервалы $I(t_1, t_2)$, $I(t_3, t_4)$ и $I(t_6, t_7)$ - критические интервалы по p_{10} .

Свяжем отношения, с помощью которых в 2.11.1 вводилось понятие параллельного процесса с конкуренцией, с отношениями в ОС-сетях. Как обычно, действия представлены переходами, а изменения условий - местами, не являющимися ресурсами. Таким образом, места-ресурсы играют вспомогательную роль и не интерпретируются как изменения условий. Поэтому, если для остальных мест ОС-сети остается актуальным правило, требующее, чтобы место изменяло свою разметку лишь один раз, на ресурсы это правило не распространяется. Отношения li , so и con следующим образом выражаются через сетевые отношения.

Пусть $N = (P_d \cup P_c, T, F_d \cup F_c, M_0)$ - ОС-сеть, а элементы x и y принадлежат множеству $X_d = P_d \cup T$:

$$x \text{ li } y \iff (x F_d^+ y \vee y F_d^+ x) \vee (x = y),$$

$$x \text{ con } y \iff \exists p \in P_c, \exists I_1, I_2 \in N_d : x \in I_1 \wedge y \in I_2 \wedge I_1 \parallel I_2 \parallel p,$$

$$x \text{ so } y \iff \neg (x \text{ li } y \vee x \text{ con } y) \vee (x = y).$$

Таким образом, отношение li связывает те и только те элементы ОС-сети, которые соединены некоторым путем в подсети действий N_d . Отношение con связывает те и только те элементы, которые принадлежат двум разным критическим интервалам по некоторому ресурсу из P_c . Отношение so связывает те элементы, которые не соединены никаким путем в подсети N_d (но их может соединить путь в сети N , проходящий через некоторый ресурс) и они не принадлежат разным критическим интервалам по любому из ресурсов из P_c .

В случае ОС-сетей также возникает вопрос, всякая ли ОС-сеть является корректным представлением параллельного процесса с конкуренцией. Рассмотрим ОС-сеть N , показанную на рис. 16, a .

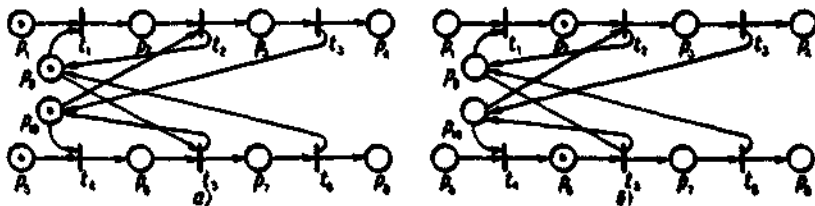


Рис. 16.

В этой сети имеются два критических интервала $I(t_1, t_2)$ и $I(t_5, t_6)$ по ресурсу p_9 и два критических интервала $I(t_2, t_3)$ и $I(t_4, t_5)$ по ресурсу p_{10} . Эта сеть полностью соответствует определению ОС-сети. В ней

имеются последовательные переходы (например, $t_1 \text{ li } t_3, t_5 \text{ li } t_6$), параллельные переходы (например, $t_1 \text{ со } t_4, t_3 \text{ со } t_6$), и конкурирующие переходы (например $t_1 \text{ со } t_5, t_1 \text{ со } t_6, t_3 \text{ со } t_4$). Однако при работе сети N может возникнуть ситуация, показанная на рис. 16,б, когда ни один из переходов t_1, t_3, t_5, t_6 не может сработать. Это означает, что перечисленные переходы соответствуют действиям, которые могут не реализоваться, что противоречит определению параллельного процесса с конкуренцией, в котором любое действие происходит ровно один раз. Причина возникновения такой ситуации состоит в том, что интервал $I(t_1, t_2)$ и $I(t_2, t_3)$ пересекаются по переходу y_2 , а интервалы $I(t_4, t_5)$ и $I(t_5, t_6)$ пересекаются по переходу t_5 . При этом пересечение устроено таким образом, что пересекающиеся две пары интервалов являются критическими по разным ресурсам, и начало интервала $I(t_2, t_3)$ входит в интервал $I(t_1, t_2)$ как внутренний (не начальный) переход, а начало интервала $I(t_5, t_6)$ выходит в интервал $I(t_4, t_5)$ как внутренний переход. Мы ограничимся введением ОС-сетей как сетевых представлений параллельных процессов с конкуренцией, не обсуждая их связи с сетями Петри, моделирующими системы, которые порождают такие процессы.

2.12. Сети Петри и программирование

Среди приложений теории сетей Петри к задачам моделирования дискретных систем наибольшее развитие получили работы, связанные с попытками использовать аппарат сетей Петри, их модификации и обобщения, для описания и изучения структурной динамики программ, в первую очередь - так называемых параллельных программ.

В этом разделе будет изложен один из подходов к применению сетей Петри для решения задач параллельного программирования, опирающийся на введенную ранее алгебру регулярных и иерархических сетей, элементы которых интерпретируются специальным образом элементами параллельных программ.

2.12.1. Сети Петри и семантика структур управления

Под структурой управления программы понимают совокупность базовых структурных единиц - операторов (а также модулей, подмодулей) — и специальных управляющих примитивов, позволяющих в процессе

Кононюк А.Е. Графы

исполнения программы формировать сложные вычислительные процессы из более простых, задаваемых упомянутыми структурными единицами. К числу управляющих примитивов относятся, например, операторы управления в алгоритмических языках, такие как условные операторы, операторы перехода, операторы цикла и т.п. К управляющим примитивам следует отнести и специальные разделители, явно или неявно указывающие порядок следования операторов.

Рассмотрим основные типы структур управления в программах и методы их представления с помощью сетей Петри и их обобщений.

Обычная для алгоритмических языков последовательная структура управления представляет собой совокупность операторов различных типов, которые связаны некоторым отношением следования. Начальный оператор (единственный в программе) не следует ни за каким-либо другим оператором. После безусловного оператора следует ровно один оператор. После условного оператора следует два оператора (используется также оператор выбора, за которым следует некоторая совокупность операторов). После заключительного оператора не следует никакой другой оператор. Исполнение программы с последовательной структурой управления начинается с начального оператора. После завершения исполнения безусловного оператора начинается выполнение следующего за ним оператора. После завершения исполнения условного оператора начинается исполнение одного из двух следующих за ним операторов в зависимости от значения логического условия, вычисленного в условном операторе. Исполнение программы завершается исполнением заключительного оператора.

Последовательная структура управления может быть представлена как связанная автоматная сеть Петри с определенным топологическим ограничением, элементы которой интерпретируются специальным образом.

Такая сеть имеет единственное головное место, все ее переходы делятся на "безусловные" с одним выходным местом и "условные" с двумя выходными местами. Каждый "безусловный" переход сети интерпретируется как безусловный оператор, каждый "условный" переход интерпретируется как условный оператор. Если в последовательной программе оператор a' следует за оператором a , то в сети соответствующие переходы связаны дугами с общим местом, которое является выходным для a и входным для a' .

Например, структура управления следующей программы (вычисление факториала) представляется с помощью интерпретированной сети Петри, изображенной на рис. 1:

Кононюк А.Е. Графы

```
begin integer x, y;  
  input(x);  
  y := 1;  
l 1: if x = 0 then goto l 2;  
  y := y * x;  
  x := x - 1;  
  goto l 1;  
l 2: output(y)  
  stop end
```

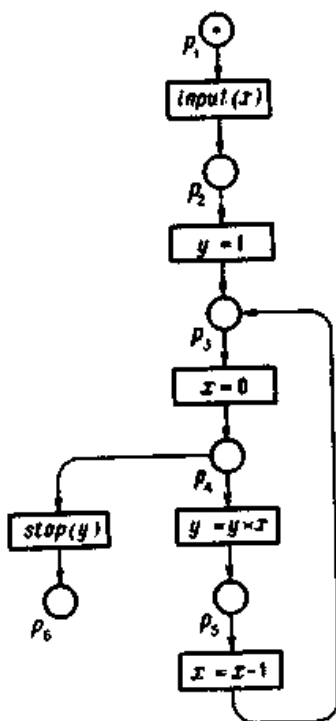


Рис. 1

Переходы в интерпретированной сети изображены прямоугольниками, внутри которых выписаны сопоставленные переходам операторы. Интерпретированная сеть на рис. 1 описывает только структуру последовательного управления, но не моделирует сам механизм

Кононюк А.Е. Графы

управления. Действительно, в терминах сети Петри нельзя описать семантику условного оператора. Поэтому, когда место p_4 получает фишку, то эта фишка сообщает лишь о том, что условный оператор выполнен, но не несет информацию о вычисленном значении условия. В результате выбор оператора для продолжения исполнения последовательной программы моделируется в сети недетерминированной альтернативой (оператор *output* (y) или оператор $y: = y \times x$), указывающей оба возможных варианта продолжения. Для того чтобы иметь средства моделирования условного управления, достаточно выбрать одно из описанных ранее обобщений сети Петри, например, ингибиторную сеть, и интерпретировать переходы такой сети операторами программы.

Последовательно-параллельные структуры управления представляют собой обобщение последовательных за счет включения в набор управляющих примитивов специальных операторов или указателей, которые выделяют *параллельные ветви* программы, исполняемые независимо друг от друга. Ветви имеют общее начало — точку расхождения ветвей — и конец — точку схождения ветвей. Совокупность ветвей с общим началом и концом образует *сегмент* параллельной структуры. При исполнении программы процесс вычислений продвигается до начала первого сегмента, после чего "расщепляется" на столько копий, сколько ветвей содержит сегмент. Каждый из параллельных процессов вычислений ветвей протекает независимо от других и, достигая конца ветви, останавливается, ожидая, пока все остальные процессы в сегменте не достигнут конца сегмента. В конце сегмента все процессы "сливаются" в один. Параллельные ветви сегментов могут, в свою очередь, содержать сегменты.

В последовательно-параллельных языках программирования для выделения параллельных ветвей используются специальные программные примитивы, такие как скобки параллельности `parbegin ... parend`, `cobegin ... coend` или специальные операторы, отмечающие начало и конец параллельного сегмента. Примером могут служить операторы `fork (a, b, c,...)` и `join (a, b, c, ...)`. Первый оператор открывает сегмент с параллельными ветвями, помеченными метками a, b, c, \dots , второй — замыкает сегмент с ветвями, помеченными метками a, b, c, \dots .

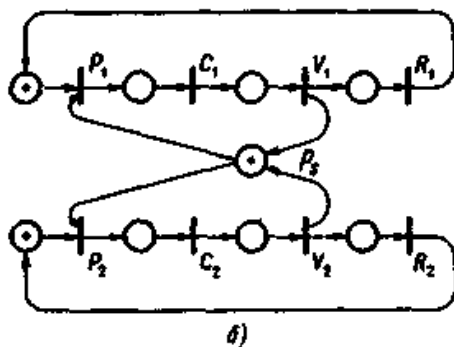
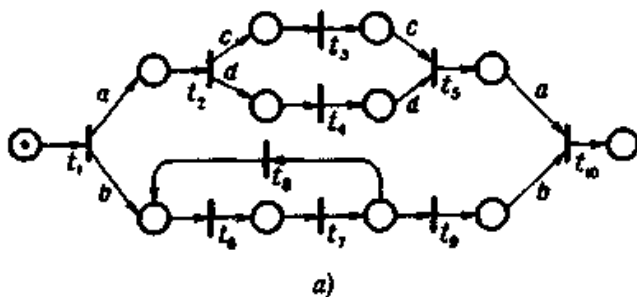


Рис. 2

На рис. 2,а показан пример последовательно-параллельной структуры, изображенной с помощью сети Петри. Переход t_1 интерпретируется как оператор *fork* (a, b), инициирующий две параллельные ветви, помеченные метками a и b , переход t_2 — как оператор *fork* (c, d), переход t_5 соответствует оператору *join* (c, d), переход t_{10} — оператору *join* (a, b).

Характерным свойством "чистой" последовательно-параллельной структуры управления является отсутствие каких-либо взаимодействий между параллельными ветвями, переходов-скачков из одной ветви в другую. Такие структуры управления обладают существенным недостатком — малой гибкостью управления. В реальных условиях существует необходимость организации взаимодействий параллельно протекающих процессов исполнения ветвей. Эти взаимодействия

Кононюк А.Е. Графы

нужны для организации обменов между ветвями результатами промежуточных вычислений и для обеспечения раздельного доступа к общим ресурсам ветвей, таким как общие области рабочей памяти, общие внешние устройства и т. п.

Для организации взаимодействий нужны дополнительные программные средства, обеспечивающие саму организацию взаимодействий и правильность их реализации в процессе исполнения параллельных ветвей. В большинстве случаев нужны средства, которые позволяли бы приостанавливать процесс исполнения некоторой ветви, пока не придут данные от другой ветви или пока другие ветви не освободят общий ресурс. Тот отрезок ветви, на котором требуется "захват" общего ресурса, называют *критическим интервалом* ветви.

Дейкстра предложил для этих целей механизм семафоров, включающий специальные переменные нового типа (*семафоры*) и две операции P и V , аргументами которых могут быть только переменные типа семафоров. Область значений семафора - целые неотрицательные числа. Если область значений сужена до двух — 0,1, семафор называют бинарным. Операция V изменяет значение s семафора на $s + 1$. Действие операции P определяется следующим образом:

- если $s \neq 0$, то P уменьшает значение s на 1,
- если $s = 0$, то P не изменяет значение s и не завершается до тех пор, пока некоторая другая ветвь не изменит значение s с помощью операции V .

Существенным является тот факт, что операции P и V считаются "неделимыми". По отношению к V это означает следующее. Операция V состоит из трех фраз:

- считывание значения семафора из памяти,
- увеличение значения семафора,
- помещение нового значения в память.

Неделимость $V(s)$ заключается в том, что с самого начала выполнения этой операции до ее завершения доступ к переменной-семафору s запрещен для всех других операций. Аналогично дело обстоит с операцией P . Обычно P предшествует критическому интервалу ветви, а V завершает его. В каждый момент времени, когда значение семафора s изменяется с 0 на 1, только одна из операций может завершиться и разрешить вход в критический интервал только одному процессу исполнения ветви.

Ниже следует пример параллельной программы на алголоподобном языке, дополненном скобками `parbegin ... par end`, выделяющими сегмент с двумя параллельными ветвями *process 1* и *process 2*, разделенными запятой, и семафором *mutex*. Каждая из параллельных

Кононюк А.Е. Графы

ветвей представляет собой последовательный цикл, содержащий критический интервал (*critical section*) и остальную часть цикла (*remainder of cycle*):

```
.begin  
  semaphore mutex;  
  mutex := 1;  
parbegin  
  process 1:  
    begin L1: P(mutex);  
      critical section 1;  
      V(mutex);  
      remainder of cycle 1; goto L1 end ,  
  process 2:  
    begin L2: P(mutex);  
      critical section 2;  
      V(mutex);  
      remainder of cycle 2; goto L2 end  
parend  
end.
```

В этом примере решается задача взаимного исключения исполнения критических интервалов двух циклических процессов параллельных ветвей.

Последовательно-параллельная структура управления, дополненная семафорным механизмом, наглядно изображается с помощью сетей Петри. На рис. 2, б показана сеть, моделирующая структуру программы с взаимно исключенными критическими интервалами двух циклических ветвей. Переход P_i соответствует операции P в ветви i ($i = 1, 2$); переход C_i - критическому интервалу ветви i ; V_i - операции V ветви i ; R_i - остатку цикла в ветви i . Место p_s соответствует семафору s . Механизм семафоров обеспечивает организацию сложных взаимодействий между параллельными ветвями и широко используется в языках программирования. Тем не менее при программировании сложных взаимодействий в параллельных программах число потенциальных ошибочных ситуаций возрастает по сравнению с последовательными программами. Наряду с обычным зацикливанием выражений наиболее частыми ошибками являются

Кононюк А.Е. Графы

взаимные блокировки и отталкивания процессов исполнения параллельных ветвей.

Взаимная блокировка {deadlock} возникает, например, в следующей ситуации. Пусть две ветви *A* и *B* требуют при своей работе доступа к двум общим ресурсам. Типичная ошибочная ситуация, в которой может возникнуть блокировка, показана с помощью сети Петри на рис. 3.

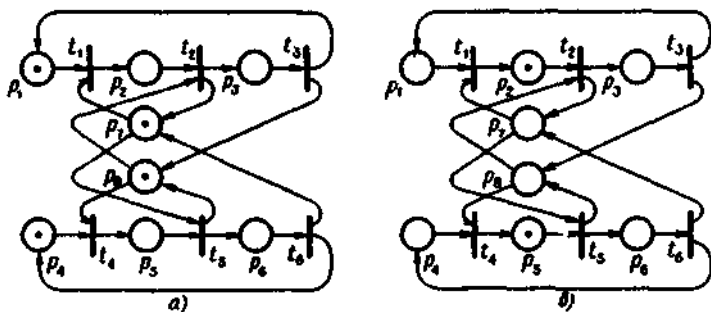


Рис. 3

Пусть процесс исполнения ветви *A* (переходы t_1, t_2, t_3) после выполнения действия t_1 захватил один из ресурсов, изображаемый фишкой в месте p_7 , после чего место p_7 имеет разметку 0. Процесс исполнения ветви *B* (переходы t_4, t_5, t_6) после выполнения действия t_4 захватил другой ресурс (место p_8 имеет разметку 0). Оба процесса остановились, *A* перед t_2 , *B* перед t_5 , так как для выполнения следующих действий каждому из них нужен второй ресурс, захваченный партнером. При этом ни один из процессов не может вернуть захваченный ресурс. Возникла блокировка, показанная на рис. 3.

Характерной задачей, неправильное решение которой может привести как к блокировке, так и к отталкиванию, является задача о пяти обедающих философях, рассматриваемая ниже в 2.12.3.

Рассмотренные выше структуры управления параллельных программ мы охарактеризовали как последовательно-параллельные: основу их составляют параллельно исполняемые последовательные ветви. Если из последовательно-параллельной программы удалить дополнительные средства распараллеливания, то программа становится обычной последовательной. Таким образом, можно считать отправной точкой такого способа организации параллельных программ последовательно-алгоритмическую структуру управления.

Противоположный подход состоит в том, что все операторы программы изначально считаются независимыми, параллельно

Кононюк А.Е. Графы

исполняемыми, а формирование вычислительных процессов организуется с помощью ограничений, накладываемых на порядок исполнения действий.

Метод параллельного программирования, в котором на независимые операторы накладываются ограничения на порядок их выполнения в форме явно указываемых или неявно подразумеваемых индивидуальных условий готовности, связанных с каждым оператором, получил название *асинхронного программирования*. Условия готовности динамически проверяются и разрешают или не разрешают (но не предписывают) начать выполнение операторов, с которыми данные условия связаны. Условия готовности берут на себя всю организацию управления, так что в асинхронных программах можно не разделять средства управления на средства, формирующие последовательности операторов и параллельные ветви, и на средства синхронизации. Если программа имеет иерархическую структуру, то каждый составной оператор может быть организован внутри по тому же асинхронному принципу.

Из сказанного следует, что асинхронный принцип организации вычислений является, в некотором смысле, дополнительным по отношению к последовательно-параллельному и оба принципа эквивалентны в том смысле, что могут моделировать друг друга (при условии, что последовательно-параллельная программа содержит средства синхронизации параллельных ветвей). Относительные достоинства и недостатки этих дополняющих друг друга методов программирования зависят от внутренней структуры программируемых задач и от свойств, реализующих программы вычислительных систем.

Условия готовности операторов в асинхронных программах могут формулироваться в разных терминах. В зависимости от этого можно выделить разные типы асинхронного управления.

Событийное управление основано на том, что условия готовности формулируются как логические функции от некоторых *событий*. Событием может быть инициирование или завершение какого-либо оператора программы (программные события), прерывание в системе или сигнал об освобождении некоторого ее ресурса (системные события).

При *потокном управлении* действие (оператор или операция) программы может выполняться, если готовы все необходимые для него аргументы (операнды). Условие готовности в этом случае носит стандартный характер и не выписывается явно в программе, а неявно подразумевается. В *обратном потокном управлении* действие может выполняться, если его результат необходим в качестве аргумента для

Кононюк А.Е. Графы

некоторого другого действия. В этом случае второе действие как бы вызывает первое в качестве процедуры.

В *динамическом управлении* условие готовности является логическим выражением, зависящим от переменных той же памяти, над которой определены операторы программы.

Ниже мы рассмотрим более подробно особенности событийного и потокового асинхронного управления. Их семантика удобно и наглядно описывается в терминах асинхронных сетей - сетей Петри и их модификаций. При этом специфика определенного типа управления будет в основном отражена в интерпретации переходов, мест и фишек сетей. В реальных программах события могут иметь различный характер. Мы ограничимся единственным типом событий — завершениями действий, а под действиями будем понимать исполнение операторов программы. В этом случае семантика событийного управления может быть представлена с помощью сети Петри, элементами которой интерпретируются следующим образом. Переходам сопоставлены операторы, места хранят информацию о событиях. Событие может не произойти, произойти, повториться. Соответствующее место не имеет фишки, имеет одну фишку или несколько фишек. Условие срабатывания перехода трактуется как условие готовности соответствующего оператора. Срабатывание перехода изменяет разметку его входных мест, и тем самым формируется информация о происшедшем событии — исполнении соответствующего оператора.

Механизм событийного управления, базирующийся на формализме сетей Петри, позволяет описывать более сложные управляющие ситуации, чем механизм последовательно-параллельного программирования с семафорами, и, следовательно, порождать более богатые множества эквивалентных вычислительных процессов. Так, на рис. 4 представлена с помощью сети Петри структура управления, которую не удастся запрограммировать с помощью обычных семафоров без использования искусственно введенных условных операторов.

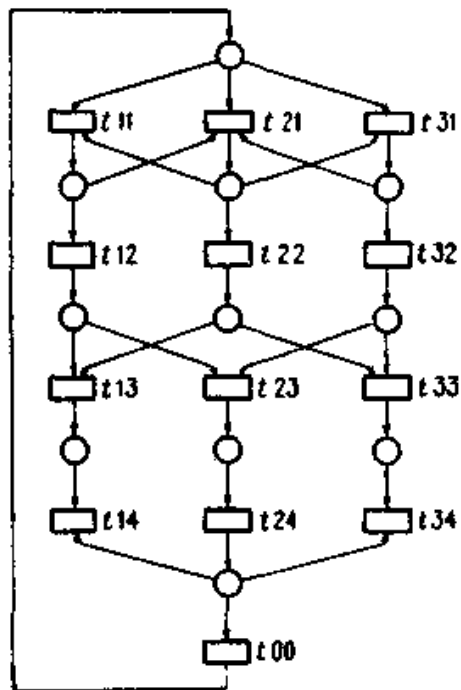


Рис. 4

В этом примере каждый из трех операторов t_{i3} , где $i=1, 2, 3$, может выполняться только после того, как выполнятся операторы t_{j2} и t_{k2} , где $j=1, k=2$ для $i=1$, $j=1, k=3$ для $i=2$ и $j=2, k=3$ для $i=3$. Причина, по которой с помощью семафоров нельзя запрограммировать такую структуру управления, состоит в том, что неделимая операция P применяется только к одному семафору.

Для того чтобы аппарат сетей Петри применять не только для описания семантики событийного управления, но и непосредственно использовать в языке параллельного программирования, разработаны аналитические методы представления сетей, которые позволяют задавать структуру асинхронного управления программы в виде управляющих выражений. Эти выражения трактуются как формулы алгебры регулярных и иерархических сетей, рассмотренной ранее. Управляющие выражения воспринимаются управлением системы, исполняющей программу, и управление организует вычислительный процесс, задаваемый этим выражением.

Кононюк А.Е. Графы

При интерпретации иерархических сетей простым переходам сопоставляются простые, неделимые операторы, составным переходам — составные операторы. Места интерпретируются как переменные события (управляющие переменные), а условия срабатывания переходов — как условия готовности операторов. Исполнение иерархически организованной асинхронной программы с событийным управлением можно представить как работу рекурсивной процедуры исполнения составного оператора. При включении оператора генерируется локальная копия этой процедуры, работа которой сводится к проверке условий готовности пассивных внутренних операторов. Каждая такая проверка инициируется некоторым событием, например, завершением оператора. Завершение оператора происходит по тем же правилам, что и завершение составного перехода в иерархической сети с ожиданием.

2.12.2. Потокосые сети

Если в событийном асинхронном управлении условия готовности привязаны к управляющим событиям, то в потокосом управлении они изменяются исключительно под влиянием потоков данных, которыми обмениваются операторы (или операции). Исполнение оператора может начаться, если для этого оператора готовы входные данные (аргументы). Для того чтобы обнаружить наличие или отсутствие данных для оператора, необходима память "более активная", чем обычная адресуемая память прямого доступа, в которой любая ячейка (переменная) общедоступна (не резервирована для определенных операторов) и всегда содержит некоторую "обезличенную" информацию без указания истории ее получения и каким операторам она предназначена. Поэтому в потокосых асинхронных программах используется понятие распределенной памяти, состоящей из элементов специального типа — *очередей (каналов)*. Каждая очередь приписана некоторой группе операторов, и указывается, какие операторы могут записывать в очередь и какие считывать из нее. Обычно ограничиваются одной парой операторов — записывающим и считывающим. Функционально очередь представляет собой элемент памяти ограниченной или неограниченной емкости, хранящий в каждый момент времени строго упорядоченную последовательность данных (возможно, пустую). В последовательности выделяется первый элемент (*голова очереди*) и последний элемент (*хвост очереди*). Любая запись данного в очередь увеличивает ее длину на 1, и вновь записанный элемент становится последним (запись в хвост очереди).

Кононюк А.Е. Графы

Любое считывание данного уменьшает длину очереди на 1, первый элемент удаляется из очереди, а второй становится головой очереди. Считывание возможно, если очередь не пуста. Запись возможна, если очередь имеет неограниченную емкость или число данных в очереди меньше ее емкости. Предполагается, что информация о состоянии очереди (пуста, не пуста, переполнена) всегда доступна управлению. Введение распределенной памяти, состоящей из очередей, позволяет конкретизировать условия готовности операторов в потоковом управлении: оператор можно исполнять, если каждая его входная очередь (или определенная часть входных очередей) не пуста. Иногда требуется, чтобы входные очереди содержали не менее некоторого, указанного для данного оператора, числа данных. Иногда добавляется дополнительное условие, чтобы выходные очереди были пусты или не переполнены.

В литературе рассматривались различные варианты программного представления потоковых асинхронных вычислений и архитектуры потоковых вычислительных машин.

Выше было показано, что механизм событийного асинхронного управления описывается в терминах сетей Петри, а для того, чтобы запрограммировать его, можно использовать управляющие выражения, основанные на алгебре регулярных и иерархических сетей. Семантику потокового управления также можно описывать в терминах сетей Петри, более того, можно использовать модификацию алгебры регулярных сетей и с ее помощью программировать асинхронные потоковые программы. Назовем *потоковыми сетями* интерпретированные регулярные сети, в которых переходы интерпретируются как операции (функции), места — как очереди, а фишки — как данные. Дополнительно принимаются следующие соглашения: переход с n входными местами (места упорядочены) может интерпретироваться только n -местной операцией; имеется неявный, скрытый механизм организации очереди данных-фишек в каждом месте.

На рис. 5 показаны три потоковые сети, задаваемые следующими формулами и интерпретирующими функциями, которые сопоставляют каждому переходу некоторую операцию:

Конюк А.Е. Графы

- а) $1((t_1, t_2); t_3, (t_3, t_4); t_5); t_6$.
 $f_1(t_1) = a, f_1(t_2) = b, f_1(t_3) = +,$
 $f_1(t_4) = c, f_1(t_5) = \times, f_1(t_6) = \div$.
- б) $1((t_1, t_2); t_3, (t_4, t_5); t_6); t_7$.
 $f_2(t_2) = a, f_2(t_2) = b, f_2(t_3) = +, f_2(t_4) = c,$
 $f_2(t_5) = d, f_2(t_6) = +, f_2(t_7) = +$.
- в) $1((t_1, t_2); t_3, t_4); t_5, t_6); t_7$.
 $f_3(t_1) = a, f_3(t_2) = b, f_3(t_3) = +, f_3(t_4) = c,$
 $f_3(t_5) = +, f_3(t_6) = d, f_3(t_7) = +$.

Здесь символы $+$, \times , \div обозначают бинарные операции сложения, умножения и деления нацело. Символы a, b, c, d обозначают одноместные константные операции, которые для любого значения аргумента вырабатывают одно и то же значение, равное соответственно a, b, c или d (т.е. $\forall k, a(k) = a$ и т.д.).

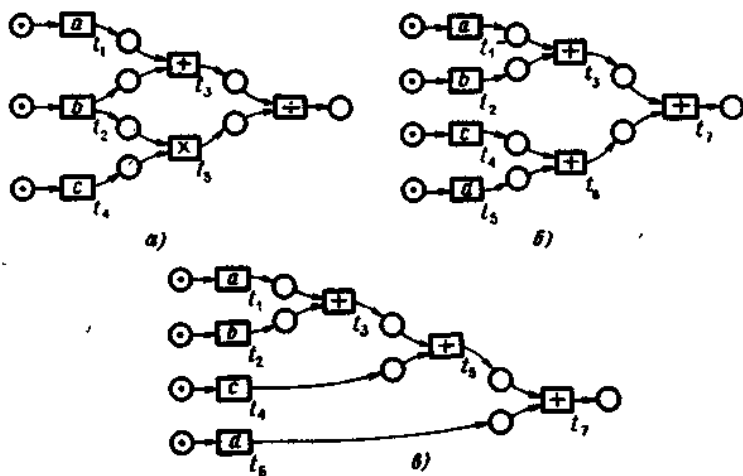


Рис. 5.

В потоковых сетях операция, соответствующая переходу, может исполниться, если данные-фишки присутствуют во всех входных местах перехода. После исполнения операции результат вырабатывается в таком числе экземпляров, каково число выходных мест-очереди у соответствующего перехода.

Кононюк А.Е. Графы

Если подставить вместо символов переходов в формуле потоковой сети символы интерпретирующих их операций или функций, то получится потоковое, управляющее выражение, составленное из интерпретирующих операций и операций управления. Одна и та же интерпретирующая операция может подставляться вместо разных переходов. Условимся отмечать дополнительными индексами вхождения одной и той же операции, сопоставленной разным переходам. Это позволит избежать слияния разных переходов в один при Построении сети по потоковой формуле. Потокковые выражения, задающие сети на рис. 5, имеют следующий вид:

- (1) $1 \{ (a, b); +, (b, c); \times \}; \div,$
- (2) $1 \{ (a, b); +_1, (c, d); +_2 \}; +_3,$
- (3) $1 \{ ((a, b); +_1, c); +_2, d \}; +_3.$

Если удалить из этих потоковых выражений операции управления, скобки и дополнительные индексы, то они примут вид:

- (1) $ab + bc \times \div,$
- (2) $ab + cd + +,$
- (3) $ab + c + d +.$

Полученные выражения представляют собой обратную польскую запись выражений, которые в инфиксной скобочной нотации имеют вид:

- (1) $(a + b) \div (b \times c),$
- (2) $(a + b) + (c + d),$
- (3) $((a + b) + c) + d.$

Формулы, задающие потоковые сети на рис. 5, и соответствующие потоковые выражения не содержат операций исключения и итерации. Такие выражения и потоковые сети будем называть простыми. Процесс преобразования формул потоковых сетей в потоковые выражения, затем в обратную польскую запись и, наконец, в обычную инфиксную запись можно однозначно провести в обратном направлении. Таким образом, для однозначного указания структуры потокового управления в арифметических выражениях достаточно записать их в польской или в скобочной нотации. Поэтому для программирования простых выражений в асинхронных потоковых программах не требуется специальных управляющих примитивов. Следовательно, и асинхронный потоковый язык высокого уровня может не содержать специальных программных механизмов распараллеливания выражений.

Оставаясь в рамках простых потоковых выражений, можно обеспечить функциональную полноту потоковых программ, если использовать

Кононюк А.Е. Графы

наряду с операциями-преобразователями "условные операции", и обеспечить конвейерное исполнение потоковых выражений- за счет подходящей разметки мест-очереди.

Например, для того чтобы запрограммировать условное выражение **if a < b then (a + c) else (a - c)**

можно использовать условную операцию $B(x,y,z)$, в которой первый аргумент имеет тип "логический":

$$B(x, y, z) = \begin{cases} y, & \text{если } x = \text{истина,} \\ z, & \text{если } x = \text{ложь.} \end{cases}$$

Потоковое выражение для рассматриваемого условного выражения имеет вид:

$$1 \{ \{ (a, b); <, (a, c); +, (a, c); - \}; B \},$$

где a, b, c — константные операции, бинарная операция. $<$ вырабатывает логическое значение.

Простая потоковая сеть, задаваемая этой формулой, показана на рис. 6, а польская запись формулы выглядит следующим образом:

$$ab < ac + ac - B.$$

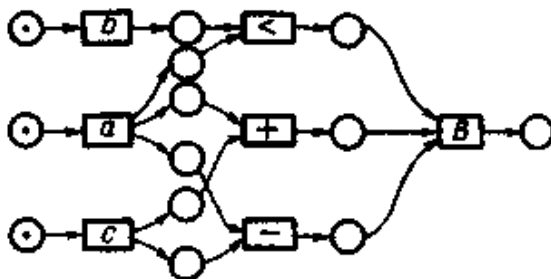


Рис. 6.

Недостатком введенного представления условного выражения в потоковой сети является необходимость вычисления обеих его альтернатив. Другой способ обеспечения функциональной полноты потоковых программ - использование регулярных сетей общего вида, т.е. использование управляющих операций исключения \parallel и итерации $*$. В этом случае к множеству операций и функций, интерпретирующих переходы сети, добавляются специальный класс унарных операций, которые мы назовем *фильтрами*. Условия срабатывания интерпретированного фильтром перехода меняется: он может сработать, если его входное место содержит данное и одновременно

значение фильтра для этого данного истинно. Если переход-фильтр может сработать, то он, как обычно, забирает фишку-данные и посылает его без изменений во все свои выходные места.

Рассмотренное выше условное выражение можно запрограммировать с применением фильтров $<$ и \geq следующим образом:

$$1((a, b); ((<; (a, c); +) \parallel (\geq; (a, c); - 1))).$$

Соответствующая потоковая сеть показана на рис. 7.

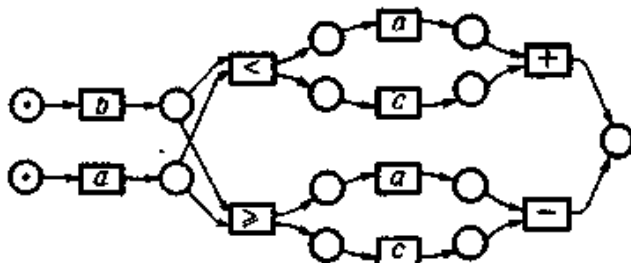


Рис. 7

На этом примере видно, что фильтры следует использовать в сочетании с операцией исключения.

2.12.3. Типы управления

Для того чтобы в параллельных программах иметь возможность в удобной форме задавать сложные структуры управления, большое распространение получила идея выделения управляющих примитивов в отдельный подязык управления, который встроен в язык параллельного программирования. Этот подязык управления должен позволять описывать сложные управляющие взаимодействия и в то же время он должен основываться на относительно простом, формальном семантическом базисе, позволяющем достаточно строго и надежно задавать корректные структуры управления и/или верифицировать их автоматически. Так, в ряде работ предложена формальная методика описания структур управления с помощью управляющих выражений, семантика которых представляется в терминах сетей Петри. Алгебра регулярных и иерархических сетей, рассмотренная ранее, предназначена, в частности, для аналогичных целей. На ее основе предложен подязык управления, являющийся частью базового языка параллельного программирования.

Основной конструкцией подязыка управления служит формула управления, составленная из операторов (модулей) языка как аргументов и

Кононюк А.Е. Графы

из операций управления. Оператору соответствует атомарная сеть, состоящая из единственного перехода (простого в случае простого оператора и составного в случае составного оператора или модуля). Операции управления — это операции над сетями, введенные ранее. Таким образом, семантика структуры управления, заданной формулой управления, представима регулярной или иерархической сетью.

Формула управления задает некоторую конкретную структуру управления. Однако, наряду с операциями в алгебре сетей (и структур управления), можно рассматривать и функции над множествами сетей (и структур управления). Такие функции будем называть *функциями управления* или *типами управления*. В контексте структур управления последний термин используется приблизительно в том же смысле, в котором используется понятие "тип данных" так как функция управления задает класс структур управления, обладающих некоторым общим свойством, например, семейство последовательных структур, конвейерных структур и т.п.

Тип управления может быть описан программистом, ему может быть присвоено имя (идентификатор), которое далее может использоваться в формулах управления. В данном параграфе мы обсудим принципиальную сторону использования типов управления и приведем примеры их описания и применения для конструирования параллельных программ.

Функция (тип) управления может быть описана обычным образом по следующей схеме:

функция-управления: := имя-функции-управления
(список-формальных-параметров) = формула-управления.

Формальный параметр — это параметр-идентификатор, который при использовании (вызове) функции управления заменяется фактическим оператором (именем оператора) или структурой из операторов (сетью). Если в описании функции управления используются только скалярные параметры, то эта функция задает *жесткий тип* управления, описывающий некоторую фиксированную абстрактную структуру управления, конкретизация которой может осуществляться только путем непосредственной замены формальных параметров конкретными операторами (их именами) и сетями из операторов.

Например, можно описать взаимное исключение исполнения двух операторов, введя тип управления `mutex 2` с помощью следующего описания:

`mutex 2 (a, b) = 1 * (a | b)`

где a и b — формальные параметры.

Кононюк А.Е. Графы

Для описания параллельного исполнения двух пар последовательных операторов можно ввести тип `parseq 4` с помощью следующего описания:

`parseq 4 (a, b, c, d) = 1((a; b), (c; d)).`

Семантика описанных типов управления изображается сетями на рис. 8, а и б.

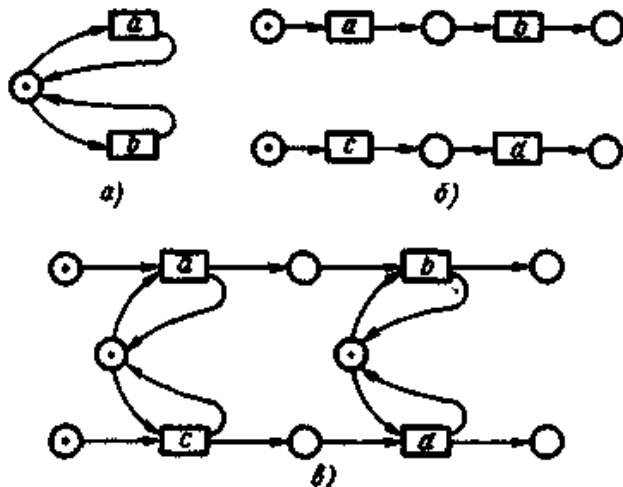


Рис. 8.

Наконец, можно описать еще один тип управления, используя ранее описанные типы управления:

`parcx 4 (a, b, c, d) = {parseq 4 (a, b, c, d), mutex 2 (a, c), mutex 2 (b, d)}.`

Сеть, иллюстрирующая последний тип управления, показана на рис.8,в. Заметим, что формула управления в левой части последнего описания — это не просто формула алгебры сетей, описанной ранее, а формула, в которой использованы функции управления точно таким же образом, как используются функции в арифметических выражениях. Символы a, b, c, d в левой части описания типа `parcx 4` являются фактическими параметрами по отношению к типам (функциям) встречающимся в правой части. В качестве фактических параметров могут, таким образом, выступать формальные или конкретные операторы и структуры управления, в том числе задаваемые функциями управления. Везде далее мы будем применять термин

Кононюк А.Е. Графы

"формула управления" в расширенном смысле, допуская использование как операций, так и функций управления.

Введенные выше типы управления являются жесткими типами с фиксированным числом операторов или модулей. В общем случае желательно иметь типы управления, описывающие семейство подобных структур с "похожей" топологией управляющих связей, но с разным числом операторов. Если ограничиться только жесткими типами, то пришлось бы описывать большой набор функций управления: например, взаимное исключение для 2, 3, 4 и т.д. операторов. Чтобы избежать такой ситуации, наряду со скалярными объектами алгебры управления (операторами, сетями из операторов) необходимо рассматривать составные объекты — векторы и матрицы операторов и сетей из операторов. Другими словами, следует обобщить алгебру сетей на случай векторов и матриц сетей.

Операции алгебры регулярных и иерархических сетей, введенные ранее для "скалярных" сетей, распространяются на векторы и матрицы сетей следующим образом.

Унарные операции (итерация и разметка) "просачиваются" при применении их к векторам и матрицам из операторов и сетей до скалярных элементов. Например, формула ,

$$\bullet ((a, b), (c, d)) = ((\bullet a, \bullet b), (\bullet c, \bullet d))$$

задает матрицу 2×2 из сетей вида $\bullet x$, где $x = a$ или b или c или d .

Бинарные операции обобщаются на случай векторов сетей за счет ассоциативности этих операций. При таком обобщении удобней использовать префиксную форму операций над сетями, поэтому изменим обозначения бинарных операций на более наглядные обозначения следующим образом:

символ " Δ " наложения заменим на символ Δ ,

символ " \triangleleft " присоединения заменим на символ \triangleleft ,

символ " ∇ " исключения заменим на символ ∇ .

Тогда

$$\Delta(a, b, c, d) = (a, b, c, d),$$

$$\triangleleft(a, b, c, d) = (a; b; c; d),$$

$$\nabla(a, b, c, d) = (a \mid b \mid c \mid d).$$

Бинарные операции распространяются на случай матриц по следующему правилу просачивания: матрица, к которой применена бинарная операция, транспонируется, после чего операция применяется к каждой из векторных строк. Например,

$$\nabla(a, b), (c, d) = (\nabla(a, c), \nabla(b, d)).$$

Кононюк А.Е. Графы

В данном случае результатом применения операции исключения к матрице 2×2 из элементарных сетей является вектор из двух сетей, задаваемых формулами $\nabla(a, c)$ и $\nabla(b, d)$ (или $(a \parallel e)$ и $(b \parallel d)$).

Для распространения функций управления на случай, когда формальный параметр может заменяться составным значением, также используются описанные выше правила просачивания. При этом возможность замены формального параметра составным значением дает возможность задавать "динамические" типы управления:

$$\mathbf{mutex}(\mathbf{vec} \ x) = 1 \cdot \nabla x,$$

где $\mathbf{vec} \ x$ означает, что функция непосредственно применима к вектору операторов.

Теперь можно использовать тип $\mathbf{mute} \ x$ в формуле управления

$$\mathbf{mutex}(\{a, b\}),$$

в которой вектор (a, b) - фактический параметр (пара операторов). Эта формула эквивалентна формуле управления

$$1 \cdot \nabla(a, b),$$

т.е. формуле, стоящей в правой части описания типа $\mathbf{mutex} \ 2$.

Аналогично формула $\mathbf{mutex}((a, b, c))$ эквивалентна формуле $1 \cdot \nabla(a, b, c)$.

Если вместо формального параметра-вектора подставляется фактический параметр-матрица $Y = (Z_1, \dots, Z_n)$, где Z_i - вектор, то функция \mathbf{mutex} применима к нему, но только после предварительного унарного просачивания функции по следующему правилу:

$$\mathbf{mutex}(Y) = \mathbf{mutex}(\{Z_1, \dots, Z_n\}) = \{ \mathbf{mutex}(Z_1), \dots, \mathbf{mutex}(Z_n) \}.$$

Пусть динамический тип управления описан следующим образом:

$$\mathbf{parex}(\mathbf{vec} \ x, \mathbf{vec} \ y) = 1 \Delta(\nabla x, \nabla y, \Delta * \nabla(x, y)).$$

Формула $\mathbf{parex}((a, b, c), (d, e, f))$ задает следующую структуру:

$$\begin{aligned} \mathbf{parex}(\{a, b, c\}, \{d, e, f\}) &= \\ &= 1 \Delta(\nabla(a, b, c), \nabla(d, e, f), \Delta * \nabla(\{a, b, c\}, \{d, e, f\})) = \\ &= 1 \Delta(\nabla(a, b, c), \nabla(d, e, f), \Delta * (\nabla(a, d), \nabla(b, e), \nabla(c, f))) = \\ &= 1 \Delta(\nabla(a, b, c), \nabla(d, e, f), \Delta (* \nabla(a, d), * \nabla(b, e), * \nabla(c, f))) = \\ &= 1 (\{a; b; c\}, \{d; e; f\}, * (a \parallel d), * (b \parallel e), * (c, f)). \end{aligned}$$

Соответствующая сеть показана на рис. 9.

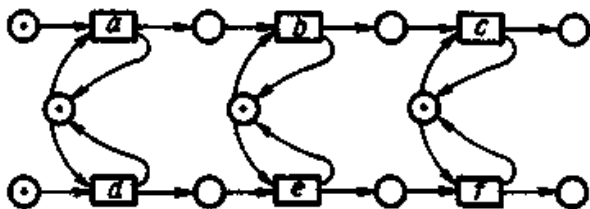


Рис. 9.

Для функции управления $seqex (matrx) = \triangleleft \nabla x$, где $matrx$ указывает, что эта функция непосредственно применима к матрице операторов, структура управления, задаваемая формулой

$$seqex ((a, b, c), (d, e, f)) = \triangleleft \nabla ((a, b, c), (d, e, f)) = \triangleleft (\nabla(a, d), \nabla(b, e), \nabla(c, f)) = \triangleleft (a \parallel d); (b \parallel e); (c \parallel f)),$$

показана на рис. 10, а.

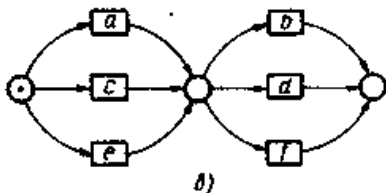
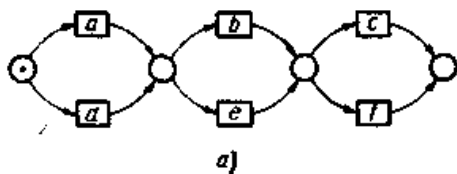


Рис. 10.

Структура управления, задаваемая другой формулой, которая использует тот же тип управления $seqex$, показана на рис. 10, б:

$$seqex ((a, b), (c, d), (e, f)) = \triangleleft \nabla ((a, b), (c, d), (e, f)) = \triangleleft (\nabla(a, c, e), \nabla(b, d, f)) = \triangleleft (a \parallel c \parallel e); (b \parallel d \parallel f).$$

Двумерная сеточная структура с переменным числом операторов по обоим измерениям может быть определена с помощью следующего типа управления:

Конюк А.Е. Графы

$\text{lattice}(\text{matrix}) = 1 \Delta (\Delta \triangleleft x, \Delta \triangleleft \Gamma x)$.

Для случая матрицы 3×3 структура управления показана на рис. 11:

$$\begin{aligned} \text{lattice}((a_{11}, a_{12}, a_{13}), (a_{21}, a_{22}, a_{23}), (a_{31}, a_{32}, a_{33})) &= \\ &= 1 \Delta (\Delta \triangleleft ((a_{11}, a_{12}, a_{13}), (a_{21}, a_{22}, a_{23}), (a_{31}, a_{32}, a_{33})), \\ \Delta \triangleleft ((a_{11}, a_{21}, a_{31}), (a_{12}, a_{22}, a_{32}), (a_{13}, a_{23}, a_{33}))) &= \\ &= 1 \Delta (\Delta (\Delta \triangleleft (a_{11}, a_{21}, a_{31}), \triangleleft (a_{12}, a_{22}, a_{32}), \triangleleft (a_{13}, a_{23}, a_{33}))), \\ \Delta (\triangleleft (a_{11}, a_{12}, a_{13}), \triangleleft (a_{21}, a_{22}, a_{23}), \triangleleft (a_{31}, a_{32}, a_{33}))) &= \\ &= 1 ((a_{11}; a_{21}; a_{31}), (a_{12}; a_{22}; a_{32}), (a_{13}; a_{23}; a_{33})), \\ (a_{11}; a_{12}; a_{13}), (a_{21}; a_{22}; a_{23}), (a_{31}; a_{32}; a_{33})) & \end{aligned}$$

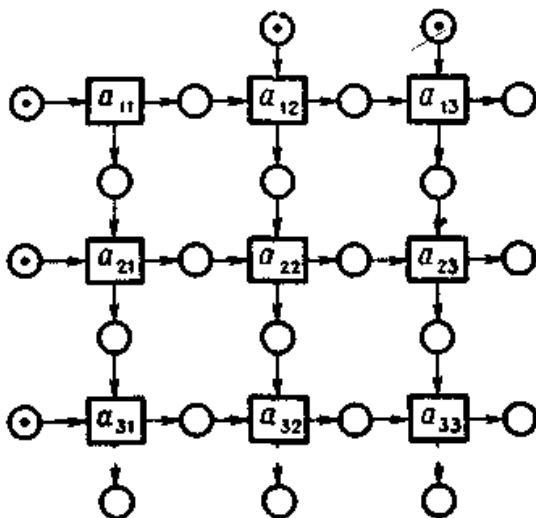


Рис. 11.

Программируемые типы управления дают возможность конструировать разнообразные структуры управления. В частности, техникой типов управления удобно описываются многие известные управляющие конструкции последовательных и параллельных языков. После того как программист описал набор "абстрактных" типов управления и проверил их правильность, он может с их помощью конструировать конкретные структуры управления в программах.

Рассмотрим в качестве примера, иллюстрирующего возможности программирования с типами управления, каким образом можно

Кононюк А.Е. Графы

специфицировать структуру управления параллельной программы, решающей известную задачу о пяти философях.

Пять философов гуляют в саду и размышляют. Когда философ чувствует, что проголодался, он заходит в столовую, где стоит круглый стол с пятью стульями и миской спагетти посреди стола. На столе — пять вилок, по одной слева и справа от каждого стула. Философ садится за стол, берет вилки (если они на месте) и ест спагетти (для этого обязательно нужно две вилки). Утолив голод, философ кладет вилки на стол и выходит в сад размышлять, пока вновь не почувствует, что проголодался. Задача состоит в том, чтобы синхронизировать независимые действия философов и не допускать взаимной блокировки, когда все философы сидят за столом, каждый взял по одной вилке и никто из них не может начать есть спагетти. Решение задачи — не позволять всем пяти философам одновременно находиться в столовой.

Пусть набор действий всех пяти философов представлен матрицей PH размерности 5×7 , в которой i -я строка $PH[i,]$ соответствует действиям i -го философа. Каждый из философов выполняет семь действий, образующих следующий вектор операторов:

- $PH[i, 1]$ (философ i входит в столовую),
- $PH[i, 2]$ (философ i берет левую вилку),
- $PH[i, 3]$ (философ i берет правую вилку),
- $PH[i, 4]$ (философ i ест спагетти),
- $PH[i, 5]$ (философ i кладет левую вилку),
- $PH[i, 6]$ (философ i кладет правую вилку),
- $PH[i, 7]$ (философ i выходит из столовой).

Для программирования задачи о пяти философях сконструируем следующие типы управления:

$$\text{act}(\text{vec } X) = 1 * \Delta \{X[1], \Delta \{X[2], X[3]\}, X[4], \Delta \{X[5], X[6], X[7]\}.$$

При замене формального параметра-вектора X строкой матрицы PH формируется сеть, описывающая поведение одного философа. При замене параметра X матрицей PH функция act просачивается, так что результатом выражения $\text{act}(PH)$ является вектор из пяти сетей, описывающих поведение философов. Наконец, формула управления

$\Delta \text{act}(PH)$ задает параллельную структуру управления (сеть), показанную на рис. 12.

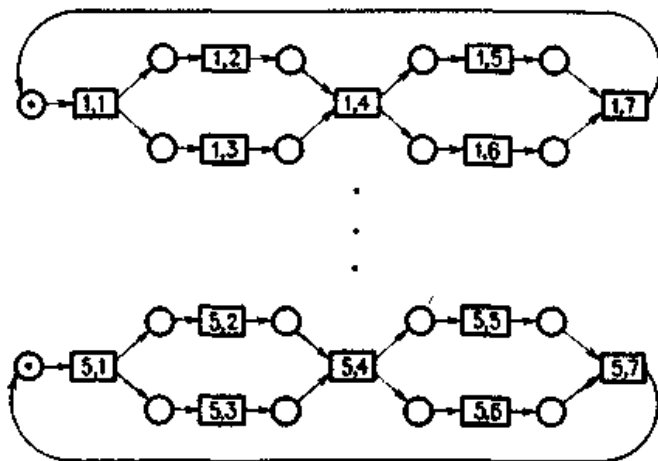


Рис. 12.

Эта структура описывает независимые параллельные действия философов.

Для вектора длины 7 следующая функция `lfork` выделяет его "критический участок". Если вместо X подставить вектор $PH[i]$, то этот участок включает последовательность действий философа i , при которых он использует левую от него вилку:

`lfork (vec X) = <1(X[2], X[4], X[5]).`

Аналогично функция `rfork` будет использована для выделения критического участка, связанного с использованием философом правой вилки:

`rfork (vec X) = <1(X[3], X[4], X[6]).`

Вспомогательная функция `shift` осуществляет циклический сдвиг вектора длины 5:

`shift (vec X) = Y, где $Y[i] = X[(i + 1) \bmod 5], i = 1, \dots, 5.$`

Тип управления

`parmutex (vec X, vec Y) = $\Delta(1 \cdot \nabla(X, Y))$`

задает покомпонентное взаимное исключение элементов векторов X и Y .

Тогда формула управления

`Δ parmutex(lfork(PH), shift(rfork(PH)))`

задает структуру управления, показанную на рис. 13.

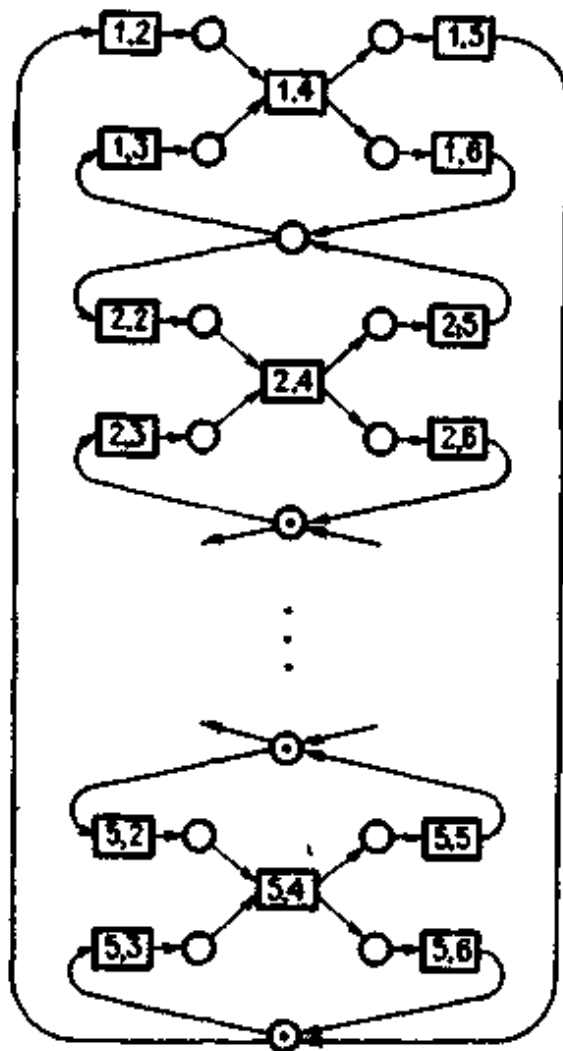


Рис. 13

Эта структура указывает, что каждая вилка является общим ресурсом для сидящих рядом философов, и исключает возможность

Кононюк А.Е. Графы

одновременного исполнения ими действий, для которых требуется вилка.

Наконец, тип управления

$$\mathbf{four}(\mathbf{vec} X) = (4 * X)$$

разрешает параллельное исполнение не более четырех операторов или сетей из вектора, подставляемого вместо X . Поэтому формула управления

$$\Delta \mathbf{four}(\mathbf{act} \{PH\})$$

указывает, что не более четырех философов может одновременно находиться в столовой.

Теперь остается методом последовательного наложения отдельных слоев записать результирующую формулу управления, задающую структуру управления параллельной программы, которая решает задачу о пяти философях:

$$\{\Delta \mathbf{act} \{PH\}, \Delta \mathbf{parmutex} \{\mathbf{rfork} \{PH\}\},$$

$$\mathbf{shift} \{\mathbf{rfork} \{PH\}\}, \Delta \mathbf{four}(\mathbf{act} \{PH\})\}.$$

Соответствующая сеть показана на рис. 14.

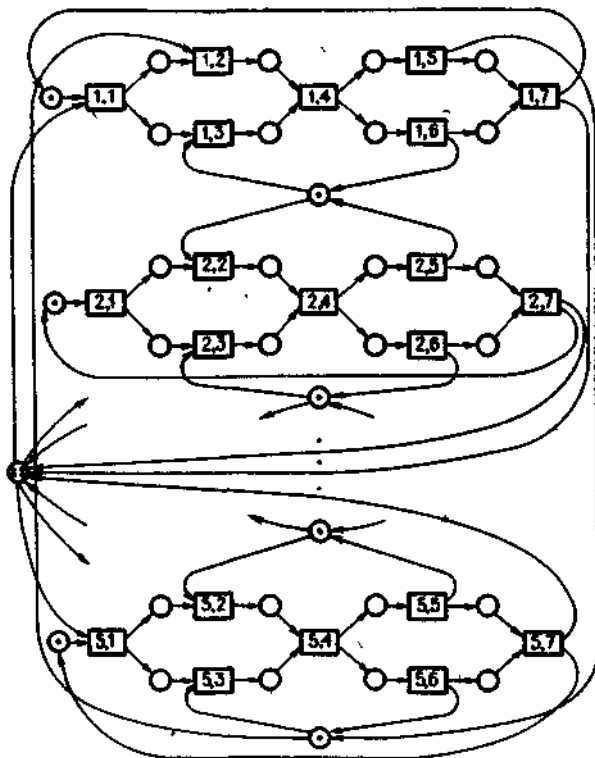


Рис. 14.

2.12.4 Петри-машина

Регулярные и иерархические сети, представленные формулами, можно "обрабатывать" с помощью ЭВМ. Такая обработка может включать преобразования сетей, их оптимизацию по заданным критериям, моделирование их функционирования. Конечной целью программного представления сетей является применение их как компонентов систем программирования, ориентированных на описание и моделирование параллельных структур. Примером программного имитатора иерархических сетей с ожиданием может служить так называемая Петри-машина, которая строит по заданной формуле иерархическую сеть с ожиданием и имитирует ее функционирование. Петри-машина является основой интерпретатора управления для асинхронных параллельных программ.

Кононюк А.Е. Графы

Петри-машина состоит из четырех блоков:

— блок операций реализует операции над сетями,

— блок развертки строит внутреннее представление сети в форме так называемой развернутой сети,

— блок ввода осуществляет ввод формул и построение развернутой сети с помощью первых двух блоков,

— блок имитации моделирует функционирование сети.

Блок ввода представляет собой однопроходный транслятор с языка формул во внутреннее представление развернутых сетей. Синтаксическая структура этого языка аналогична структуре простых арифметических выражений. Простой синтаксис входного языка позволил найти простое программное решение блока. Грамматика языка расширена до автоматной, реализован стековый метод трансляции.

Блок операций представляет собой набор процедур и функций, реализующих операции над сетями и некоторые вспомогательные действия. Для внутреннего представления сетей, которые могут иметь произвольный объем и структуру, использовалась имеющаяся в языке Паскаль возможность динамического построения конечных графов с помощью списочных структур и процедур динамического размещения данных.

Иерархическая сеть разбивается на уровни, имеющие одинаковую структуру. К высшему уровню относятся переходы, которые не являются внутренними для других переходов сети. Их вложенность считается равной единице, так как вся сеть рассматривается как составной переход. Для каждого перехода его вложенность на единицу больше вложенности охватывающего перехода.

Петри-машину можно представить как универсальную процедуру генерации независимых копий процедуры управления функционированием составного перехода. Каждая такая копия генерируется для данного составного перехода, когда он становится активным. Она организует локальное управление работой этого перехода в течение всего времени его активности и исчезает после его завершения. В иерархической сети два и более составных перехода могут одновременно находиться в активном состоянии, и, следовательно, несколько копий управления работают параллельно. Если некоторые составные переходы сети имеют общее стороннее место, то при ее функционировании может возникнуть конфликтная ситуация, когда локальные управления переходов обращаются к этому месту. Локальное управление любого из этих составных переходов не может самостоятельно разрешить конфликт, оно может решать только вопросы, связанные с функционированием внутренней сети своего составного модуля.

Кононюк А.Е. Графы

Для разрешения конфликтов, связанных с общими сторонними местами, нужна специальная процедура, называемая *арбитром*. Ее копии порождаются при активации составных модулей вместе с копиями основного управления. При работе арбитра используется сеть, построенная из заданной иерархической сети специальным образом (развернутая сеть). В ней наряду с переходами заданной сети имеются так называемые *теневые переходы*, которые являются представителями конфликтующих переходов на тех уровнях сети, на которых разрешаются конфликты между данными переходами.

Рассмотрим произвольную иерархическую сеть N с множеством переходов TT и множеством мест P и введем следующие обозначения:

$$J(t) = \{t' \in TT \mid t' \text{ — внутренний переход } t\},$$

$$Q(t) = \{t' \in TT \mid t' \text{ — объемлющий переход для } t\},$$

$$T(p) = p \cup p', \text{ т.е. множество переходов, инцидентных месту } p.$$

В P выделяется множество конфликтных мест сети $K_N = \bigcup_{t \in TT} K(t)$, где

$$K(t) = \{p \in P \mid T(p) \cap J(t) \neq \emptyset \wedge T(p) \setminus J(t) \neq \emptyset\}.$$

Таким образом, конфликтные места для составного перехода t инцидентны как внутренним переходам перехода t , так и внешним по отношению к нему переходам.

Пусть P_t — множество внутренних мест перехода t (напомним, что место p является внутренним для t , если $T(p) \subseteq J(t)$), а $P(t) = t \cup t'$.

Будем говорить, что t - переход типа 0, если $P(t) \cap K_N = \emptyset$, и типа 1 в противном случае.

Для конфликтного места $p \in K_N$ определим охватывающий переход $g(p)$ такой, что

$$(a) p \in P_{g(p)},$$

$$(б) \forall t \neq g(p) \text{ из } p \in P_t \text{ следует, что } g(p) \in J(t).$$

Границей конфликтов для перехода t назовем переход $g(t) \in Q(t)$ такой, что

$$(a) P(t) \subseteq P_{g(t)}, \text{ и}$$

$$(б) \forall t' \neq g(t) \text{ из } P(t) \in P_{t'} \text{ следует, что } g(t) \in J(t').$$

Другими словами, граница конфликтов для перехода t — это "минимальный" объемлющий у переход, содержащий все входные и выходные места t в качестве внутренних. Граница конфликтов

Кононюк А.Е. Графы

существует для любого перехода, так как мы условились, что заданная иерархическая сеть представляет собой составной переход.

Для каждого перехода t типа 1 сети N в каждом переходе $t' \in Q(t) \cap J(g(t))$ вводим дополнительный теневого переход типа 1, соответствующий t . В переходе $g(t)$ вводим в качестве внутреннего теневого переход $s(t)$ типа 0. Значения функций инцидентности F меняем так, что для любого g типа 1 и для любого $p \in K_N \cap P(t)$

$$F'(p, s(t)) = F(p, t), F'(s(t), p) = F(t, p), F'(p, t) = 0, F'(t, p) = 0.$$

Для остальных теневых переходов значения функций F' равны 0. Между переходами типа 1 и соответствующими теневыми переходами устанавливается двусторонняя связь для передачи информации о наличии фишек в конфликтных местах. Переход t связывается со своим теневым переходом в охватывающем переходе, тот - с теневым переходом в следующем объемлющем переходе и т.д. до $g(t)$

В результате таких преобразований исходной иерархической сети получается *развернутая сеть*.

На рис. 15, *а* показана иерархическая сеть, а на рис. 15, *б* — соответствующая развернутая сеть.

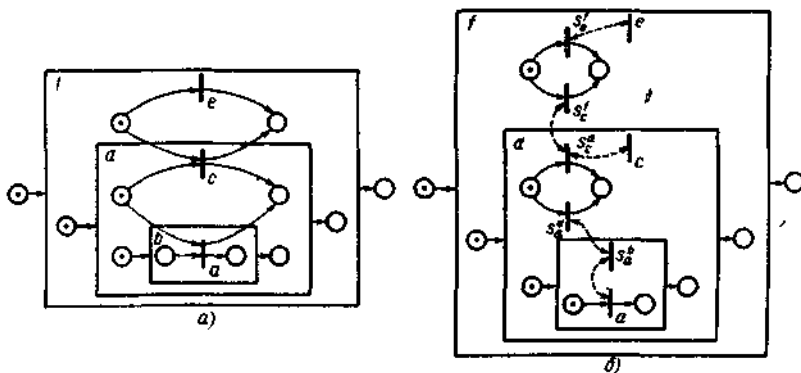


Рис 15

Теневые переходы изображены светлыми барьерами, связи между ними и переходами типа 1 — пунктирными стрелками.

В процессе имитации сети для каждого составного перехода t генерируется копия основного управления C_i и копия арбитра A_i . Управление C_i проверяет локальные условия активации переходов, охватываемых переходом t , и переводит их из состояния в состояние. Арбитр A_i ведает конфликтными местами p такими, что $g(p) = t$.

Кононюк А.Е. Графы

При определении функционирования сетей с ожиданием вводилось два состояния переходов — активный и пассивный, а изменения состояний рассматривались как мгновенные события. При моделировании сетями реальных систем такие события имеют длительность. Аналогично при программном моделировании сетей проверки условий активации, условий завершения и т.п. также имеют длительность. Поэтому в Петри-машине с каждым переходом связано несколько состояний: он может быть пассивным, проверяемым, готовым, активным, ждущим, остановившимся.

Проверка условий активации различна для переходов типа 0 и типа 1. Если для перехода типа 0 выполнено локальное условие активации, то он переводится в состояние "готов". Для перехода t типа 1 основное управление охватывающего перехода проверяет локальные условия активации и, если оно выполнено, проверяемый переход переводится в состояние "ждущий". Соответствующему теневому переходу "наверх" посылается запрос на фишки из мест $P(t) \cap K_N$. Обработку этого запроса и возможную передачу запроса на верхние уровни осуществляет арбитр уровня перехода t , он же передает ответ назад о результате проверки. Если ответ положителен, то переход t переводится в состояние "готов", в противном случае — в состояние "пассивный". Арбитр некоторого уровня формирует запросы от некоторого перехода к верхним уровням только в том случае, если он удовлетворил запросы этого перехода на своем уровне.

Переход, находящийся в состоянии "готов", может быть приведен основным управлением в активное или пассивное состояние. Когда составной переход активирован, генерируется его локальное основное управление и локальный арбитр. Все внутренние переходы считаются пассивными.

Составной переход завершается, если ни один из охватываемых им переходов не является активным или ждущим и локальное условие активации всех этих переходов не выполнено. В процессе завершения происходит смена разметки сети. Основное управление, завершая переход типа 0, передает фишки в его выходные места и переводит переход в пассивное состояние. Если же переход имеет тип 1, то управление дополнительно сообщает соответствующему теневому переходу о завершении перехода. Арбитр, приняв этот сигнал, посылает фишки в выходные места теневого перехода. Если теневой переход имеет тип 0, то на этом передача сообщения "вверх" обрывается, а в противном случае сообщение идет на верхние уровни.

В заключение отметим, что аастоящая работа не охватывает всей проблематики теории сетей Петри и ее приложений к задачам моделирования дискретных систем. За рамками работы остались прежде всего различные обобщения и модификации сетей Петри, вводимые как проблемно-ориентированные средства высокого уровня для моделирования систем специального вида: радиоэлектронных устройств, сетевых протоколов, экономических систем и т.п. Кроме тогр, не освещены систематически результаты применения сетей Петри к задачам обнаружения типовых ситуаций в дискретных системах, таких, как, например, взаимные блокировки. Это связано в первую очередь с тем, что описанные в литературе результаты не образуют критическую массу, достаточную для их монографического обобщения. Что касается общей теории сетей, то она, образует внутренний, "технологический" уровень сетевой проблематики. Поэтому она требует более специального изложения и погружения в контекст современных математических основ обработки информации.

2.13. Модели параллельных вычислений

Сети Петри определены как модели систем с параллельными действиями. И, как мы видели, сети Петри имеют хорошие возможности моделирования, с их помощью можно моделировать большое число систем. Однако сети Петри — не единственная модель параллельных вычислений. Предложено, исследуется и используется множество других моделей. В этом разделе мы представим некоторые из этих моделей и исследуем их взаимосвязь с сетями Петри. Целью этого раздела является определение моделей, которые могут использоваться в системах моделирования, и их сравнительной мощности моделирования.

Основной задачей, проистекающей из намерения соотнести различные модели друг с другом, является задача установления соответствующего метода сравнения моделей параллельных вычислений. Мы хотели бы иметь возможность доказывать, что модель *A* является «менее мощной», чем модель *B*, или что модель *A* «эквивалентна» модели *B*. Понятия эквивалентности и включения имеют в данном случае особую важность.

Некоторые исследования по соотнесению различных моделей уже проведены. Собраны воедино описания нескольких моделей. В частности, в ряде работ дано общее определение управляющей структуры, которое позволяет определять единым образом различные модели. Это привело к работам, в которых сравниваются различные

Кононюк А.Е. Графы

модели для получения иерархии моделей, соотнесенных по их мощности моделирования. Независимый результат получен в ряде других работ, где сравнивается большое число моделей и строится другая иерархия с подобной структурой.

Оба результата получены с помощью использования языков моделей для их сравнения. Класс моделей A определяет класс языков $L(A)$. Два класса моделей, A и B , будут эквивалентны по определению, если $L(A) = L(B)$. Это означает, что для любого определенного экземпляра a класса моделей A с языком $L(a)$ должен существовать экземпляр b класса B с идентичным языком $L(a) = L(b)$. Если языки правильно характеризуют эти модели, то они являются искомым средством для сравнения двух классов моделей.

Однако, как мы видели, не вполне ясно, как определить язык для моделей параллельных вычислений. Исследования в области определения языков сетей Петри привели к по меньшей мере 12 различным определениям языков, большинство из них, очевидно, различны. Различия в этих языках могут привести к различиям в соотношениях эквивалентности и включения между моделями. С другой стороны, если различия между моделями действительно существенны, то они могут быть не чувствительны к (незначительным) изменениям в определениях эквивалентности и включения. Таким образом, подобность результатов весьма важна из-за того, что в этих работах использовались различные определения эквивалентности и включения. Однако нельзя утверждать, что эти результаты бесспорны. Ряд авторов также сравнили большое число моделей параллельных вычислений и пришли к другим выводам. Их сравнение основывается на очень детальном анализе структуры и *пространства состояний* отдельных представителей классов моделей. Этим объясняется значительное отличие результатов одних авторов от результатов других. Установлено, что существуют различия в мнениях исследователей по вопросам о том, какие модели должны сравниваться, как их сравнивать? Сравнение, проводимое в этом разделе, основано как на структурных характеристиках, так и на характеристиках поведения. Мы будем говорить, что класс моделей A является *меньшим* или *равным по мощности моделирования* (включается в) классу B , если для любого данного экземпляра a класса A существует алгоритм построения экземпляра b класса B , для которого верно, что:

1. Каждая структурная компонента модели a представляется (небольшим) различным множеством компонент модели b . Размер модели b (число элементов) отличается в худшем случае на мультипликативную константу от размера модели a , причем константа

Кононюк А.Е. Графы

определяется классами моделей A и B , а не конкретными экземплярами a и b .

2. Любая последовательность действий в a может быть промоделирована последовательностью в b , с длиной последовательности в b , отличающейся не более чем на мультипликативную константу от длины последовательности в a .

3. Модель b заходит в тупик только тогда, когда заходит в тупик модель a . Модель заходит в тупик, если все ее действия становятся невозможными.

Цели введения этих ограничений очевидны. Первое ограничение устанавливает структурное подобие двух моделей; второе ограничение гарантирует, что две модели ведут себя одинаково. Однако мы не требуем абсолютного соответствия между двумя моделями. Мы допускаем представление действия в одной модели (короткой) последовательностью действий в другой модели или представление компоненты (подобной позиции или переходу) набором (небольшим) компонент. Следовательно, действие в одной модели может моделироваться последовательностью из двух действий в другой модели. Последнее ограничение требует, чтобы более мощная модель не совершала ошибок, когда их не совершает менее мощная. Это исключает возможность построения модели, которая недетерминированно выбирает одно из нескольких действий и останавливается, если оказывается, что был сделан неверный выбор.

Две модели *эквивалентны*, если они включают друг друга. Это предполагает, что любой экземпляр какой-либо модели преобразуется в экземпляр другой.

С учетом сказанного выше, рассмотрим, изложенные в литературе, соотношения между следующими моделями параллельных вычислений:

1. Конечные автоматы.
2. Маркированные графы.
3. Графы вычислений.
4. P/V-системы.
5. Системы с сообщениями.
6. Графы UCLA.
7. Системы сложения векторов.
8. Системы замещения векторов.
9. Расширенные сети Петри.

Для каждого класса моделей вначале определим модель и приведем пример. Затем обсудим связь ее с другими моделями параллельных вычислений.

2.13.1. Конечные автоматы

Ранее показано, что *конечные автоматы* легко преобразуются в сети Петри. Конечные автоматы использовались несколькими исследователями в качестве модели параллельных вычислений. Бредт определил модель, основанную на концепциях, вложенных в аппаратуру ЭВМ. Каждый процессор моделируется конечным автоматом с входными и выходными каналами, которые связывают один процессор с другим. Состоянием каждого входного и выходного канала является либо 0, либо 1. Поскольку каждый выходной канал одного процессора является входным каналом другого процессора, и существует конечное число процессоров и конечное число каналов, каждый с конечным числом состояний, то и вся система имеет конечное число состояний.

Гильберт и Чандлер использовали модель с общей памятью, а не каналы связи. Это означает, что их модель в большей степени, чем модель аппаратуры Бредта, направлена на моделирование программных процессов с совместно используемой памятью, но все же является не чем иным, как моделью с конечным числом состояний, и, следовательно, включена в модель сети Петри.

2.13.2. Маркированные графы

Маркированные графы были рассмотрены ранее. Как подкласс сетей Петри маркированные графы, очевидно, обладают более ограниченной мощностью моделирования. Маркированные графы непосредственно не сопоставимы с конечными автоматами, но, по-видимому, являются двойственными к ним. Таким образом, мы получаем изображенное на рис. 1 соотношение между сетями Петри, конечными автоматами и маркированными графами.

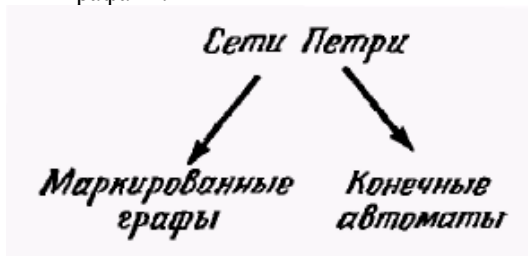


Рис. 1. Соотношение между сетями Петри, маркированными графами и конечными автоматами.

2.13.3. Графы вычислений

Одной из наиболее ранних моделей параллельных вычислений была *модель графов вычислений*. Ее предложили главным образом для представления параллельного выполнения программ, вычисляющих арифметические выражения.

Граф вычислений G определяется как ориентированный граф $G=(V, A)$, где $V = \{v_1, v_2, \dots, v_n\}$ — множество узлов; $A = \{a_1, a_2, \dots, a_m\}$ — множество дуг.

Каждая дуга $a_i \in A$ есть упорядоченная пара узлов (v_j, v_k) , представляющая дугу из v_j к v_k . Каждой дуге $a_i = (v_j, v_k)$ сопоставлена четверка $(I_{j,k}, V_{j,k}, W_{j,k}, T_{j,k})$. Каждая дуга представляет *очередь* элементов данных, полученных процессором в узле v_j и используемых процессором в узле v_k . $I_{j,k}$ есть число элементов данных, находящихся *первоначально* в очереди, соответствующей дуге из v_j к v_k . Узел v_k подготовлен, если на каждой дуге, направленной к узлу v_k из каждого узла v_j , присутствует не менее $T_{j,k}$ элементов данных. $T_{j,k}$ называется *пороговым* значением. При выполнении операции, соответствующей узлу v_k , удаляется $W_{j,k}$ ($W_{j,k} \leq T_{j,k}$) элементов данных из очереди, соответствующей дуге, направленной к v_k . Когда операция, соответствующая v_k , завершается, то она помещает $V_{k,r}$ элементов данных в очереди, соответствующие каждой дуге (v_k, v_r) , направленной из узла v_k к узлу v_r .

На рис. 2 изображен пример графа вычислений.

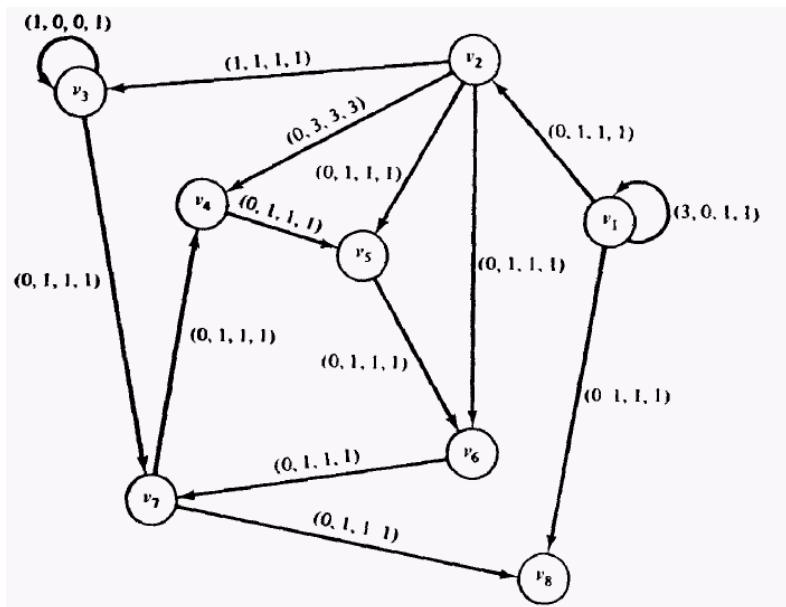


Рис. 2. Граф вычислений.

В начальном состоянии узел v_1 подготовлен, поскольку имеет один вход, и в очереди этого входа присутствуют три элемента данных. При выполнении v_1 он удаляет один элемент данных из этой очереди и по завершении операции помещает один элемент данных на дугу из v_1 к v_2 , один — на дугу из v_1 к v_8 . В этом новом состоянии может выполняться либо v_1 , либо v_2 , так как оба имеют достаточно элементов данных во входных очередях для удовлетворения пороговых условий. Граф вычислений легко моделируется сетью Петри. Каждая дуга представляется позицией, а каждый узел графа вычислений становится переходом. Переход, соответствующий узлу v_k , имеет $T_{j,k}$ входных дуг из позиции, представляющей дугу из v_j к v_k . Это гарантирует то, что переход будет подготовленным только тогда, когда пороговое условие выполнено. Однако, когда переход запущен, он может удалить только $W_{j,k}$ фишек, поэтому $T_{j,k} - W_{j,k}$ дуг направляются обратно от перехода v_k к позиции, представляющей дугу из v_j к v_k . Кроме того, $V_{k,r}$ меток помещаются в позицию, представляющую дугу из v_k к v_r . Начальная маркировка определяется значениями $I_{j,k}$.

На рис. 3 изображена сеть Петри, построенная описанным выше способом для графа вычислений на рис. 2.

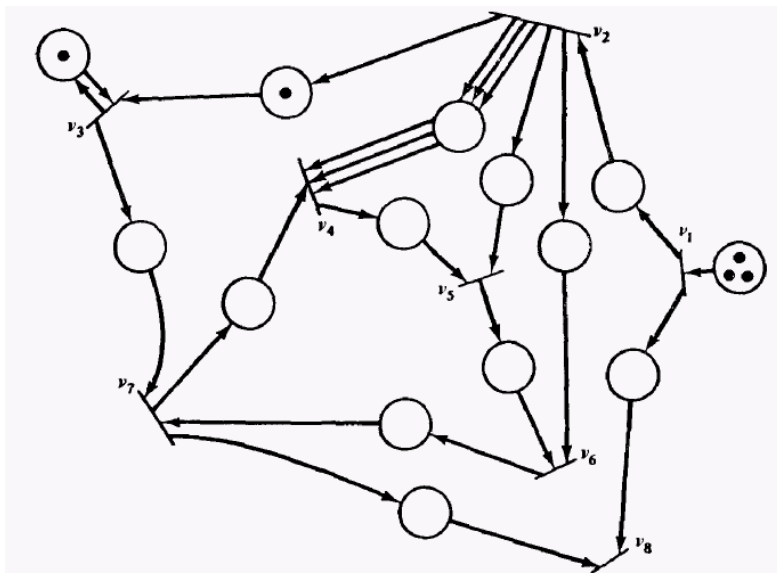


Рис. 3. Сеть Петри, эквивалентная графу вычислений, изображенному на рис. 2.

Легко показывается, что маркированные графы могут быть про- моделированы графами вычислений, для которых $T_{j,k} = W_{j,k}$ для всех узлов v_j и v_k . Однако графы вычислений являются более мощным средством, чем маркированные графы, благодаря возможности моделировать ситуации с $T_{j,k} > W_{j,k}$.

С другой стороны, графы вычислений и конечные автоматы не сопоставимы, как маркированные графы и конечные автоматы. Графы вычислений не могут моделировать принятие решений или условное выполнение — это ограничение справедливо и для маркированных графов. Таким образом, наша иерархия моделей принимает вид, показанный на рис. 4.



Рис. 4. Добавление графов вычислений к иерархии моделей.

Карп и Миллер подробно исследовали графы вычислений, особенно проблемы активности и безопасности. В действительности их интересовало обеспечение и определение условий завершенности графа вычислений (т. е. условия неактивности). Так как дуги (позиции) представляют очереди данных, то исследования ограниченности, проводимые Карпом и Миллером, были направлены на определение максимальной длины очереди. Эти различия в обозначениях и целях, а также отличия в определении моделей между графами вычислений и маркированными графами послужили причиной того, что никто не попытался соотнести результаты и алгоритмы Карпа и Миллера по графам вычислений с работой по маркированным графам.

2.13.4. P/V-системы

P- и V-операции над семафорами впервые введены Дейкстрой для решения проблем синхронизации в системах параллельных процессов. Как таковые они могут использоваться для моделирования синхронизации и связи таким же образом, как и сети Петри. Патил использовал этот подход, когда определял задачу о курильщиках сигарет, чтобы показать ограниченность систем, которые могут использовать только P- и V-операции между процессами. P/V-системы тем не менее популярны, и поэтому имеется много литературы по

Кононюк А.Е. Графы

вычислительной технике, в которой обсуждаются или применяются эти операции.

Ранее показано, что P- и V-операции можно промоделировать с помощью сетей Петри. Доказательство Патила свидетельствует о том, что это включение собственное: существуют задачи (например, задача о курильщиках сигарет), которые можно решить в сетях Петри, но нельзя с использованием только P- и V-операций. Однако P/V-системы являются достаточно мощным средством, чтобы включать модели графов вычислений и модели конечных автоматов.

Чтобы преобразовать конечный автомат в P/V-систему, мы используем для моделирования каждого состояния автомата отдельный процесс. Каждому состоянию сопоставим семафор. Пусть $Q = \{q_1, q_2, \dots, q_n\}$ — множество состояний, а $\delta: Q \times \Sigma \rightarrow Q$ — функция переходов с множеством действий Σ . Сопоставим состоянию q_i семафор S_i и процесс. В начале процесс выполняет $P(S_i)$. В общем случае он будет ждать до тех пор, пока автомат не перейдет в состояние q_i . После выполнения $P(S_i)$ процесс выбирает произвольное $\sigma \in \Sigma$, для которого определена функция $b(q_i, \sigma)$, и выполняет $V(S_j)$, где $q_j = \delta(q_i, \sigma)$. После чего этот процесс возвращается по петле к своему $P(S_i)$. Рис. 6 иллюстрирует преобразование конечного автомата, изображенного на рис. 5.

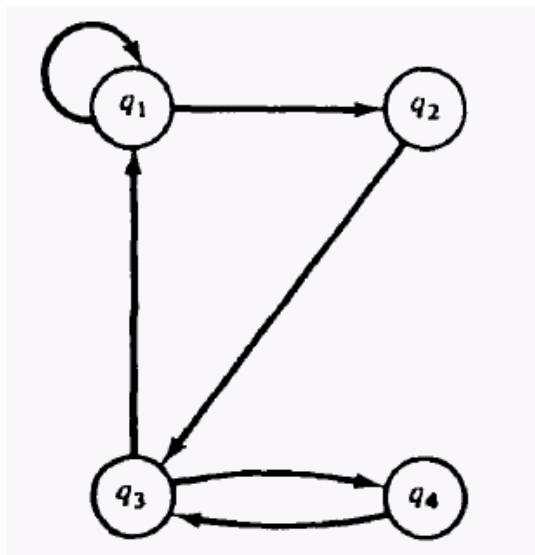


Рис. 5. Конечный автомат.

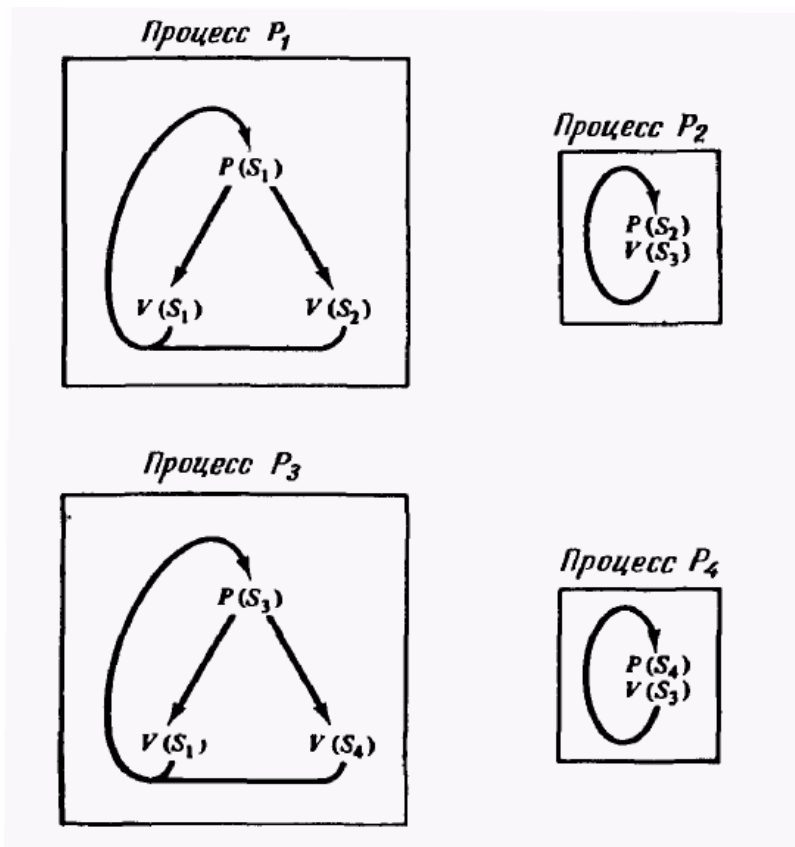


Рис. 6. P/V-система для конечного автомата на рис. 5.

Семафоры первоначально равны нулю, кроме семафора начального состояния, который инициализируется единицей.

Для преобразования графа вычислений в P/V-систему каждой дуге (v_j, v_k) графа ставим в соответствие семафор $S_{j,k}$. Значением семафора будет число элементов данных, ожидающих в очереди для этой дуги. Таким образом, первоначально значением семафора $S_{j,k}$ будет $I_{j,k}$. Для каждого узла в графе вычислений создается процесс. Процесс узла v_k сначала выполняет $T_{i,k}$ P-операций над семафорами $S_{j,k}$ для всех дуг (v_j, v_k) , направленных в v_k . Этим обеспечивается то, что каждая очередь будет содержать не менее $T_{i,k}$ элементов данных. Затем, поскольку каждая P-операция уменьшает семафор, а правильным результатом является только уменьшение $S_{j,k}$ на $W_{j,k}$, то мы выполняем $T_{i,k} - W_{j,k}$

Кононюк А.Е. Графы

V-операций над $S_{j,k}$, чтобы восстановить правильное значение $S_{j,k}$. Мы завершаем процесс узла v_k , выполняя $V_{k,r}$ V-операций над семафором $S_{k,r}$ для каждой дуги (v_k, v_r) , направленной из узла v_k .

Это преобразование проиллюстрировано на рис. 7 для узлов v_3 и v_4 графа вычислений, приведенного на рис. 2.

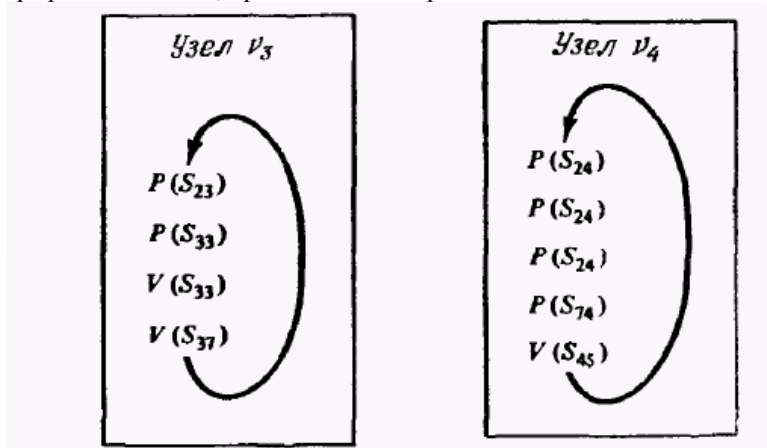


Рис. 7. P/V-система процессов для двух узлов графа вычислений на рис. 2.

Отметим, что граф вычислений может проверять и осуществлять ввод из нескольких источников за один шаг, тогда как P/V-системы могут проверять и осуществлять ввод из нескольких источников только с помощью последовательности проверок и вводов из отдельных источников. Неспособность к проверке и вводу из нескольких источников одновременно является ключевым моментом доказательства Патила ограниченности P/V-систем. Проблема состоит в том, что пока вы захватываете один ресурс, другой процесс может захватить второй ресурс, что приводит к тупику. Такой проблемы для графов вычислений не существует, так как источники не используются совместно процессами — не может быть двух узлов, разделяющих входные дуги. Это замечание существенно для построения P/V-системы, которая не попадала бы в тупик (завершалась) до тех пор, пока соответствующий граф вычислений не завершится.

Добавление P/V-систем к нашей иерархии моделей приводит ее к виду, изображенному на рис. 8.



Рис. 8. Добавление P/V-систем в иерархию моделей.

2.13.5. Системы передачи сообщений

Использование P- и V-операций для организации взаимодействий процессов в системе может осуществляться до тех пор, пока нет лучшего механизма связи. Одним из предложений по улучшению этого механизма является предложение использовать *сообщения*. Система с сообщениями — это набор процессов, которые взаимодействуют с помощью сообщений. Над сообщениями возможны две операции: *послать* и *получить*. Передача сообщения подобна V-операции, а прием сообщения подобен P-операции. Если при выполнении операции *получить* нет ни одного сообщения, то получатель ждет до тех пор, пока сообщение не будет послано.

На этом механизме основана схема моделирования, предложенная Риддлом. Эта модель кажется наиболее подходящей для моделирования протоколов в сетях ЭВМ. Риддл рассматривает (конечное) *множество* процессов, которые взаимодействуют с помощью *сообщений*. Сообщения посылаются и запрашиваются специальными процессами, называемыми *канальными* процессами (почтовые ящики). Канальные процессы предоставляют, что существенно, комплект сообщений, которые посланы, но еще не приняты, или комплект запросов на сообщения от приемников, которые выданы, но еще не удовлетворены. Другие процессы системы называются

Кононюк А.Е. Графы

программными процессами и описываются на языке моделирования программных процессов (ЯМПП). Пример системы из трех процессов приведен на рис. 9.

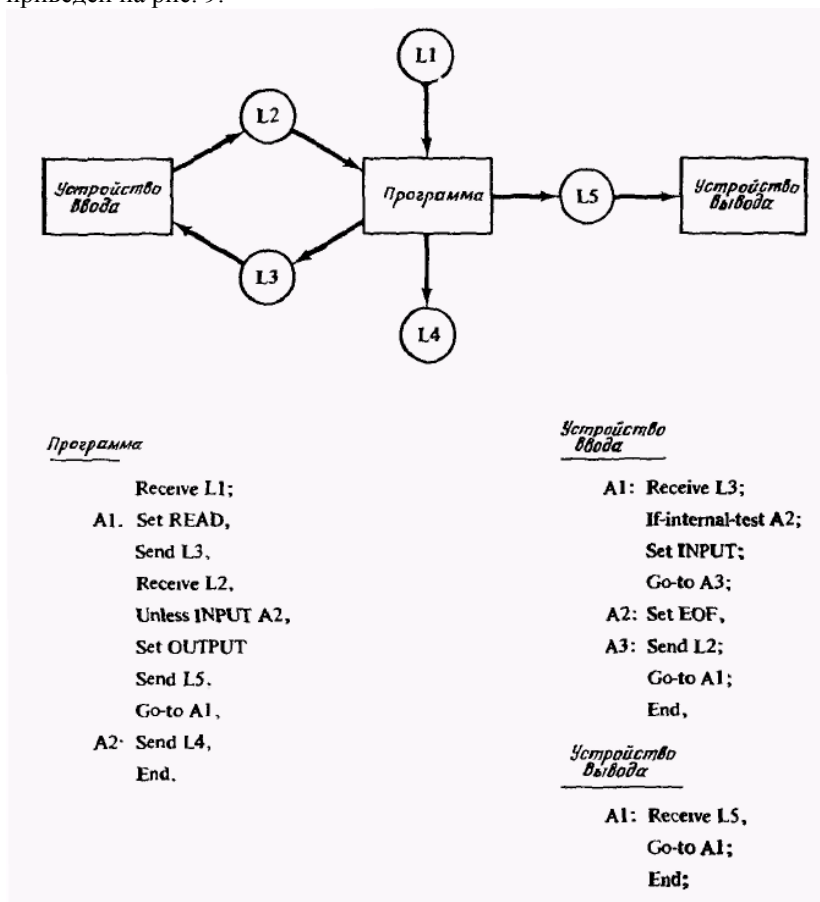


Рис. 9. Система процессов, описанных на языке моделирования программных процессов.

Как видно из примера, описание процессов на ЯМПП является, по существу, схемой. Интерес представляет только деятельность по передаче сообщений в системе. Сообщения являются абстрактными элементами, единственной характеристикой которых является тип.

Кононюк А.Е. Графы

Число типов сообщений в системе может быть только конечным. Сообщения посылаются из или принимаются в буфер сообщений в каждом из процессов. Существует только по одному буферу на процесс. Предложениями ЯМПП являются:

set t: Поместить сообщение типа *t* в буфер сообщений.

send l: Послать сообщение в буфер сообщений канального процесса *l*.

receive l: Запросить сообщение из канального процесса *l*. Ждать (если необходимо) до тех пор, пока не будет получено сообщение. Сообщение помещается в буфер сообщений.

unless t s: Проверить тип сообщения в буфере сообщений и перейти к предложению *s*, если сообщение имеет тип, отличный от *t*.

if-internal-test s: Моделировать внутреннюю, зависящую от данных, проверку. Либо продолжать обработку, выполняя следующее предложение, либо перейти к предложению с меткой *s*.

go-to s: Передать управление предложению *s*.

end: Завершить процесс.

Система с ЯМПП моделирует множество параллельных процессов. Каждый процесс стартует в начале своей программы и выполняет свою программу до тех пор, пока ему не встретится предложение *end*. Риддл показывает, как построить *выражение передачи сообщений*, которое представляет возможные потоки сообщений в системе и использует это выражение для исследования структуры системы и организации правильного функционирования. Это выражение передачи сообщений используется для тех же целей, что и язык сети Петри. Поэтому мы показываем, как описание системы процессов на ЯМПП может быть преобразовано в такую сеть Петри, что ее язык совпадает с выражением передачи сообщений из анализа Риддла. Это преобразование игнорирует выполнение отдельных предложений описания на ЯМПП, хотя с помощью незначительной модификации и они могли бы быть представлены в языке сети Петри. Для моделирования процесса сетью Петри используем по одной фишке на процесс в качестве программного счетчика. Присутствие сообщения в канальном процессе также представляется фишкой. Поскольку сообщения идентифицируются типом, то необходимо моделировать каждый тип сообщений в канальном процессе отдельной позицией. Очень важным свойством систем с ЯМПП является то, что число сообщений конечно. Каждый программный процесс также конечен. Только очередь сообщений занимает потенциально неограниченный объем памяти. Таким образом, способность моделировать канальные процессы и правильно представлять предложения *send* и *receive* являются наиболее важными аспектами преобразования описания на

Кононюк А.Е. Графы

ЯМПП в сеть Петри. Моделируя каналные процессы множествами позиций (по одной на каждый тип сообщений), мы можем представить предложение *send* переходом, который помещает фишку в позицию, представляющую соответствующий каналный процесс и тип сообщений. Предложение *receive* просто удаляет фишку из любой позиции канального процесса. Конкретная позиция, которая поставит фишку, определяет тип получаемого сообщения. Эта информация может использоваться в любом последующем предложении *unless*.

Единственным символом в выражении передачи сообщений является тип сообщений для тех сообщений, которые посылаются к или принимаются от канального процесса. Поскольку каждый переход в сети Петри приводит к появлению символа в языке сети Петри для этой сети Петри, то только предложения *send* и *receive* в системе с ЯМПП могут быть промоделированы. Таким образом, существуют два вида позиций в сети Петри. Один вид позиций, помеченных p_{l_i, m_j} , действует как счетчик числа сообщений типа m_j в канальном процессе l_j . Другой вид позиций представляет предложения *send* и *receive* программы ЯМПП. Пусть эти предложения однозначно помечены s_1, \dots, s_r . Мы пометим позицию, представляющую предложение s_i с сообщением типа m_j в буфере сообщений, символом p_{s_i, m_j} . Фишка в позиции, ассоциированной с предложением s_i , означает, что предложение s_i уже выполнено. Рис. 10 иллюстрирует, как предложения s_k : *send* и s_k : *receive* должны моделироваться сетью Петри.

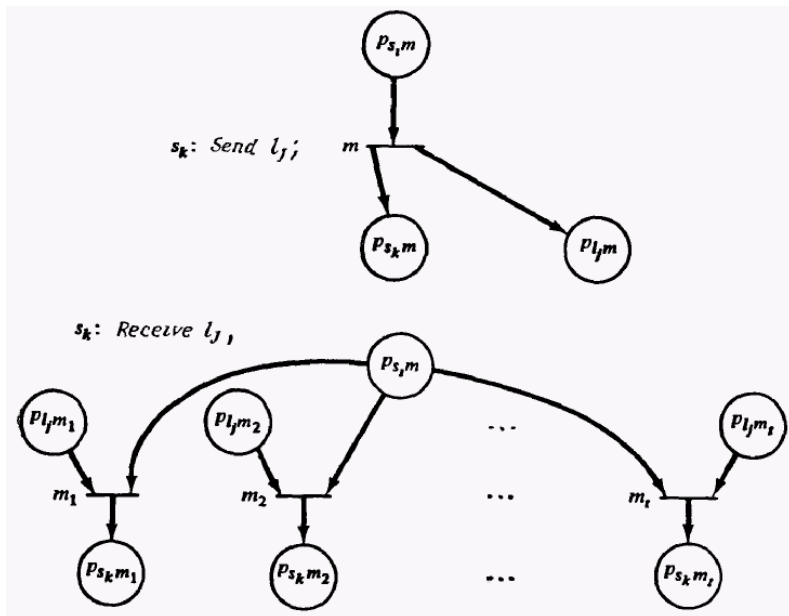


Рис. 10. Преобразование предложений *send* и *receive* в переходы сети Петри
вверху — модель предложения s_k *send* с сообщением типа m в буфере сообщений. Канальный процесс — l_i , *внизу*— модель предложения s_k *receive* из канального процесса l_i . Возможные типы сообщения в $l_i = m_1, m_2, \dots, m_r$

На рис. 10 позиция $p_{s_i,m}$ представляет позицию, ассоциированную с каким-либо предложением, которое предшествует предложению s_k . Теперь осталось показать, что существует возможность определения предложения, предшествующего другим предложениям в программе на ЯМПП. Отметим, что каждое предложение можно рассматривать как пару, состоящую из типа сообщения и номера предложения, поскольку одно и то же предложение с различными типами сообщений в буфере сообщений будет моделироваться сетью Петри различным образом. Наиболее очевидный способ определения предшественников предложения состоит в запуске в начале каждой программы на ЯМПП специального стартового предложения (которое становится стартовой позицией) и в порождении согласно описанию программ всех

Кононюк А.Е. Графы

возможных последующих предложений *send* и *receive* с соответствующим им содержимым буфера сообщений. Этот процесс повторяется для всех появляющихся предложений до тех пор, пока все предложения *send* и *receive* не будут порождены, а их последователи не будут идентифицированы. Поскольку число предложений в описании на ЯМПП и число типов сообщений конечно, то порождается только конечное число пар *предложение/Тип, сообщение*. Эта процедура подобна характеристическим уравнениям, используемым Риддлом для построения выражения передачи сообщений. На рис. 11 перечисляются предложения и их возможные последователи для системы с ЯМПП, изображенной на рис. 9.

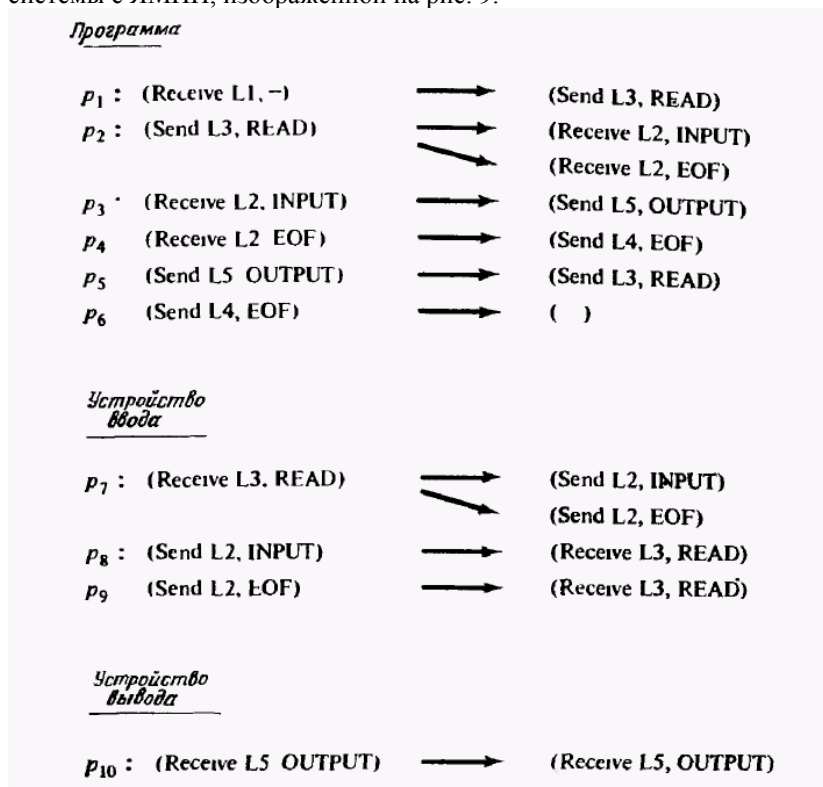


Рис. 11. Предложения и последователи для системы с ЯМПП, изображенной на рис. 9.

После того как последователи предложения определены, мы можем, используя эту информацию, идентифицировать возможные

Кононюк А.Е. Графы

предшественники предложения и, следовательно, построить сеть Петри, эквивалентную системе с ЯМПП, используя переходы, подобные приведенным на рис. 10. Специальная стартовая позиция является предшественником первого предложения каждого процесса системы. На рис. 12 система с ЯМПП, изображенная на рис. 9, преобразована в эквивалентную сеть Петри.

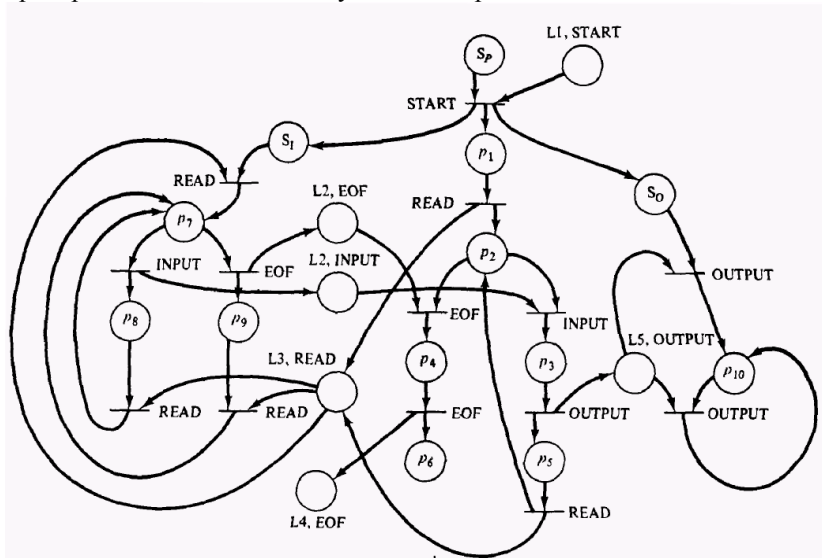


Рис. 12. Сеть Петри, эквивалентная системе с ЯМПП, изображенной на рис. 9.

Краткое описание преобразования систем передачи сообщений в сети Петри показывает, что эта модель включается по мощности моделирования в сети Петри. Оно показывает также, что множество выражений передачи сообщений, рассматриваемое как класс языков, является подмножеством класса языков сети Петри.

Поскольку P/V-системы можно моделировать системами передачи сообщений с сообщениями только одного типа, то P/V-системы включаются в системы передачи сообщений. Легко построить систему с сообщениями для решения задачи о курильщиках сигарет, поэтому включение P/V-систем в системы с сообщениями является собственным. С другой стороны, системы с сообщениями не способны воспринимать входные сообщения от нескольких источников одновременно и поэтому не эквивалентны сетям Петри.

Кононюк А.Е. Графы

При попытке моделирования перехода с несколькими входами может возникнуть один из следующих двух случаев:

1. Процесс будет пытаться получить фишки (сообщения) из всех своих входов, но будет недопустимым, и поэтому будет заблокирован, задерживая при этом фишки, которые нужны для того, чтобы позволить продолжать работу другим переходам. Это приведет к тупикам в системе с сообщениями, которые не соответствуют тупикам в сети Петри, что нарушает третье ограничение.

2. Процесс будет уклоняться от создания лишних тупиков, определяя, что оставшиеся нужные фишки отсутствуют, и возвращая фишки в позиции (канальные процессы), из которых они были получены. Такие действия могут выполняться произвольно часто, а это означает, что не существует ограничения на длину последовательности действий в системе с сообщениями, соответствующей ограниченной последовательности запусков переходов в сети Петри. Таким образом, при этом нарушается наше второе ограничение.

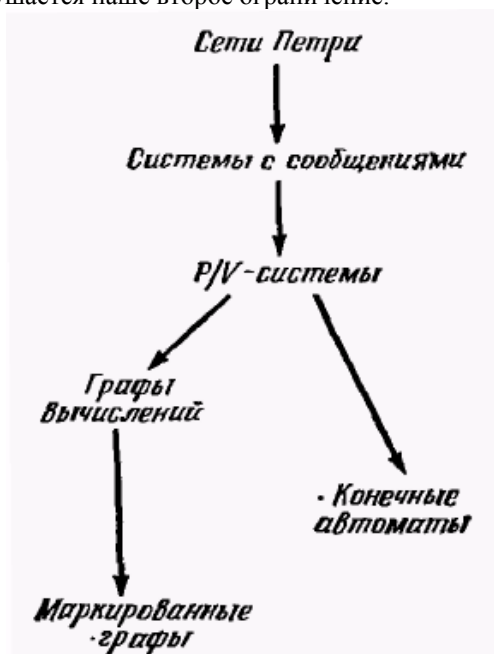


Рис. 13. Добавление систем с сообщениями к иерархии моделей.

Риддл представил преобразование, которое подпадает под случай 1 и приводит к лишним тупикам. В любом случае мы видим, что системы с

сообщениями не могут моделировать произвольные сети Петри (при сформулированных нами ограничениях). Поэтому в результате мы получаем иерархию, приведенную на рис. 13.

2.13.6. Графы UCLA

Сеть Петри является графовой моделью параллельных вычислений. Другая графовая модель разработана в Калифорнийском университете в Лос-Анджелесе под руководством профессора Эстрина. Эта модель является *сложной биологической графовой моделью вычислений* (или графовой моделью UCLA). В этой модели системы представляются графом со сложными ориентированными дугами. Сложная дуга — это дуга с (потенциально) многими источниками и назначениями.

Комбинационная логика управляет последовательностью операций в вершинах. Если входной логикой вершины является логика И (*), то для того, чтобы разрешить действия, фишки должны присутствовать на каждой входной дуге. При логике ИЛИ (+), фишки необходимы только на какой-либо одной входной дуге. Выполнение вершины удаляет разрешающие фишки на входных дугах и помещает фишки на выходные дуги в соответствии с выходной логикой. При выходной логике И фишки помещаются на все выходные дуги, тогда как при логике ИЛИ фишки помещаются на какую-либо одну выходную дугу. Число фишек, задействованных в данной паре вершина-дуга, называется *степенью* (или кратностью) этой пары и может быть любым целым неотрицательным числом.

На рис. 14 приведен пример графа UCLA. Заметим, что некоторые дуги имеют несколько источников (окончаний) и назначений (начал).

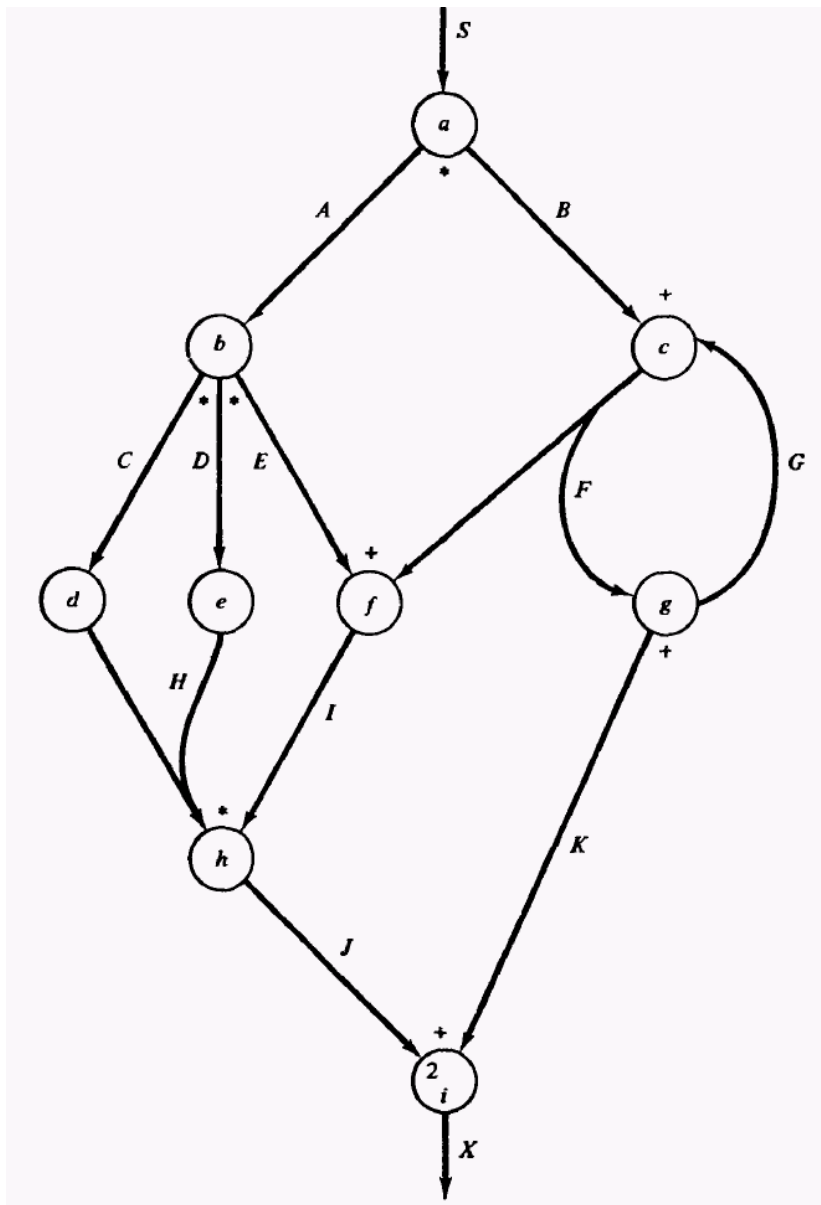


Рис. 14. Сложный биологический граф (граф UCLA).

Кононюк А.Е. Графы

Кроме того, отметим, что логика каждой пары дуга-вершина помечена в графе либо * для логики И, либо + для логики ИЛИ. Степень дуги указывается числом там, где дуга соединяется с вершиной. Степень опускается, если она равна единице, так же, как и обозначение логики, когда только одна дуга является входом в вершину. В приведенном примере вершина a может быть запущена, как только дуга S имеет фишки. Когда вершина a запускается, она удаляет фишку из дуги S и помещает фишки на дугу A и дугу B (логика И). Вершина g , с другой стороны, будет помещать фишки либо на дугу K , либо на дугу G (логика ИЛИ). Вершина i является разрешенной, когда присутствуют две фишки на дуге J или одна фишка на дуге K .

Определение 1. Граф UCLA есть шестерка $C = (V, A, L, Q, S, F)$, где $V = \{v_1, v_2, \dots, v_r\}$ — множество вершин; $A = \{a_1, a_2, \dots, a_s\}$ — множество дуг; $L = (L^-, L^+): V \rightarrow \{*, +\}$ — входное (L^-) и выходное (L^+) отображения логики на вершины графа; $Q = (Q^-, Q^+): V \times A \rightarrow N$ — входная (Q^-) и выходная (Q^+) степень каждой пары дуга-вершина; $S \in A$ — начальная дуга; $F \in A$ — конечная дуга.

Дуги графа определяются как упорядоченные пары множеств вершин. Первая компонента пары есть множество входных вершин, а вторая компонента — множество выходных вершин. Начальная дуга имеет пустое множество входных вершин, а конечная дуга — пустое множество выходных вершин.

Преобразование графа UCLA в сеть Петри достаточно просто из-за подобности этих систем. Каждая дуга в графе UCLA представляется позицией в сети Петри. Кроме того, представим вершину v позицией p_v и двумя переходами t'_v и t''_v . Первый переход t_v представляет инициирование операции, связанной с вершиной v , а второй переход представляет завершение этой операции. Это схематично изображено на рис. 15 (моделирование вершин графа UCLA переходами инициирования и завершения не строго обязательно, но удобно).

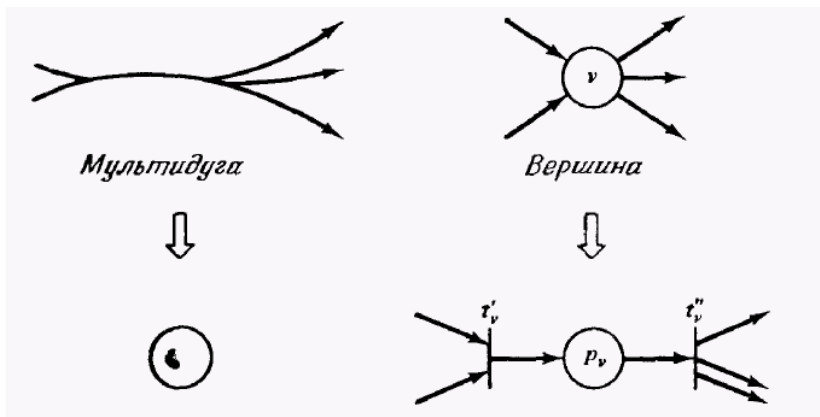


Рис. 15. Представление основных элементов графа UCLA в сети Петри.

На рис. 16 показано как входная и выходная логики графов UCLA представляются эквивалентными сетями Петри.

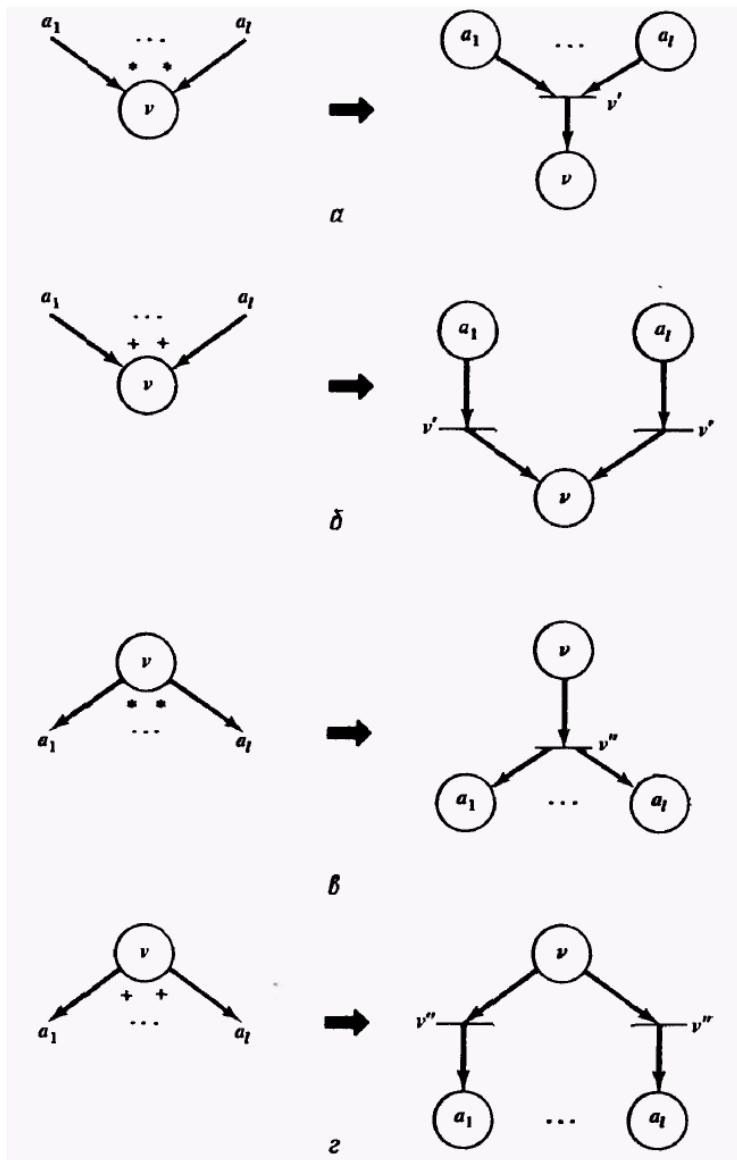


Рис. 16. Преобразование частей графа UCLA в сети Петри.

Степень больше единицы моделируется несколькими дугами между позициями и переходами в сети Петри.

На рис. 17 граф UCLA с рис. 14 преобразован в эквивалентную сеть Петри.

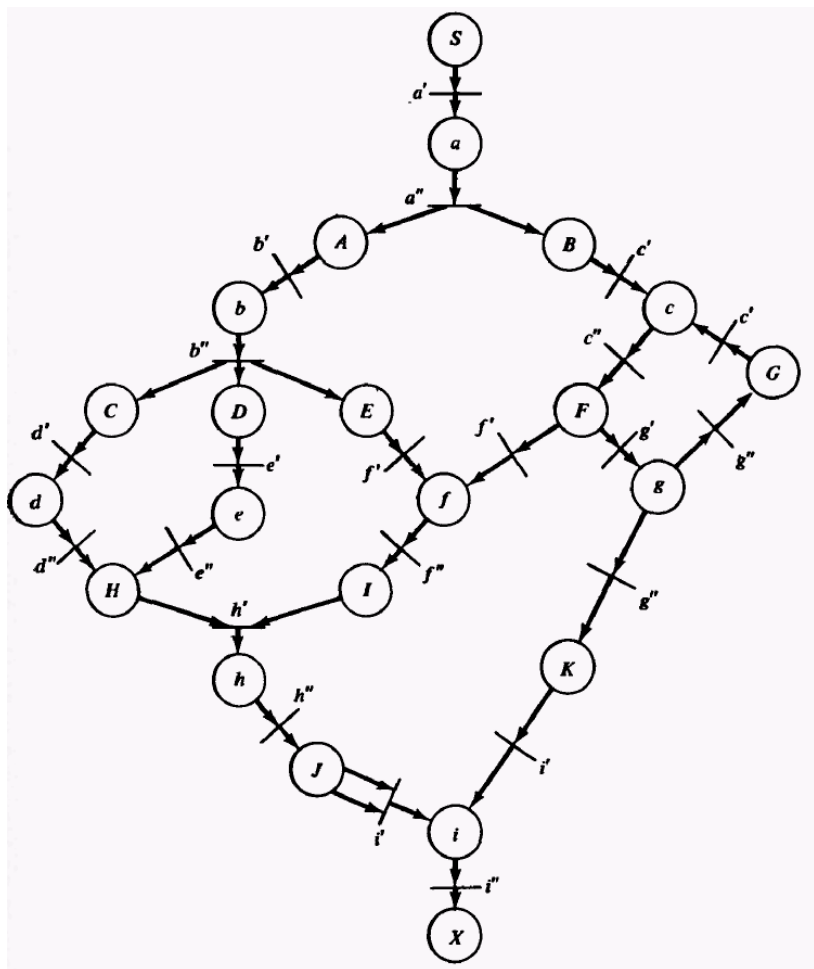


Рис. 17. Сеть Петри, эквивалентная графу UCLA, на рис. 14.

Это преобразование показывает, что мощность моделирования графов UCLA вкладывается в мощность моделирования сетей Петри. Очевидно, что сеть Петри может быть преобразована в эквивалентный граф UCLA посредством представления позиций дугами графа UCLA, а переходов — вершинами с входной и выходной логикой И. Таким

Кононюк А.Е. Графы

образом, эти две модели эквивалентны по своей мощности моделирования. На рис. 18 показана модифицированная иерархия моделей.

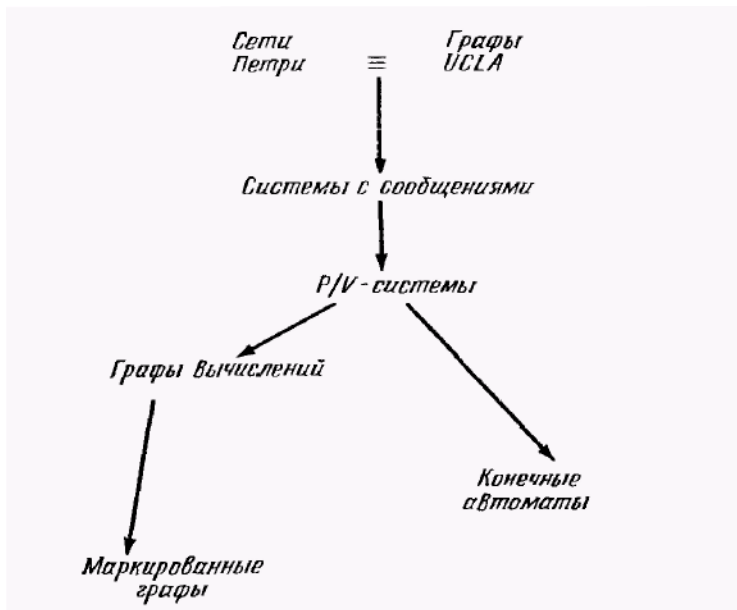


Рис. 18. Добавление графов UCLA к иерархии моделей.

2.13.7. Системы замещения и сложения векторов

Если просмотреть библиографию, то можно заметить, что в названиях большинства ссылок упоминаются не сети Петри, а *системы сложения векторов*. Системы сложения векторов были введены Карпом и Миллером как математическое средство анализа систем параллельных процессов. Благодаря их простому математическому определению системы сложения векторов обычно используются для формального доказательства свойств сетей Петри или подобных систем.

Определение 2. Система сложения векторов V есть пара $V = (B, s)$, где $B = \{b_1, b_2, \dots, b_m\}$ — множество из m векторов, называемых базисными векторами или векторами смещения. Вектор s есть начальный вектор. Все векторы состоят из n целых величин. Элементы s неотрицательны.

Кононюк А.Е. Графы

Множество достижимости системы сложения векторов V обозначается $R(V)$ и может быть определено рекурсивно с помощью следующего определения:

Определение 3. Множество достижимости $R(V)$ для системы сложения векторов $V = (B, s)$ есть наименьшее множество, для которого верно, что:

1. $s \in R(V)$;
2. Если $x \in R(V)$ и $(x + b_j) \geq 0$, то $(x + b_j) \in R(V)$, либо посредством следующего определения:

Определение 4. $x \in R(V)$, если существует такая последовательность $b_{i_1}, b_{i_2}, \dots, b_{i_k}$ базисных векторов, что

$$s + \sum_{j=1}^r b_{i_j} > 0 \text{ для всех } r: 0 \leq r < k.$$

$$x = s + \sum_{j=1}^k b_{i_j} \text{ и}$$

Из этих определений легко видеть, что системы сложения векторов и сети Петри эквивалентны. По данной сети Петри мы можем построить систему сложения векторов, начальным вектором которой является начальная маркировка, а базисные векторы взаимно однозначно соответствуют переходам. N компонент векторов системы сложения векторов соответствуют маркировкам n позиций сети Петри или (в случае базисных векторов) изменениям, происходящим из-за запуска соответствующего перехода.

Аналогично система сложения векторов может быть преобразована в эквивалентную сеть Петри использованием позиций для компонент векторов и переходов для представления базисных векторов. На рис. 19 иллюстрируется эквивалентность этих двух моделей.

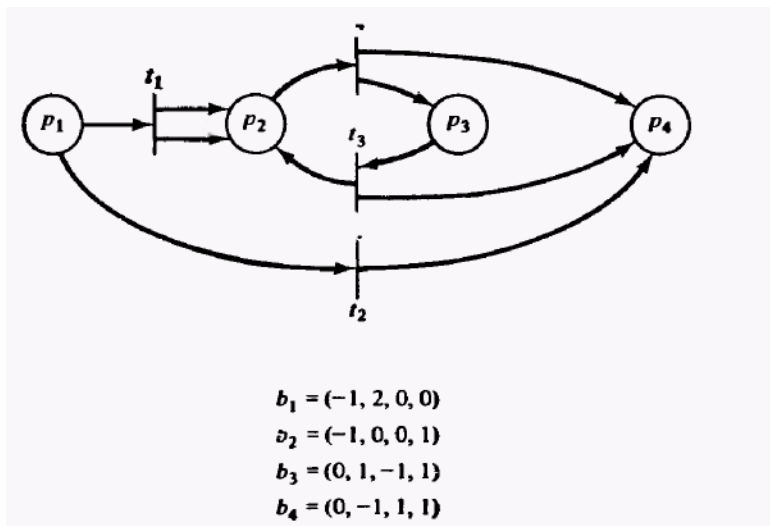


Рис. 19. Сеть Петри и эквивалентная ей система сложения векторов.

Фактически система сложения векторов эквивалентна сетям Петри без петель. Напомним, что в присутствии петель изменение может быть нулевым, а число фишек в позиции из петли должно быть ненулевым. Это не уменьшает мощность системы сложения векторов, поскольку мы видели, что сети Петри без петель эквивалентны обычным сетям Петри. Однако для более непосредственного моделирования сетей Петри с петлями в модели, подобной системе сложения векторов, Келлер определил *системы замещения векторов*.

Определение 5. Система замещения векторов состоит из начального вектора $s \geq 0$ и m пар векторов (U_i, V_i) , таких, что $U_i \leq V_i$.

Векторы U_i называются *векторами проверки*. Множество достижимости переопределяется так, что s принадлежит множеству достижимости, и если x входит в него и $x + U_i \geq 0$, то $x + V_i$ также входит в множество достижимости.

В модели системы замещения векторов явно определяется проверка на разрешение перехода от действия по запуску перехода. Эквивалентность систем замещения векторов сетями Петри (общего вида) очевидна.

Кононюк А.Е. Графы

Добавляя системы сложения векторов и системы замещения векторов в нашу иерархию, получаем иерархию, изображенную на рис. 20.

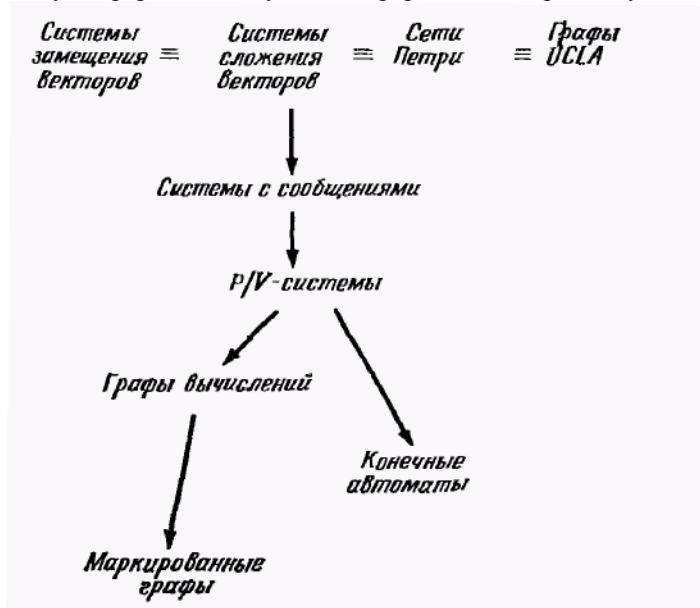


Рис. 20. Добавление систем сложения векторов и систем замещения векторов к иерархии моделей.

Важность систем сложения векторов и систем замещения векторов заключается в их простом математическом определении и полезности этого определения для доказательства математических свойств систем.

2.13.8. Расширенные модели сетей Петри

В качестве последнего дополнения к нашей иерархии вспомним о моделях расширенных сетей Петри, изученных ранее: сети Петри с областями ограничений, переходами исключающее ИЛИ, переключателями, сдерживающими дугами, приоритетами или временными ограничениями. Мы видели, что все эти модели эквивалентны машинам Тьюринга. Таким образом, эти модели строго включают модели сетей Петри. Окончательная иерархия моделей изображена на рис. 21.

Кононюк А.Е. Графы

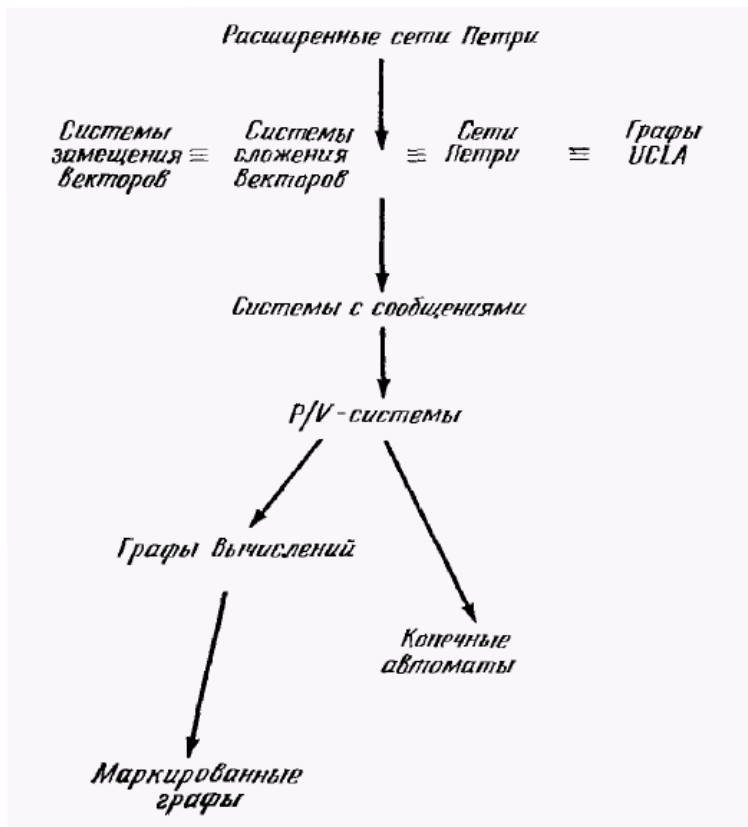


Рис. 21. Полная иерархия моделей параллельных вычислений.

ОБЗОР ТЕОРИИ КОМПЛЕКТОВ

Теория множеств используется в математике и вычислительной технике длительное время. *Теория комплектов* является естественным расширением теории множеств. Комплект, подобно множеству, есть набор элементов из некоторой области. Однако в отличие от множества комплекты допускают присутствие *нескольких* экземпляров одного и того же элемента. В теории множеств элемент есть либо элемент множества, либо не элемент множества. В теории комплектов элемент может входить в комплект *нуль раз* (не входить \ в комплект) или один, два, три или любое заданное число раз.

В качестве примера рассмотрим следующие комплекты над областью

$$\{a, b, c, d\}: B_1 = \{a, b, c\}, B_2 = \{a\}, B_3 = \{a, b, c, c\}, B_4 = \{a, a, a\}, B_5 = \{b, c, b, c\}, B_6 = \{c, c, b, b\}, B_7 = \{a, a, a, a, a,$$

$$b, b, c, d, d, d, d, d, d\}.$$

Некоторые комплекты являются множествами (например, B_1 и B_2). Так же как и в множествах, порядок элементов в комплекте не важен. Поэтому B_5 и B_6 являются одним и тем же комплектом (упорядоченные комплекты являются последовательностями).

В теории множеств основным понятием является отношение *включения*. Это отношение связывает элементы и множества и определяет, какие элементы являются членами каких множеств. Основным понятием теории комплектов является функция *числа экземпляров*. Эта функция определяет число экземпляров элемента в комплекте. Обозначим число экземпляров элемента x в комплекте B через

$$\#(x, B) \text{ (читается «число } x \text{ в } B\text{»)}.$$

Исходя из этого понятия, можем определить основы теории комплектов. Большинство понятий и обозначений заимствованы из теории множеств. Если мы ограничим число элементов в комплекте так, что $0 \leq \#(x, B) \leq 1$, то получим теорию множеств.

А.1. Членство

Функция $\#(x, B)$ определяет число экземпляров элемента x в комплекте B . Отсюда следует, что $\#(x, B) \geq 0$ для всех x и B . Мы различаем нулевой и ненулевой случаи. Элемент x является *членом* комплекта B , если $\#(x, B) > 0$. Это мы будем обозначать через

Кононюк А.Е. Графы

$x \in B$. Аналогично, если $\#(x, B) = 0$, то $x \notin B$.

Определим пустой комплект \emptyset , не имеющий членов (для всех x : $\#(x, \emptyset) = 0$).

А.2. Мощность

Мощность $|B|$ комплекта B есть общее число экземпляров элементов в комплекте

$$|B| = \sum_x \#(x, B).$$

А.3. Включение и равенство комплектов

Комплект A есть *подкомплект* комплекта B (обозначается $A \subseteq B$), если каждый элемент A является элементом B по меньшей мере не большее число раз: $A \subseteq B$ тогда и только тогда, когда $\#(x, A) \leq \#(x, B)$ для всех x . Два комплекта *равны* ($A = B$), если $\#(x, A) = \#(x, B)$ для всех x .

Из этих определений мы можем непосредственно показать, что $A = B$ тогда и только тогда, когда $A \subseteq B$ и $B \subseteq A$. $\emptyset \subseteq B$ для любого комплекта B ; из $A = B$ следует $|A| = |B|$; из $A \subseteq B$ следует $|A| \leq |B|$.

Комплект A *строго включен* в комплект B ($A \subset B$), если $A \subseteq B$ и $A \neq B$. Отметим, что $\#(x, A) < \#(x, B)$ не следует из $A \subset B$, хотя и $|A| < |B|$.

А.4. Операции

Над комплектами определяются четыре операции. Для двух комплектов A и B мы определим:

объединение комплектов

$$A \cup B: \#(x, A \cup B) = \max(\#(x, A), \#(x, B));$$

пересечение комплектов

$$A \cap B: \#(x, A \cap B) = \min(\#(x, A), \#(x, B));$$

сумму комплектов

$$A \vdash B: \#(x, A \vdash B) = \#(x, A) + \#(x, B);$$

разность комплектов

Кононюк А.Е. Графы

$$A - B : \#(x, A - B) = \#(x, A) - \#(x, A \cap B).$$

Эти операции обладают большинством ожидаемых свойств. Объединение, пересечение и сумма коммутативны и ассоциативны, кроме того, справедливы ожидаемые включения:

$$A \cap B \subseteq A \subseteq A \cup B;$$

$$A - B \subseteq A \subseteq A + B.$$

Различие между объединением и суммой очевидно:

$$|A \cup B| \leq |A| + |B|;$$

$$|A + B| = |A| + |B|.$$

К сожалению, различия между $A \cap B$ и $A - B$ нельзя так же легко проиллюстрировать, что объясняется невозможностью для операции разности удаления элементов из комплекта, которые не входят в него.

А.5. Пространство комплектов

Определим *область* D как множество элементов, из которых составляются комплекты. Пространство комплектов D^n есть множество всех таких комплектов, что элементы их принадлежат D и ни один элемент не входит в комплект более n раз. Иначе говоря, для любого $B \in D^n$:

1. Из $x \in B$ следует $x \in D$.
2. Для любого $x \in B$ $\#(x, B) \leq n$.

Множество D^∞ есть множество всех комплектов над областью D без какого-либо ограничения на число экземпляров элемента в комплекте.

А.6. Отображения Париха

Для конечной области $D = \{d_1, d_2, \dots, d_n\}$ существует естественное соответствие между каждым комплектом B над D и n -вектором $f = (f_1, f_1, \dots, f_n)$, определяемым соотношением $f_i = \#(d_i, B)$. Этот вектор известен как отображение Париха.

А.7. Примеры

Пусть $D = \{a, b, c, d\}$ — область. Тогда для следующих комплектов $A = \{a, b\}$, $B = \{a, a, b, c\}$, $C = \{a, a, a, c, c\}$ имеем:

Конюк А.Е. Графы

$$|A| = 2, |C| = 5;$$

$$A \cup B = \{a, a, b, c\} = B, \quad A \cup C = \{a, a, a, b, c, c\} = B \cup C;$$

$$A \cap C = \{a\}, \quad B \cap C = \{a, a, c\};$$

$$A + B = \{a, a, a, b, b, c\}, \quad A - B = \emptyset;$$

$$C - A = \{a, a, c, c\}, \quad C - B = \{a, c\}.$$

Кононюк А.Е. Графы

Литература

1. Применение теории графов связей в технике. - М.: Мир, 1974. – 96 с.
2. Котов В.Е. Сети Петри. - М.: Наука, 1984. – 160 с.
3. Питерсон Дж. Сети Петри и моделирование систем. - М.: Мир, 1984. – 297 с.
4. Артамонов Г. Т., Тюрин В. Д. Топология сетей ЭВМ и многопроцессорных систем. - М.: Радио и связь, 19с.
5. Корнеев В. В. Параллельные вычислительные системы. - М.: Нолидж, 19с.
6. Олифер В. Г., Олифер Н. А. Компьютерные сети: Принципы, технологии, протоколы // Учебное пособие, 2 изд. - СПб.:Питер, 20с.
7. International technology roadmap for semiconductors. Edition 2004. // 194 p.
8. The Thickness of Graphs / A Survey, P. Mutzel, T. Odenthal, M. Scharobrodt // Proc. Symp. Frontiers of Massively Parallel Computation. – 1998, Feb. - P. 96-101.
9. Thompson C. D. Area-Time complexity for VLSI // Proc. ACM, Symp. Theory of Computing– P. 81-88.
10. Theis T. N., Res J. The future of interconnection technology // IBM. Develop № 3. - P. 379-390.11.
11. Кристофидес Н. Теория графов: Алгоритмический подход. - М.:Мир, 319с.