Парадигма развития науки

А. Е. Кононюк

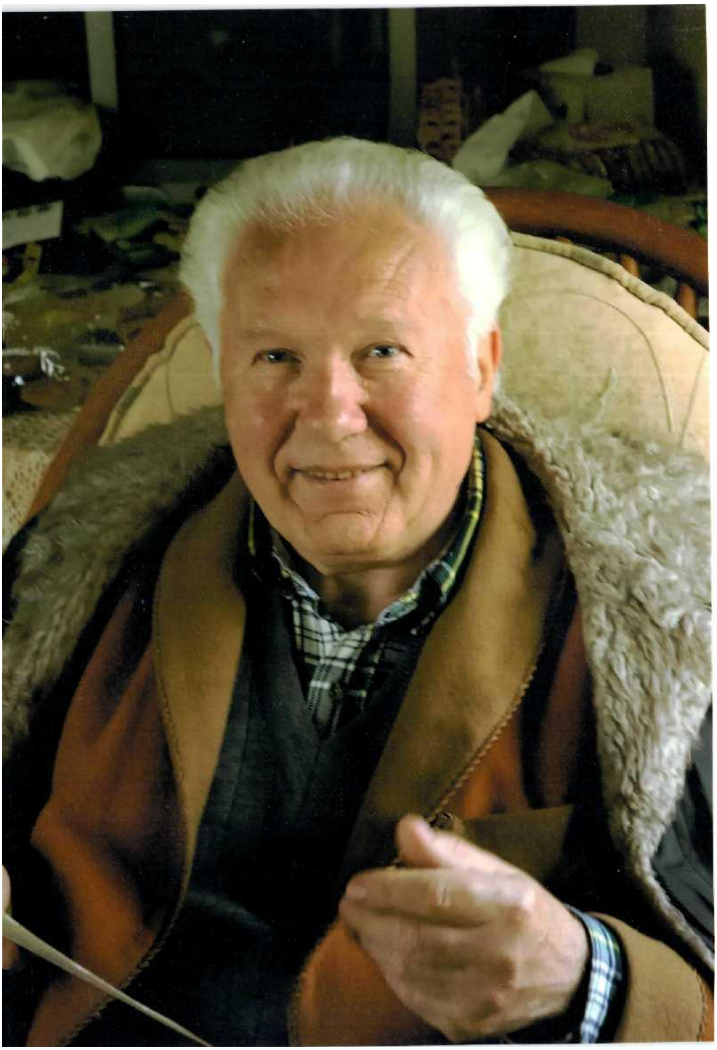
Истины и информация

(Фундаментальная теория представления истин и информации)

Книга 7

Фреймы как средства представления истин и информации

Киев
«Освіта України»
2016



Кононюк Анатолий Ефимович

Не познав информацию не познаешь истину

Структурная схема
парадигмы развития науки

УДК 51 (075.8)
ББК В161.я7
К65

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

Кононюк А. Е.
К213 Истины и информация. — В 16-и кн. Кн.7. — К.: Освіта України. 2016.—618 с.

ISBN 978-966-373-693-8 (многотомное издание)
ISBN 978-966-373-694-11 (книга 7)

Многотомная работа посвящена систематическому изложению общих формализмов, математических моделей и алгоритмических методов, которые могут быть исспользуемых при моделировании и исследованиях математических моделей истин и информации.

Развиваются представления и методы решения, основанные на теориях эвристического поиска и автоматическом доказательстве теорем, а также процедуральные методы, базирующиеся на классе проблемно-ориентированных языков, сочетающих свойства языков программирования и автоматических решателей задач отображения истин и информации различными математическими средствами.

В работе излагаются основы теории формализованного отображения истин и информации такими средствами математики как: множества, отношения, поверхности, пространства, алгебраические системы, матрицы, графы, математическая логика и др.

Для бакалавров, специалистов, магистров, аспирантов, докторантов всех специальностей.

УДК 51 (075.8)
ББК В161.я7

ISBN 978-966-373-693-8 (многотомное издание)
ISBN 978-966-373-694-11 (книга 7)

Оглавление

Предисловие……...11
Введение…….....13
1. Введение в теорию фреймов……17
1.1. Локальная и общая теории зрительного восприятия ……..21
1.2. Параллелизм ……22
1.3. Искусственный интеллект и процессы решения задач человеком……..24
1.4. Отслеживание образа куба 12……....25
1.5. Носит ли зрительное восприятие символьную форму……30
1.6. Видение комнаты…….....33
1.7. Анализ сцен и субфреймы…….....35
1.8. Перспективы и перемена точек наблюдений……37
1.9. Заслонения…….....41
1.10. Образы и системы фреймов……43
1.11. Априорное означивание …….....44
1.12. Системы фреймов и конкретные мыслительные операции Пиаже…….....45
2. Язык, понимание и сценарии……48
2.1. Слова, предложения и смысл ……48
2.2. Рассуждение ……50
2.3. Смысловая структура рассуждений…….....54
2.4. Перевод …….....58
2.5. Активная и пассивная формы интеллектуальной деятельности....59
2.6. Сценарии…….....62
2.7. Более сложные сценарии…….....66
2.8. Вопросы, системы и концептуальные случаи……..71
3. Обучение, память и парадигмы……..…….....74
3.1. Требования к памяти……..……76
3.2. Сопоставление образцов……77
3.3. Оправдание…….....79
3.4. Суждения и сети подобия …….....81
3.5. Группы, классы и географические аналогии……....84
3.6. Аналогии и альтернативные описания…….....87
3.7. Использование фреймов в эвристическом поиске…….....91
3.8. Фреймы в качестве парадигм……94
4. Управление ……..…….....97
4.1. Централизация управления…….....97
4.2. Фреймы и процесс согласования……....100
5. Пространственные образы…….....104

5.1. Местоположение и ориентация……104
5.2. Глобальная система пространственных фреймов.....106
5.3. Совершенствование системы…….....108
5.4. Эволюция…….....110
5.5. Вопросы измерений и количественных оценок ……113
5.6. Критика логистического подхода 117
6. Суть проблемы представления знаний…….....124
7. Характерные особенности фрейм-подхода к проблеме представления знаний…….....126
7.1. Фрейм - визуальный образ……129
7.2. Фрейм-сценарий…….....132
8. Способ формализации фреймов…….....137
8.1. Примеры формализованного представления фреймов-сценариев……. …… 140
8.2. Механизмы "приспособления" фрейма к реальной ситуации.. 144
9. Анализ общей структуры интеллектуальной системы фреймов.....147
9.1. Что такое интеллектуальная система фреймов?.....147
9.2. Система решения задач……150
9.3. Система восприятия……153
9.4. Эффекторная система……..157
10. Планирование и выполнение действий в интеллектуалных системах фреймов…….....159
10.1. Анализ систем решения задач……..159
10.2. Планирующая система «Решатель задач» ……165
10.3. Обобщение планов и планирование с помощью макрооператоров…….....173
10.3.1. Представление планов……..173
10.3.2. Обобщение планов…….....176
10.3.3. Особенности планирования с макрооператорами……....180
10.4. Обобщение пространств поиска решений и планирование в абстрактных пространствах……....185
10.4.1. Принцип образования иерархии пространств……185
10.4.2. Планирование в иерархии пространств…….....188
10.5. Стратегии выполнения действий…….....195
10.5.1. Исполнительные макрооператоры…….....195
10.5.2. Обобщенные процессы планирования и выполнения.....197
10.6. Особенности планирования при неполном описании мира (пространства, среды) ……...199
10.7. Планирование в процедуральных предписаниях……204
10.7.1. Построение простых планов…….....204
10.7.2. Построение условных и циклических планов……...208
10.8. Многоцелевое планирование……216

10.8.1. Общая постановка…….....216
10.8.2. Планирование с ограничениями…….....217
10.8.3. Кооперация отправителей……219
10.9. Особенности представления планов в динамическом пространстве……221
11. Языковые формы представления фреймов…….....230
11.1. Структура и задачи подсистемы языковых форм представления фреймов…….....230
11.2. Формальные грамматики……..234
11.2.1. Основные определения……234
11.2.2. Формальные грамматики……235
11.2.3. Трансформационные порождающие грамматики (ТПГ).....244
11.3. Классификация вопросно-ответных систем фреймов, понимающих естественный язык……253
11.3.1. Системы, использующие форматы частного вида.....253
11.3.2. Системы, основанные на запоминании текста.....253
11.3.3. Системы с ограниченной логикой…….....254
11.3.4. Системы с общим выводом……255
11.4. Синтаксический анализ……..……..256
11.4.1. Синтаксические анализаторы КС-языков……257
11.4.2. Анализаторы языков, описываемых трансформационными грамматиками……...258
11.4.3. Анализ естественных языков, описываемых расширенными сетями переходов……..259
11.5. Семантическая интерпретация……269
11.5.1. Общие сведения о семантической интерпретации.....269
11.5.2. Семантическая интерпретация в системах с ограниченной логикой…….....273
11.5.3. Семантическая интерпретация в системах с общим выводом.286
11.6. Вывод ответа…….....296
11.6.1. Доказательство и извлечение ответа в системах с общим выводом…….....296
11.6.2. Вывод ответа в системах с ограниченной логикой……...303
11.7. Формирование ответа в ограниченном естественном языке..305
11.8. Компьютеризация науки, ее проблемы и следствия.....307
11.8.1. Эпистемология и когнитивная наука…….....308
12. Методы и средства образования фреймов…….....312
12.1. Метод формального образования фреймов……312
12.2. Методы визуального отображения сцен…….....327
12.2.1. Структура и задачи визуального отображения сцен.....327
12.2.2. Формальное описание структуры понятия «сцена».....330
12.2.3. Трехмерными модели объектов (истин) ……336

12.2.4. Разбиение сцены на отдельные объекты……339
12.2.5. Монокулярное определение трехмерной структуры сцены.....352
12.2.6. Формирование моделей, опознавание объектов и
описание сцены…….358
12.2.7. Образовывающие операторы понятий…….....367
13. Предпосылки создания автоматизированных систем
образования фреймов …….....381
13.1. Системный характер сложных объектов (истин) и процессов
образования фреймов…. …… 381
13.2. Связи системных объектов и процессов с окружающей
средой и их функция…….....385
13.3. Структура САОФ и процессов образования фреймов.....390
13.4. Функционально-структурные свойства объектов и
процессов образования фреймов……396
13.5. Методические основы построения теории
автоматизированного образования фреймов…….....399
14. Многоуровневый итерациооный метод образования фреймов.....405
14.1. Многоуровневая декомпозиция процессов
образования фреймов…….....405
14.2. Примеры бразования фреймов технологических операций.....417
14.2.1. Фрейм операции «Поиск решений аналогов» ……..417
14.2.2. Фрейм операции «Преобразование процессов-аналогов».....421
14.2.3. Фрейм операции «Синтез технологических процессов».....423
14.2.4. Фрейм операции «Имитационное моделирование».....425
14.2.5. Фрейм операции «Анализ проектных вариантов».....427
14.2.6. Фрейм операции «Оценка вариантов» …….....428
14.2.7. Фрейм операции «Корректировка и улучшение»
проектного варианта…….....430
14.3. Диалоговые процедуры многоуровневого итерационного
метода образования фреймов……431
14.3.1. Рациональное разделение функций между разработчиком
фреймов и ЭВМ…….....431
14.3.2. Взаимодействие разработчика фреймов
с ЭВМ при различных формах диалога…….....434
14.3.3. Формализованный язык образования фреймов (ФЯОФ).....439
15. Математическое моделирование при автоматизированном
образовании фреймов…….....444
15.1. Иерахическая систеа математического моделирования
объктных фреймов…….....444
15.2. Образование типовых структурно-логических
фреймов моделей объектов……....456

15.3. Математическое моделирование процессов образования фреймов…….....462
15.4. Автоматизированное образование фреймов процессов по типовым структурно-логическим моделям…….....470
16. Построение моделей элементов технических устройств и формализованный язык их описания для автоматизированного образования фреймов…….....475
16.1. Виды моделей элементов технических устройств и способы их декомпозиции…….....475
16.2. СТРУКТУРНО-КИНЕМАТИЧЕСКИЕ МОДЕЛИ И МЕТОДЫ КОДИРОВАНИЯ ПОВЕРХНОСТЕЙ……...483
16.3. Операции и отношения используемые при образовании фреймов формы детали и размерных сязей ее элементов……489
16.3.1. Модели формы деталей машин…….....489
16.3.2. Математическая модель размерных связей элементов детали…….....495
16.4. Формализованный входной язык САОФ и правила образования на нем фреймов……...497
17. Синтез схем процесса образования фреймов……...507
17.1. Некоторые закономерности образования фреймов……..507
17.2. Фрейм функциональной структуры принципиальной схемы технологического процесса…….....519
17.2.1. Этапы (подфреймы) технологического процесса.....519
17.2.2. Фреймы алгоритмов назначения заготовительного этапа.....522
17.3. Фреймы алгоритмов формировании черновых и чистовых этапов (подфреймов) …….528
17.3.1.Формирование чистовых этапов (подфреймов) принципиальной схемы технологического процесса……533
18. Образование фреймов моделирования процесса проектирования технологических маршрутов…….....539
18.1. Закономерности синтеза фрейма маршрута механической обработки деталей…….539
18.2. Общая структура фреймов алгоритмов синтеза маршрута обработки деталей…….....552
18.3. Фреймы алгоритмов формирования маршрута обработки основных поверхностей деталей класса «тела вращения».....556
18.3.1. Дифференциация укрупненных операций в черновых этапах (подфреймах) …….....556
18.3.2. Дифференциация укрупненных операций в получистовых этапах (подфреймах) …….....558
18.3.3. Дифференциация укрупненных операций в чистовых этапах (подфреймах) …….....561

18.4. Образование фрейма автоматизации синтеза управляющих
программ……....562
18.4.1. Системная математическая модель перехода…….....562
18.4.2. Синтез траекторий фрезерной обработки сложных
областей на станках с ЧПУ…….....569
19. Образование фреймов процессов сборки технических
устройств…….576
19.1. Основные конструктивно-технологические свойства
технических устройств…….....576
19.2. Образование фрейма принципиальной схемы сборки.....590
19.3. Образование фрейма операционного технологического
процесса сборки…….....609
Литература…….611

Предисловие

Развитие информационно-поисковых систем высокого уровня, диалоговых систем, базирующихся на естественном языке, интерактивных человеко-машинных систем, предназначенных для совместного решения задач управления, проектирования, научных исследований и т.п., то есть развитие так называемых систем искусственного интеллекта (СИИ), а также роботов выдвинуло на первый план задачу представления информации в виде знаний в подобных системах. Необходимо подчеркнуть, что проблема представления знаний является принципиально новой, не встречавшейся ранее при создании различных автоматических и автоматизированных систем отображения и переработки информации и управления. В относительно небольшом объеме памяти интеллектуальные системы должны хранить большое число данных о мире задач, решаемых системой в процессе ее функционирования. Решение этой проблемы возможно лишь при специальной организации баз данных, одним из видов которой является **фреймовая организация**.

Создателем теории фреймов является М.Минский. Он рассматривает **два вида фреймов, которые принято называть статическими (или просто фреймами) и динамическими (сценариями). Фрейм любого вида - это та минимально необходимая структурированная информация, которая однозначно определяет данный класс объектов (истин)**. Наличие фрейма позволяет относить объект (истину) к тому классу, который им определяется. Простейшими примерами фреймов могут служить характеристические функции множеств в обычной математике. Однако в интеллектуальных системах в подавляющем большинстве случаев приходится иметь дело не с числовой, а с **символьной информацией** (например, текстами на естественном языке или зрительными изображениями). Для информации подобной природы определение фреймов представляет собой нелегкую проблему. М.Минским описаны некоторые подходы, использование которых многообещающе.

Книга построена следующим образом. В главе 1 излагается суть теории фреймов и рассматриваются вопросы, связанные с восприятием человеком зрительных образов. Подробно анализируются вопросы распознавания образов на базе системы фреймов. Обсуждаются

трудности, возникающие при изменении места положения субъекта относительно рассматриваемых предметов.

Глава 2 посвящена проблеме понимания смысла в предложениях естественного языка, организации предназначенных для этих целей систем фреймов и сопоставлению ряда точек зрения относительно способов построения программ, понимающих естественный язык.

В главе 3 обсуждаются вопросы обучения и его роль в восприятии новой информации; структура памяти и поиск фреймов, наиболее подходящих для представления некоторой ситуации; методы представления в теории решения задач.

Глава 4 посвящена проблеме организации системы поиска информации и управления подбором значений для терминальных вершин, обсуждению процесса выдвижения гипотез и оценке их правдоподобия, а также вопросам иерархии в системах фреймов.

В главе 5 затрагивается ряд спорных вопросов, связанных с наличием у человека общей картины мира и необходимостью иметь нечто подобное в системах искусственного интеллекта. М.Минский предлагает использовать в этих целях глобальный пространственный фрейм (GSF), хотя и отмечает ограниченность данной модели.

Мы надеемся, что специалисты, работающие в области разнообразных человеко-машинных систем в управлении, проектировании и научном эксперименте, получат возможность ознакомиться с идеями **фреймовой организации информации**, и это знакомство окажется полезным для их практической работы.

Введение

Излагаемый в этой работе материал освещает одну из наиболее важных и сложных проблем, обсуждаемых в рамках исследований по "искусственному интеллекту", - проблеме представления знаний в памяти ЭВМ. Суть ее заключается в том, что любое "осмысленное" поведение искусственной системы в условиях реального внешнего мира требует наличия у этой системы **специально организованной модели этого мира**. Данные ряда фундаментальных наук и в первую очередь психологии, генетики, цитологии позволяют утверждать, что **способность к информационному моделированию, к внутреннему воссозданию окружающей обстановки является основополагающей и необходимой в жизни и деятельности не только человека, но и животных.** Создание искусственного интеллекта является целью бурно развивающегося научного направления, вся история которого свидетельствует в пользу правильности **модельного подхода** к решению данной проблемы. Особую значимость приобретают вопросы представления знаний о свойствах, характеристиках и закономерностях реальных внешних сред для построения систем представления истин, обладающих широкими функциональными возможностями и высокой степенью автономии. Подобные кибернетические устройства разрабатываются как комплексные системы, способные воспринимать и анализировать информацию о внешнем мире, принимать самостоятельные решения и формировать управляющие воздействия для исполнительных органов с целью реализации принятых решений. Очевидно, что модель мира истин должна отражать совокупность объектов (истин) и отношений реального мира, существенных для решения некоторого множества задач, на которые нацелены задачи функционирования истин.

Сложность решения проблемы представления чрезвычайно велика, и это объясняется в первую очередь недостатком наших знаний о механизмах человеческого мышления. Результаты, полученные при изучении человеческого интеллекта, оказывают все большее влияние на решение проблемы искусственного интеллекта. Последние в свою очередь помогают ученым глубже понять принципы работы человеческого мозга.

Существует несколько направлений исследований в области искусственного интеллекта, отличающихся, в частности, своими подходами к проблеме представления знаний. Наиболее известные

методы машинного представления знания: **логистический, теоретико-графовый, а также метод, использующий для описания мира вектор-функции, определенные на нормированных пространствах**.

Логистический метод, используемый, например, при построения такой известной системы для решения задач, как STRIPS (Р.Файкс, Н.Нильсон), основан на привлечении **языка исчисления предикатов первого порядка для формирования модели внешнего мира, на использовании понятий пространства состояний, а также методов доказательства теорем и эвристических методов как основных механизмов поиска решений.** Модель в данном случае представляет собой систему аксиом - предложений языка исчисления предикатов первого порядка, определяющую всю совокупность объектов (истин), характеристик и свойств внешнего мира истин, существенных для его функционирования. В случае относительно простых, статических сред системы аксиом выглядят достаточно компактно, а существующие поисковые процедуры (такие, как метод резолюций и его модификации, эвристика "анализа целей и средств" и др.) оказываются мощным средством для выработки планов действий. Но как только возникает задача **создания машинной модели реальной, динамичной, недетерминированной внешней среды**, то логистический подход оказывается несостоятельным вследствие резкого усложнения, как самих конструкций моделей, так и формализованного представления в них смысловых отношений между элементами внешней среды.

Кроме того, с усложнением внешнего мира число формализующих его аксиом лавинообразно растет, что приводит не просто к громоздкости машинной модели окружающей среды, но к ряду принципиальных трудностей. Они связаны, во-первых, с выбором только того подмножества из всего множества аксиом, которое имеет непосредственное отношение к решаемой в данный момент времени задаче, и, во-вторых, с активизацией и выполнением лишь тех дедуктивных процедур, которые существенны для получения конечного результата. Проблема заключается совсем не в том, чтобы из множества выведенных отобрать нужные теоремы, а в том, чтобы не выводить ненужных.

Аналогичные по характеру трудности возникают при использовании **теоретико-графового метода**, в рамках которого **модель внешнего мира представляется в виде графа**, **узлы которого соответствуют**

возможным состояниям внешней среды, а дуги - возможным действиям, переводящим систему из одного состояния в другое.

Ограничены возможности описания реального мира и с помощью **вектор-функций**, определенных на нормированных пространствах, что имеет место в случае методов, использующих основные положения теории автоматического управления (Ф.М.Кулаков).

Одним из возможных путей решения проблемы явился **подход, предполагающий использование семантических связей между понятиями, включенными в модель внешнего мира и учет прагматики внешнего мира.** Это нашло свое отражение в исследованиях ряда ученых, например, П.М.Амосова в области развития М-сетей, Д.А.Поспелова, В.Н.Пушкина и Ю.И.Клыкова по ситуационному управлению, а также ряда зарубежных специалистов, в том числе Р.Шенка по теории семантической зависимости (conceptual dependency), Дж.Уилкса в области семантики предпочтений (preference semantics), Ч. Ригера по теории семантических наложений (conceptual overlays) и др.

Наиболее значительной среди появившихся 70-е годы XX в. была теория фреймов (frames) М. Минского, привлекшая к себе пристальное внимание специалистов в области искусственного интеллекта (Р.Шенк, Р.Абельсон; Дж.Лаубш; Дж.Майлопулос, П.Коэн, А.Борджида, Л.Шугар; Д.А.Поспелов, Е.Н.Ефимов; Н.Н.Перцова и др.). Впервые теория была опубликована в 1974 г.

В своей теории М.Минский отказался от попыток формировать модель внешнего мира на основе разрозненных, отдельных фактов или понятий. Центральным моментом является его утверждение о том, что **любая <u>машинная модель</u>, отражающая сложности реального мира, должна строиться в виде достаточно большой совокупности определенным образом сформированных данных - <u>фреймов</u>, представляющих собой <u>модели стереотипных (часто повторяющихся) ситуаций</u>. <u>Ситуация</u>** понимается здесь в обобщенном смысле, т. е. это **может быть <u>действие, рассуждение, зрительный образ, повествование и т.д</u>. <u>Фрейм представляет собой не одну конкретную ситуацию, а наиболее характерные, основные моменты ряда близких ситуаций, принадлежащих одному классу</u>**. В переводе с английского **frame означает скелет, остов, рамка**, что лишний раз подчеркивает общность представленных в нем сведений о

моделируемом явлении. **Графически фрейм можно изобразить в виде сети, состоящей из узлов и связей между ними. Каждый узел представляет собой определенное понятие, которое - и в этом заключается основной смысл теории - может быть, а может и не быть задано в явном виде.** В последнем случае оно может быть конкретизировано в результате процесса согласования данного фрейма с некоторой конкретной ситуацией, имеющей место во внешнем мире. **Незаданные в явном виде узлы называются терминалами**. Они образуют нижние уровни графовой структуры, тогда как на верхних уровнях располагаются понятия, которые всегда справедливы в отношении представляемой данным фреймом ситуации. Таким образом, **совокупность заданных в явном виде узлов - понятий образует основу для "понимания" любой конкретной ситуации из определенного для данного фрейма класса ситуаций.** "Понимание" происходит путем конкретизации терминалов и согласования возможных для каждого из них понятий с вполне определенной, существующей во внешнем мире обстановкой. **Центральным моментом является использование одних и тех же терминалов различными фреймами, что позволяет координировать информацию, собираемую из разных источников.** Группы связанных между собой фреймов объединяются в системы, которые могут отражать действия, причинно-следственные связи, изменения понятийной точки зрения и т. д.

В своей теории М.Минский не проводит границы между теорией человеческого мышления и теорией построения "думающих" машин (искусственного интеллекта). Он полагает, что процессы человеческого мышления базируются на хранящихся в его памяти материализованных, многочисленных запомненных структурах данных - **фреймах, с помощью которых человек осознает зрительные образы (фреймы визуальных образов), понимает слова (семантические фреймы), рассуждения, действия (фреймы-сценарии), повествования и т.д. Процесс понимания при этом сопровождается активизацией в памяти соответствующего фрейма и согласованием его терминальных вершин с текущей ситуацией**. В случае неудачи из памяти с помощью **сети поиска информации**, объединяющей **системы фреймов**, "выбирается" другой фрейм, терминалы которого, возможно, окажутся между собой в более подходящих отношениях применительно к той же рассматриваемой ситуации. Процесс последовательной замены одного фрейма другим особенно наглядно проявляется в таких областях человеческого мышления, как понимание естественного языка, рассуждение, вывод

по аналогии и др. Это следует из наших интуитивных представлений о процессе мышления, который начинается с наводящих на мысль, но несовершенных образов, прогрессивно заменяемых лучшими, но всё еще несовершенными идеями.

Представление знаний о мире с помощью систем фреймов оказывается весьма плодотворным во многих областях исследований по искусственному интеллекту, начиная от понимания естественного языка и кончая проблемами машинного восприятия слуховых и зрительных образов. Подтверждением тому служат многочисленные работы таких известных специалистов в области искусственного интеллекта, как Р.Шенк, Р.Абельсон, Ч.Ригер, Е.Чарняк, Дж.Уилкс и др., посвященные конкретизации и развитию теории фреймов.

Вместе с тем следует отметить, что подход М.Минского к решению проблемы представления знаний не лишен недостатков. В частности, человек способен понять не только известные, но и новые ситуации, к чему фреймы пока еще не приспособлены. Видимо, разработка механизмов машинного "понимания" новых фактов на базе системы фреймов точно так же, как и развивающихся во времени действий, еще впереди.

Существует еще целый ряд вопросов, которые не затрагиваются автором в данной книге. Например, проблема принятия решений на базе системы фреймов, в том числе проблема планирования действий истин. Однако это не следует понимать как недостаток данной работы, поскольку в ней ставится цель изложить в ней общие вопросы теории фреймов и дать толкование ее основных положений.

1. Введение в теорию фреймов

Теоретические исследования, проводимые в области представления истин и информации в целом являются недостаточно общими для того, чтобы объяснить либо с практической точки зрения, либо феноменологически эффективность человеческого мышления. Основные структурные элементы, образующие фундамент для

развертывания процессов восприятия истин, хранения информации, мышления и разработки языковых форм общения, должны быть более крупными и иметь более четкую структуру; их фактическое и процедуральное содержание следует более тесно увязывать друг с другом с тем, чтобы получить возможность объяснить феномен силы и "быстродействия" человеческого мышления. Аналогичной точки зрения придерживаются специалисты научных центров, работающие над решением проблем искусственного интеллекта. Известно, например, предложение А.Ньюэлла и Г.Саймона решать проблему представления в терминах "пространств задач" или предложение С. Пейперта и М.Минского, разбить всю совокупность сведений, необходимых системе представления истин и информации, на "микромиры". Иную форму те же взгляды принимают в работах известных теоретиков, таких как Р.Шенк, Р.Абельсон и Д.Норман, которые используют более крупные структуры для изучения механизмов понимания естественного языка. В этом проявляется стремление ученых выйти за рамки исследований чисто бихейвористического и формально-логического направлений и отказаться от попыток решать проблему представления истин и информации с помощью наборов разрозненных простых структур данных. (Бихейворизм (от англ. behavior - поведение) - одно из направлений в психологии, в основе которого лежит утверждение о том, что предметом психологии является поведение, а не сознание. Основной задачей психологов, по мнению основоположника бихейворизма Дж.Уотсона, является установление объективно наблюдаемых отношений в соответствии с известной схемой "стимул - реакция" и сведение к ним всех понятий о внутренних, психических процессах.)

В настоящей работе делается попытка связать воедино результаты некоторых из вышеупомянутых исследований и создать единую и стройную теорию представления истин и информации на основе фреймов и сценариев.. Отмечаются ее недостатки, поскольку здесь ставится больше вопросов, чем дается на них ответов.

Отправным моментом для данной теории служит тот факт, что человек, пытаясь познать новую для себя истину (ситуацию) и отразить ее в виде информации или по-новому взглянуть на уже привычные вещи, выбирает из своей памяти некоторую **структуру данных (образ истины)**, называемую М.Минским нами **фреймом**, с таким расчетом, чтобы путем изменения в ней отдельных деталей сделать ее пригодной

для понимания и отражения более широкого класса истин и информации (явлений или процессов).

Фрейм является структурой данных для представления стереотипной ситуации (истины). **С каждым фреймом ассоциирована информация разных видов.** Одна ее часть указывает, каким образом следует использовать данный фрейм, другая - что предположительно может повлечь за собой его выполнение, третья - что следует предпринять, если эти ожидания не подтвердятся.

Фрейм можно представлять себе в виде сети представления истин и информации, состоящей из узлов и связей между ними. "Верхние уровни" фрейма четко определены, поскольку образованы такими понятиями, которые всегда справедливы по отношению к предполагаемой ситуации. На более низких уровнях имеется много особых вершин-терминалов или "ячеек", которые должны быть заполнены характерными примерами или данными.

Каждым терминалом могут устанавливаться условия, которым должны удовлетворять его задания. Простые условия определяются маркерами, например, в виде требования, чтобы заданием терминала был какой-либо субъект, или предмет подходящих размеров, или указатель на субфрейм определенного типа. (Субфреймы, фреймы и суперфреймы - это иерархически упорядоченные элементы, образующие системы фреймов). Более сложными условиями задаются отношения между понятиями, включенными в различные терминальные вершины.

Группы семантически близких друг к другу фреймов объединены в систему фреймов. Результаты существенных действий представляются в виде трансформаций между фреймами системы. Это дает возможность моделировать такие понятия, как внимание и ценность информации, сделать более экономичными некоторые типы вычислений, а также показать эффективность использования фреймов в системах представления истин и информации .

При зрительном восприятии образов истин системы фреймов используются следующим образом: различные фреймы соответствуют различным позициям наблюдателя, анализирующего одну и ту же сцену, а трансформации между ними отражают результаты перемещения наблюдателя из одного места в другое. Для систем других типов различия между фреймами могут соответствовать

результатам выполнения каких-либо действий, определенным причинно-следственным связям между объектами (истинами) мирового пространства или разным точкам зрения по одним и тем же вопросам. Одни н те же терминалы могут входить в состав нескольких фреймов системы - это один из центральных моментов теории, позволяющий согласовывать информацию, поступающую из различных источников (истин).

Теория фреймов во многом выигрывает благодаря возможности использования в ней ожиданий и других видов предположений. Терминалы фрейма в обычном своем состоянии заполнены так называемыми "заданиями отсутствия" или заранее заготовленными значениями, т. е. информацией о деталях истин (частностях), которые не обязательно должны присутствовать в какой-либо конкретной ситуации. Связь заданий отсутствия со своими терминалами не является жесткой и неизменной, поэтому они легко могут быть заменены другими истинными сведениями, более подходящими к текущей ситуации. Задания отсутствия могут, таким образом, выполнять роль переменных, служить для аргументации с помощью примеров (что часто делает излишним применение логических кванторов), представлять информацию общего вида и описывать наиболее вероятные случаи, указывать на способы проведения полезных обобщений и т. д.

Системы фреймов связаны, в свою очередь, сетью поиска информации. Если предложенный фрейм нельзя приспособить к реальной ситуации, т. е. если не удается найти такие задания терминалов, которые удовлетворяют условиям соответствующих маркеров, сеть поиска информации позволяет выбрать более подходящий для данной ситуации фрейм. Подобные структуры дают возможность **использовать в системах фреймов различные методы представления информации**, что имеет особое значение для разработки механизмов понимания.

После выбора фрейма в процессе согласования терминалам присваиваются такие значения, которые удовлетворяют всем условиям соответствующих маркеров. Ход процесса согласования частично контролируется информацией, связанной с самим фреймом (включая указания на то, как реагировать на непредвиденные обстоятельства), и в значительной степени опытом решения аналогичных или близких по смыслу задач. Если согласование внешних данных с маркерами

терминалов неудовлетворительное, то информация, полученная на его основе, может быть с успехом применена при выборе альтернативного фрейма.

Отметим, что схемы, предложенные в настоящей работе, не вполне совершенны. Во-первых, некоторые варианты представления информации обсуждаются безотносительно к тем процессам, в которых они должны использоваться. Иногда приводятся только лишь описания свойств, которыми следует снабдить те или иные структуры. Маркеры и задания терминалов рассматриваются так, будто известны их соединения и связи с более крупными структурными единицами.

Структуры, связывающие в единое целое системы фреймов, могут оказаться полезными при объяснении ряда явлений, характерных для естественного интеллекта.

1.1. Локальная и общая теории зрительного восприятия истин и информации

Когда мы входим в комнату, нам кажется, что мы видим всю возникающую перед глазами картину (сцену) с одного взгляда. В действительности же **зрительное восприятие - это длительный процесс**. На все требуется время: и на то, чтобы рассмотреть детали и собрать о них нужную информацию, и на то, чтобы, сделав предположения, проверить их и прийти к определенным заключениям, и на то, чтобы оценить полученные данные с учетом преследуемых целей, собственных знаний и ожиданий. Тем не менее, все это происходит настолько быстро и естественно, что невольно вызывает удивление и требует объяснений.

Некоторые ученые считают неудовлетворительными те теории, в которых делается попытка трактовать феномен зрительного восприятия с позиции **дискретных, последовательных, символьных процессов**. Им кажется, что хотя машинные программы, написанные на основе этих теорий, действительно могут демонстрировать эффект "видения", для живых существ они неприемлемы ввиду грубости восприятия и малого быстродействия. Однако обычно предлагаемая альтернатива, относящаяся к крайнему случаю холизма или идеалистической "философии целостности", не может быть технически реализована. Ниже будут приведены доводы в пользу того, что **следующие один за другим символьные процессы могут объяснить**

суть явлений (истин), которые нам кажутся мгновенными и завершенными и которые имеют место при анализе сцен.

Ряд теоретиков, придерживающихся ранних гештальт-психологических концепций, стремилась объяснить феномен зрительного восприятия с помощью имеющегося в человеческом мозге электрического поля, но уже в 30-х годах XX столетия стала ясна несостоятельность подобных воззрений. Их последователи, включая приверженцев идеи интегральных преобразований, использования голограмм и интерферентных явлений, многого достигнуть также не сумели. И все же, несмотря на эти неудачи, большинство ученых по-прежнему полагает, будто требуемая скорость восприятия может быть обеспечена только посредством некоторого глобального параллельного процесса, подобного волновому.

Излагаемая в настоящей работе теория фреймов рассматривает все основные вопросы представления истин и информации на основе фреймового подхода. Общим является стремление раскрыть суть процессов, лежащих в основе соотнесения сенсорных данных либо с общими понятиями, либо с их частями, но методы, предложенные для обоснования выдвинутых положений, в корне отличны друг от друга. Гештальт-психологи стремились доказать, что эти процессы основаны главным образом на взаимодействии небольшого числа универсальных и весьма эффективных процедур, но неудачи с их эффективным выделением показали ограниченность этой идеи. В теории фреймов упор делается на многочисленность взаимодействий между сенсорными данными и образованной в процессе обучения громадной сетью символьной информации. И хотя эти взаимодействия должны, в конечном счете, основываться на том или ином наборе общих принципов, в теории фреймов изучение мыслительных процессов отделено от вопросов происхождения и развития систем представления истин и информации.

1.2. Параллелизм

Может ли оказаться целесообразной параллельная обработка информации? Вопрос этот следует считать чисто техническим (скорее, риторическим) в большей степени, чем может показаться на первый взгляд. Действительно, на уровне выявления простейших визуальных черт, текстурных элементов, характерных особенностей при стереоскопическом зрении или двигательном параллаксе

использование параллельной обработки информации следует считать целесообразным. На следующем, "предметном" уровне труднее представить себе, каким образом можно использовать параллелизм, хотя в этой связи следует отметить работы А.Гузмана по выделению "ядер" и объединению их в связанные области или Д.Вальтца по использованию семантических корней для отсортировки теневых линий, выполняемой на специальной параллельной сети.

Однако на более высоких уровнях мыслительной деятельности целесообразность применения параллелизма встречает ряд принципиальных возражений. В работах по распознаванию образов было предложено довольно много схем для выполнения параллельных операций - персептроны, интегральные преобразования и т.д. Эти схемы, интересные и с математической, и с вычислительной точек зрения, видимо, могли бы служить в качестве составных частей теории обработки сенсорной информации, но не более того. **Интегральные методы хороши, главным образом, для работы с изолированными двухмерными изображениями,** но на их основе нельзя решить задачу выделения и распознавания объектов в сложных трехмерных сценах. Почему? При анализе сложных сцен должны быть правильно выделены области, принадлежащие различным объектам (истинам), ибо только в этом случае воспринимаемая картина обретает смысл; однако для решения этой задачи, которая эквивалентна традиционной в гештальт-психологип проблеме "объект - фон", требуется так много усилий, что, как отмечалось в работе М.Минского и С.Пейперта, сама возможность и даже **целесообразность разработки методики изолированного распознавания ставится под сомнение**. Для трехмерных изображений эта проблема еще более осложняется как искажением перспективы, так и тем обстоятельством, что отдельные части предметов оказываются невидимыми из-за других объектов. В новых знаковых теориях используются методы выработки гипотез с последующим их подтверждением; эти методы кажутся нам более продуктивными. **Трудно решить любую по-настоящему сложную проблему, не уделив самого пристального внимания ее отдельным составным частям**. Однако, можно представить себе более эффективный (по сравнению с просто идеей параллелизма) последовательный процесс, при котором крупные, сложные знаковые структуры рассматриваются в качестве простейших операндов. Это открывает теоретически новую возможность для быстрого поиска крупных субструктур и, по-видимому, позволит найти секрет быстродействия механизмов человеческого мышления и восприятия зрительной информации.

1.3. Автоматизированные сети представления истин и информации и процессы решения задач человеком

Искусственный интеллект и процессы решения задач человеком

Мы не будем проводить границы между теорией человеческого мышления и теорией построения "думающих" систем (встроенных в "думающие" сети) : разделять их пока не имеет смысла, поскольку как в той, так и в другой области знаний отсутствуют концепции, достаточно общие для объяснения и тем более для моделирования сложной интеллектуальной деятельности. Однако одно отличие все же имеется. Дело в том, что у специалистов-психологов, работающих над проблемами интеллекта, наблюдается определенная тенденция к сокращению числа различных механизмов, включаемых в модели функционирования человеческого мозга. Это ведет к попыткам достигнуть большего эффекта с помощью меньшего, чем может быть обосновано, числа основных механизмов мышления. Такие теории уделяют недостаточно внимания как вопросам управления психической деятельностью, так и уточнению наших знаний об отдельных интеллектуальных процессах. Ученые, видимо, сосредоточили все свои усилия именно на этих вопросах, но ни те, ни другие, однако, не придавали должного значения изучению самой структуры знаний, особенно знаний процедурального типа.

Можно понять, почему психологи чувствуют себя не очень уверенно, оперируя сложными схемами, не основанными на тщательно выверенных механизмах мышления. Однако стремление к ограничению их числа еще не соответствует данному этапу развития науки в той мере, в какой это может иметь место в будущем. Анатомия и генетика мозга являются той областью знаний, в которой можно предположить значительно большее число разнообразных механизмов, чем это можно себе представить. Нам следует сосредоточить свое внимание скорее на **проблемах достаточности и эффективности, чем на проблеме необходимости**.

Еще сравнительно недавно главная цель работ по распознаванию образов сводилась к проблеме достаточности: **найти любые пути, ведущие к разработке алгоритмов машинного анализа сцен**. Наконец специалистам удалось **обнаружить и реализовать возможности правильного объединения отдельных черт и**

признаков истин и информации в законченные структуры образов. Следует ометить, прежде всего, работы Л.Робертса, А.Гузмана, П.Уинстона, Д.Хаффмана, М.Клоувза, Дж.Сираи, Д.Вальтца, которые характеризуют собой ряд этапов в разработке вопросов анализа изображений типа "фигура-фон", "целое-часть" и выделение структурных групп. Хотя эти работы достаточно просты, на их основе можно дать не только поверхностное толкование феномена зрительного восприятия, но и в какой-то степени объяснить быстроту и гладкость его протекания. Теория восприятия образов истин сталкивается с рядом новых вопросов при переходе от проблемы достаточности к проблеме эффективности. Каким образом различные виды "признаков" могут столь быстро, как это имеет место в человеческой практике, приводить к идентификации и описанию сложных ситуаций? Каковы способы внесения изменений при выявлении ошибок или нахождении новых доказательств? Как разрешаются противоречия? Как может быть изменена информация о местоположении объекта без перевычисления состояний других связанных с ним предметов? Как обстоит дело с движущимися объектами? Каким образом процессы зрительного восприятия используют знания, связанные с общими, невизуальными видами деятельности? Каким образом человек координирует информацию, поступающую из различных источников? Как в системе могут использоваться ожидания относительно результатов предполагаемых действий? Может ли теория объяснить феноменологические результаты зрительного восприятия образов, а также управляемое самим ходом восприятия построение и манипулирование воображаемыми сценами?

В рамках традиционных подходов бихейвористской и перцептуальной психологии было сделано очень мало, чтобы найти ответы на эти вопросы. В более поздних работах по теории символьной обработки информации, в публикациях А.Ньюэлла, Л.Пилишина и др., содержатся более конструктивные предложения по формулированию этих спорных вопросов.

1.4. Отслеживание образа куба

"...при обычном способе осмотра любого непрозрачного предмета видимая часть его поверхности обычно занимает все наше внимание, а о противоположной, даже точно такой же его части человек обычно в это время не думает; однако малейшая попытка

определить вид другой стороны предмета для построения общей картины уточняет наше первое представление о нем..."

У. Хогарт (1955)

Начнем изложение с разработки упрощенной **системы фреймов для представления** перспективных видов куба. Далее она будет модифицирована для представления внутреннего вида комнат и для приобретения, использования и обновления информации, необходимой человеку при перемещениях внутри дома.

В соответствии с использованным в работе А.Гузмана символическим представлением тел правильной формы с помощью "областей" и "связей" между ними можно допустить, что результатом восприятия внешнего вида куба является структура, подобная той, что показана на рис.1.1. Подструктуры А и В представляют детали и обозначения двух граней куба. При перемещении вправо грань А исчезает из поля зрения, тогда как видимой становится новая грань С. Если теперь, находясь по отношению к кубу в ином месте, попытаться заново провести весь анализ сцены, то придется забыть о том, что было известно об А; затем заново найти информацию о В и описать грань С.

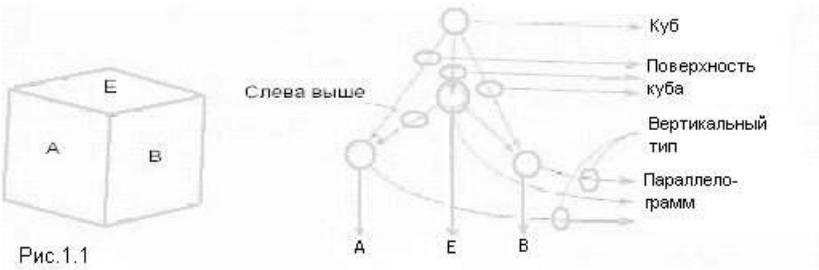


Рис.1.1

Но поскольку мы переместились вправо, то можем сохранить информацию о В, связав ее с терминалом "левой грани" второго фрейма куба. Чтобы сохранить (на всякий случай!) сведения об А, мы связываем его с дополнительным терминалом невидимой грани в новой схеме куба, показанной на рис.1.2.

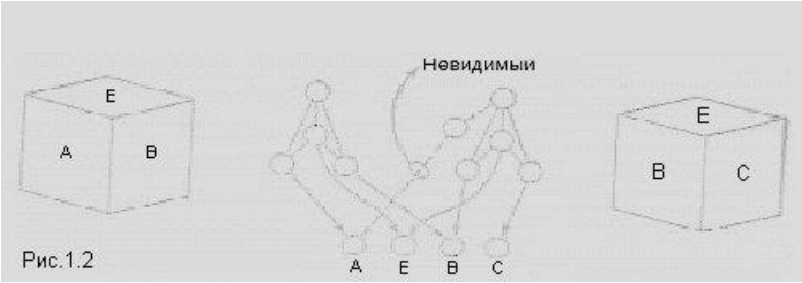


Рис.1.2

Если же потом переместиться обратно влево, то можно восстановить первоначальную картину без перцептивных вычислений, для этого потребуется только лишь восстановить связи верхнего уровня с первым фреймом куба. Теперь нам необходима информация о грани С. Для этого понадобится добавить еще одну невидимую грань справа в первом фрейме куба (рис.1.3).

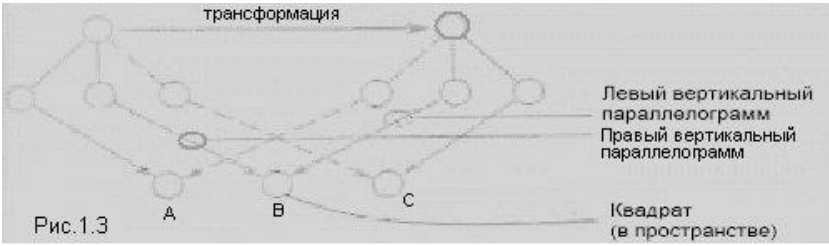


Рис.1.3

Можно продолжить эту процедуру, чтобы подобным образом представить результаты осмотра предмета с других сторон. Это приведет к более полной системе фреймов, в которой каждый фрейм представляет собой различные "перспективные виды" куба. На рис.1.4. показаны три фрейма, соответствующие перемещению влево и вправо на угол в 45 градусов.

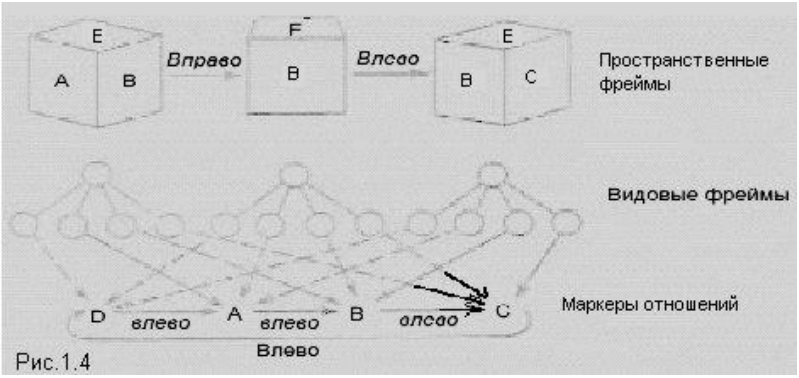


Рис.1.4

Если продолжить этот анализ, то результирующая система может стать очень большой. Для более сложных объектов требуется большее число различных проекций. Не ясно, все ли они необходимы в обычных условиях или же требуется только одна проекция для каждой их разновидности. Это зависит от обстоятельств.

Подобный тип сложной структуры, конечно же, не создается заново каждый раз, когда человек осматривает какой-либо предмет. Видимо, в долговременной памяти хранится большой набор систем фреймов, и одна из них активируется, когда данные и ожидания дают возможность предполагать, что она соответствует видимой картине. Как же это происходит? Если выбранный фрейм подходит не в той мере, как хотелось бы, и не удается быстро найти более подходящий, а вопрос достаточно важен, то происходит приспособление наилучшего из обнаруженных фреймов к реальной картине и он запоминается для последующих применений.

Строятся ли такие системы фреймов для каждого знакомого нам объекта? Это выглядело бы слишком расточительно (неэффективно). Представляется более вероятным, что у человека имеются специальные системы для представления наиболее важных объектов, а, кроме того, множество фреймов для обычно используемых "основных форм"; их сочетания образуют фреймы для новых применений.

Различные фреймы системы похожи на многочисленные "модели" объектов, описанные А.Гузманом, П.Уинстоном и др. Различные

фреймы аналогичны различным видам изображений, а имена межфреймовских указателей соответствуют перемещениям или действиям, изменяющим местоположение наблюдателя. Ниже будет обсуждаться вопрос о том, следует ли рассматривать эти виды в качестве двух- или трехмерных моделей объектов.

В каждый фрейм включены терминалы, служащие для присоединения указателей, идущих к его субструктурам. Одна и та же физическая черта может присутствовать в различных видах изображения объекта, следовательно, соответствующий ей терминал будет совместно использоваться сразу несколькими фреймами. Это позволяет представлять в одном месте информацию, собираемую не только в разное время и в разных местах, но и не зависящую от позиции наблюдателя. Это важно и для невизуальных применений систем фреймов.

Ход процесса согласования, результатом выполнения которого является решение, соответствует ли реальной ситуации выбранный фрейм или нет, зависит как от текущих целей, так и от информации, связанной с этим фреймом. Фреймы содержат в себе маркеры терминалов и другие ограничения, а цели используются для принятия решений о том, какие из этих ограничений существенны в данный момент, а какие нет. Вообще говоря, процесс согласования может содержать следующие компоненты:

1. Проверка на адекватность. Выбранный на основе ожиданий или предварительных данных фрейм должен вначале пройти проверку на правильность сделанного выбора; при этом используются знания о ранее выделенных элементах, их местоположении, об отношениях и наиболее вероятных субфреймах. Перечень текущих целей используется для принятия решения о том, какие терминалы и какие условия следует учитывать при составлении фрейма с действительностью.

2. Конкретизация. Затем фреймом запрашивается информация, необходимая для конкретизации значений тех терминалов, которые более не могут сохранять свои заранее заготовленные значения. Например, может потребоваться описание грани С, если соответствующий терминал в данный момент времени не означен и не отмечен как "невидимый". Задания должны согласовывать с текущими условиями, задаваемыми маркерами соответствующего терминала.

Так, грань С может содержать маркеры для таких ограничений или ожиданий как:
правая центральная область изображения;
должно быть означено;
должно быть видимым; если нет, рассмотреть перемещение вправо;
должно быть субфреймом "грань куба";
использует совместно с гранью В терминал левой вертикальной границы;
при неудаче рассмотреть фрейм "ящик, лежащий на боку";
тот же цвет фона, что и у грани В.

3. Управление. При получении сведений о трансформации (например, о предстоящем перемещении) выбранный фрейм передает управление соответствующему фрейму той же системы.

При более подробном рассмотрении этой схемы управления видно, что в ней содержатся возможности для использования многих видов знаний. Если попытка задания значений терминалам не удается, то результирующее сообщение об ошибке может быть использовано для выбора альтернативного варианта. Пользуясь этим, ниже рассмотрим вариант организации памяти в виде сети подобия, как это сделано в работе П.Уинстона.

1.5. О зрительном восприятие символьной формы

Можно ли действительно предполагать, что восприятие человеком трехмерных объектов столь фрагментарно и атомистично, что оно может быть представлено в терминах отношений между элементами двухмерных изображений? Отделим друг от друга эти два вопроса: являются ли зрительные образы символьными и основаны ли они на двухмерных конструкциях. Первый вопрос имеет особое значение; не вызывает сомнения, что на некотором уровне зрительное восприятие протекает в основном в символьной форме. Разногласия могут быть между теми, кто, с одной стороны, считает восприятие либо картиноподобным, либо протекающим на основе оперирования с воображаемыми геометрическим телами, и теми, кто, с другой стороны, основываясь на экспериментальных данных (Ж.Пиаже, Б.Инельдер, и др.), доказывает, что многие возможные ограничения, вытекающие из символьных представлений, и в самом деле существуют. Так, известно, что дети в своем творчестве, особенно в графике (впрочем, это относится и к взрослым) пользуются набором

весьма ограниченных символьных ингредиентов. Перспективы и заслонения обычно представляются не такими, какие они есть на самом деле, а с помощью определенных условностей. Метрические соотношения сильно искажены; сложные формы изображаются с помощью специальных знаков, которые не пользуются для представления наиболее существенных признаков. Представители "наивной" точки зрения обычно не признают подобных ухищрений и придерживаются мнения, что люди действительно "видят и манипулируют образами, подобными картинам" так, что это нельзя объяснить с помощью дискретных описаний.

Что касается второго вопроса (являются ли образы двух- или трехмерными), то его на уровне символьный описаний не существует, ибо неподходящим становится здесь само понятие измерения. Каждый вид символьного описания объекта одним целям служит хорошо, а другим - плохо. Если заданы отношения "слева-от", "справа-от" и "находится-над" между элементами некоторой структуры и представлены они в виде маркеров, определенных на парах терминалов, то при проведении определенных манипуляций с объектом его описание, выполненное на этой основе, будет достаточным для предсказания местонахождения его отдельных элементов. Задача облегчается тем, что если, например, поворачивать куб, не меняя его ориентации в пространстве (не меняя грани, которой он соприкасается со столом), то определенные свойства этих отношений будут инвариантными к подобным перемещениям. Большинство предметов обычно имеет свои верхние и нижние части. Однако если положить куб на боковую грань, то предсказания, основанные на тех же самых описаниях, сделать будет значительно труднее: люди испытывают большие затруднения при слежении за гранями шестицветного куба (т.е. куба, каждая грань которого окрашена в разный цвет), если заставить их мысленно его поворачивать. Если для тех же целей использовать более "характерные" отношения, такие как "следующий-за" или "быть-противоположным-к", то аналогичные описания изображений будут менее чувствительными к возможным поворотам объектов. В своих работах П. Уинстон описывает, каким образом систематические замены отношений (например, "слева" вместо "сзади" или "справа" вместо "спереди") могут использоваться при имитации вращения предметов.

У. Хогарт осуждал тех художников, которые слишком мало времени уделяли совершенствованию своих представлений об окружавших их

предметах. (Уильям Хогарт (1697-1764), выдающийся английский живописец, график и теоретик искусства, в 1753 г. опубликовал, свои известный теоретический трактат "Анализ красоты"). Он советовал тем, кто стремится получить правильные представления о расстояниях, отношениях и различиях между некоторыми существенными точками и линиями, принадлежащими, в худшем случае, даже наиболее асимметричным фигурам, постепенно вырабатывать в себе способность извлекать их из своей памяти, ибо это может во многом помочь тому, кто постоянно что-нибудь изобретает или рисует по памяти и способствует точному натурному воспроизведению предметов.

Таким образом, преднамеренная тренировка памяти в вопросах систематизации отношений между точками, лежащими на противоположных поверхностях тел, является, по мнению У. Хогарта, ключом к пониманию инвариантных отношений между видимыми и невидимыми частями изображений; они могут дать человеку информацию, достаточную для того, чтобы вообразить себя внутри какого-то предмета или мысленно очутиться в другой, практически недоступной точке наблюдения. Отсюда можно сделать вывод о том, что У.Хогарт отвергал "наивные" концепции в теории восприятия образов.

Некоторые люди полагают, что пространственные задачи решаются с помощью каким-то образом хранимого в памяти аналога трехмерной структуры. Если, однако, кто-либо и смог бы воссоздать такую модель, то для "интеллектуального глаза" сохранилась бы большая часть из тех традиционных проблем, которые относятся к реальному глазу, и, кроме того, появилась бы новая весьма трудная задача: создание (на основе двухмерных конструкций) образа какого-то гипотетически воображаемого предмета.

Хотя эти аргументы, как может показаться, свидетельствуют о целесообразности употребления двухмерных изображений для агрегирования и распознавания образов, их нельзя считать удовлетворительными для задач планирования и выполнения манипуляцпонкых операций. **Более естественным выглядит другой вариант представления информации в той же символьной форме, но на базе основных геометрических форм.** Так, телефонная трубка может быть описана с помощью двух усеченных сферических тел, соединенных изогнутым прямоугольным стержнем. В следующем

параграфе будет рассмотрен вопрос о совместном использовании двух и более методов, качественно отличных друг от друга, для представления одного и того же объекта.

1.6. Видение комнаты с помощью (посредством) органов зрения

Познание истин (окружающего мира) с помощью органов зрения кажется нам непрерывным. Одной из причин этого является наше постоянное движение. Более глубокое объяснение заключается в том, что обычно наши ожидания "гладко" взаимодействуют с нашим зрительным восприятием. Предположим, что вам пришлось выйти из комнаты, закрыть за собой дверь, затем вернуться, чтобы ее открыть и обнаружить совершенно другую комнату. Вы были бы поражены. Смысл этой перемены был бы едва ли менее поразительным, чем внезапное, на ваших глазах, изменение всего мира.

Теория феноменологической непрерывности утверждает, что скорость зрительного восприятия настолько велика, что наши образы могут изменяться с такой же скоростью, как и видимые нами сцены. Существует альтернативная теория, предложенная Минским: изменения в основанных на фреймах представлениях человека происходят со своей собственной скоростью; система фреймов предпочитает производить небольшие изменения, как только это становится возможным, а иллюзия непрерывности возникает вследствие постоянства заданий терминалов, общих для фреймов различных видов. Таким образом, непрерывность зависит от подтверждения ожиданий, что, в свою очередь, зависит от быстроты доступа к запомненным знаниям об окружающем нас мире.

Перед тем, как войти в комнату, вы уже заранее знаете, что увидите комнату, а не какой-нибудь пейзаж. Обычно можно указать на это по типу двери, так же как и выбрать заранее фрейм, соответствующий виду новой комнаты. Часто люди просто предполагают наличие какой-то конкретной комнаты. В этом случае значения многих заданий терминалов уже определены. Самый простой вариант фрейма комнаты - это подобие пустой внутренности коробки. Следуя рассмотренной ранее модели куба, можно считать, что структура верхнего уровня фрейма "комната" должна соответствовать схеме, показанной на рис.1.5.

Рис.1.5

Человек должен конкретизировать задания терминалов теми предметами, которые он видит. Если комната ему хорошо знакома, то некоторые задания уже заполнены данными (означены). В случае, когда отсутствуют какие-либо ожидания, на первой месте должно быть стремление выявить наиболее характерные геометрические признаки. Чтобы заполнить задания терминала "левая стена", можно вначале попытаться найти линии a и d, а затем углы ag и dg. Линию g найти в обычных условиях легко, поскольку она будет пересекаться при любом горизонтальном (на уровне глаз) осмотре помещения, проводимом слева направо. В конечном итоге углы ag, gb и ba должны соответствовать друг другу, так как все они являются частями одной и той же физической вершины.

Поскольку, однако, сам **процесс восприятия информации является направлением**, существуют основанные на знании и опыте эффективные тактические схемы. Вероятно, границу e найти легче, чем любую другую, ибо, как только мы входим в обычную прямоугольную комнату, можно ожидать, что *граница e является горизонтальной линией; она расположена ниже уровня глаз; она разделяет между собой пол и стену.* Если известны предполагаемые размеры комнаты, мы можем определить величину e и наоборот. В сценах открытых пространств линия e является горизонтом и на равнинной местности мы можем ожидать ее появления даже на уровне глаз. Если нам не удастся быстро найти этот горизонт и заполнить им соответствующее задание терминала, то следует рассмотреть отклонения, связанные с этим терминалом: возможно, что комната имеет какую-то необычную форму или, например, в ней находится крупный предмет - препятствие для визуального отыскания линии e.

Попытаемся обнаружить некоторые другие характерные признаки. Найдя линию е, следует заняться поиском ее левого и правого углов, а после этого вертикальных линий, исходящих из них. Как только будут обнаружены эти основные ориентиры, можно представить себе общую форму и размеры комнаты. Это может привести к выбору нового фрейма, который лучше согласуется с найденными формой и размерами объекта наблюдения благодаря маркерам, подтверждающим сделанный выбор и уточняющим структуру объекта с помощью дополнительных деталей.

Конечно, совершенная система зрительного восприятия должна анализировать сцену не просто как отдельную картину, а в соответствии с некоторыми установками фрейма более общего вида. Чтобы сам процесс восприятия протекал без затруднений, человек должен знать, где во внешнем и постоянно-меняющемся мире находится каждая нужная ему деталь. Это позволяет компенсировать трансформации в системах фреймов от перемещений глаз и головы, от изменений в положении туловища так же, как и от его более значительных перемещений с одного места в другое.

1.7. Анализ сцен и субфреймы

Если новая комната знакома недостаточно хорошо, то ни один заранее сформированный фрейм не может содержать сведения обо всех мелких деталях; в подобных случаях требуется проводить более глубокий анализ сцен. Однако объем работы и здесь может быть во многом уменьшен благодаря наличию таких **субфреймов, с помощью которых можно выдвигать гипотезы о структуре и связях реальных объектов (истин).** Насколько эти субфреймы-ожидания могут быть полезны, зависит как от их **адекватности рассматриваемому предмету, так и от качества процесса согласования, который устанавливает очередность сопоставления субфреймов с действительностью.** Они многое могут сказать даже о малознакомой комнате. Большинство комнат подобны коробкам, и их следует классифицировать по типам: кухня, зал, жилая комната, аудитория и т.д. Человек знает десятки разновидностей комнат, и ему известны сотни их конкретных видов; нет сомнения в том, что **они организованы во что-то типа сети подобия для того, чтобы обеспечить быстрый доступ к этой информации**.

Типичный фрейм комнаты включает в себя терминалы трех или четырех видимых стен, каждая из которых может относиться к различным типам, например к стенам с окнами, с полками, картинами и камином. У каждой разновидности комнат свои типичные стены. Представление обычной стены может содержать массив терминалов размером (3×3): (левая часть - центр - правая часть) ×(верхняя часть - средняя часть - нижняя часть), так что для предметов, относимых к стенам, можно качественно задавать их местоположение. Это может использоваться для локализации объектов с помощью внутренних пространственных отношений, например, чтобы представить такой факт, как "Y находится немного выше центра прямой, которая связывает X с Z". Если в трехмерном пространстве известно только направление восприятия (иначе, оптическая ось), то положение какого-либо элемента изображения, задаваемое соответствующим субфреймом, будет неопределенным. Линия, находящаяся в центральной части изображения, может принадлежать предмету, расположенному на передней стене комнаты, а может относиться к высокому объекту, находящемуся перед этой стеной; сами же объекты, очевидно, связаны с разными субфреймами. Решение об интерпретации линий может зависеть от обоснованных доказательств в пользу того или иного варианта, от более точной **визуальной информации**, полученной по данным стереовосприятия или двигательного параллакса, а также от той правдоподобной информации, которая может быть получена от других фреймов.

Положение границ пространственных элементов фрейма не носит совершенно четкого характера, и поэтому терминал каждого такого элемента должен содержать данные о (приблизительном) типичном местоположении его центра и некотором диапазоне относительных размеров. Мы предполагаем наличие лишь **правильных топологических ограничений**, например, что край левой стены должен всегда быть слева от любого стоящего у этой стены объекта. Процесс согласования видимой стены со всеми подобными ограничениями может приводить ко все большим затруднениям по мере того, как в описание элемента будут включаться (вопреки установленным для него размерам) предметы, предположительно находящиеся внутри него. Степень таких затруднений зависит от цели человека при анализе сцены и накопленного им опыта. Хотя данная концепция и может показаться сложной, богатство и разнообразие зрительных ощущений не должны наводить на мысль о создании каких-либо значительно более простых теорий.

1.8. Перспективы и перемена точек наблюдений

Ж. Пиаже, Б. Инельдер пишут: "Умственные способности, необходимые для координации перспективных изображений, оказываются полностью сформированными у ребенка обычно к 8-9 годам и проявляются в следующих довольно независимых друг от друга формах. Во-первых, каждому положению наблюдателя соответствует определенная совокупность отношений между видимыми предметами, например, отношения "слева - справа" или "спереди - сзади"... Они зависят от проекций и сечений, соответствующих плоскости изображения для данного наблюдателя (т. е. от перспективы). Во время этого последнего подэтапа шаг за шагом выявляется характер соответствия между точкой наблюдения и перспективой. Во-вторых, между каждой перспективой, связанной с данным положением наблюдателя, и любыми другими перспективами также существует соответствие, выражаемое особыми изменениями в отношениях типа "слева-справа", "спереди-сзади" и, следовательно, изменениями соответствующих проекций и сечений. Это соответствие между всеми возможными точками наблюдений обусловливает координацию перспектив... хотя еще только в элементарной форме".

Когда мы двигаемся по комнате, очертания находящихся в ней предметов изменяются. Каким образом можно предвидеть или компенсировать эти изменения без полного повторного анализа всей сцены? Эффект от движения глаз и поворота головы довольно прост; предметы перемещаются в рамках видимой области пространства, но не меняют при этом своих очертаний; однако изменение позиции наблюдения является причиной значительных перемен, которые зависят как от угла, так и от относительных расстояний между предметом и наблюдателем. Эта проблема особенно важна для животных, двигающихся с большой скоростью, так как у них модель внешней среды должна быть образована различными, частично проанализированными видами изображений. Видимо, эта потребность, пусть даже в самом примитивном своём варианте, послужила главным стимулом к эволюционному развитию систем фреймов, а позже и других символьных механизмов.

Если имеется обычная комната, то перемещение вдоль пунктирной линии (рис.1.6) вызывает упорядоченное изменение в очертаниях четырехугольных стен.

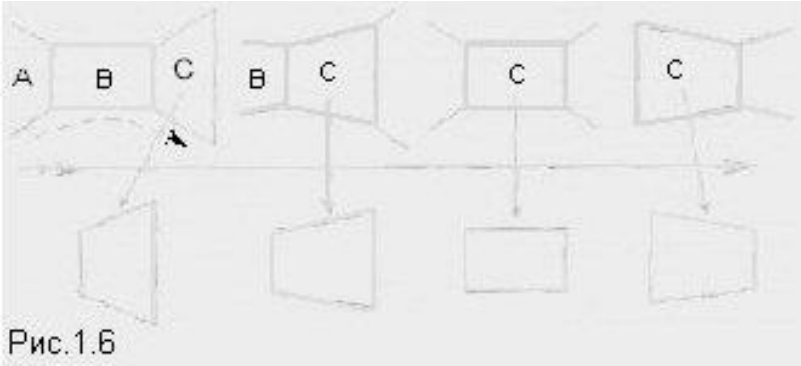


Рис.1.6

Прямоугольник, находящийся в непосредственной близости от стены, должен трансформироваться точно так же, как и сама стена. Если на левой стене в центре вычертить прямоугольник, то будет казаться, что он находится перед стеной, поскольку человек предполагает, что любой такой четырехугольник есть на самом деле прямоугольник и, следовательно, он должен лежать в плоскости, проектируемой аналогичным образом. На рис.1.7а оба прямоугольника, казалось бы, выглядят одинаково, однако тот прямоугольник, что находится справа, не согласуется с маркерами терминала для субфрейма "левый прямоугольник" (которые, например, требуют, чтобы левая сторона была длиннее правой).

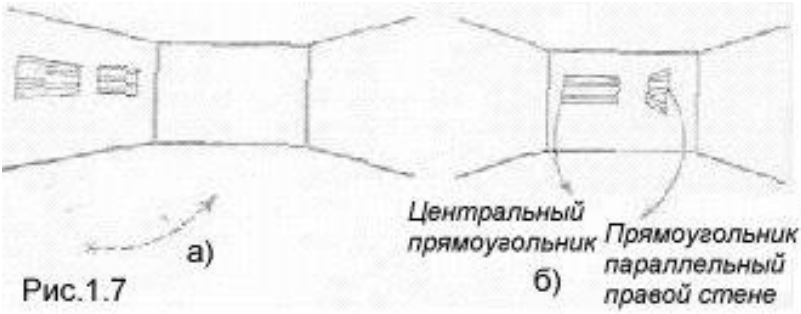


Рис.1.7

Таким образом, этот прямоугольник, представленный с помощью фрейма "центральный прямоугольник", кажется нам выдвинутым вперед и параллельным центральной стене. Итак, необходимо не просто присвоить четырехугольнику метку "прямоугольный", но и сделать то же самое для определенного фрейма, входящего в систему

фреймов "прямоугольники". Двигаясь по стрелке (рис.1.7а), можно ожидать, что любая трансформация, применимая к системе верхнего уровня, будет применима и к любой из ее подсистем (рис.1.7б). Аналогично последовательность эллиптических проекций окружности на плоскость изображения содержит конгруэнтные и потому визуально неоднозначные пары, что и показано на рис.1.8.

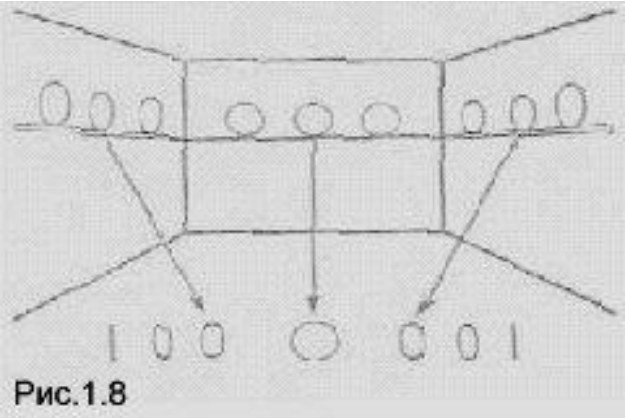


Рис.1.8

Но, поскольку предметы обычно располагаются в плоскости стен, мы предполагаем, что эллипс левой стены будет находиться слева на плоскости изображения, и что он подвержен тем же трансформациям, что и сама стена. Если предсказание не подтвердится, мы, очевидно, будем весьма удивлены.

Правдоподобно ли, что ограниченная, качественная, знаковая система может служить инструментом для адекватного представления перспективных преобразований. Люди все время недооценивают свои возможности по восприятию образов, например, в таком вопросе, как переосмысливание пространственных отношений, при изменении точки наблюдения. Уже отмечалось, что люди, считающие себя обладателями хорошего пространственного видения, часто допускают качественные ошибки при описании вращения простого многоцветного куба. И даже, в тех случаях, когда мы в действительности способны высказать точные метрические суждения об объектах видимой сцены, мы не всегда делаем это; например, лишь немногих людей заставит задуматься изображение несуществующей реально пирамиды, показанной на рис.1.9.

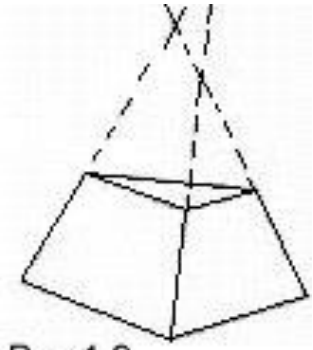


Рис.1.9

Это не пространственное изображение (перспектива) любой усеченной пирамиды. Если эта фигура была бы пирамидой, то все три ее ребра в своем продолжении сошлись бы в одной точке. Имея достаточный опыт, люди, без сомнения, могут высказывать более точные суждения, но для этого не требуются какие-либо другие механизмы. Если для выполнения некоторой работы непрофессионал использует десять фреймов, то специалист может применить тысячу и, таким образом, получить совсем иной порядок ее исполнения. В любом случае для правильного предсказания изменений перспективы в наших системах необходимо, чтобы трансформация на верхнем уровне вызывала соответствующие трансформации в системах субфреймов. В первом приближении этого можно добиться с помощью одинаковых названий трансформаций. Тогда действия "двигаться вправо", заданное для фрейма комнаты, будет вызывать то же самое действие и для объектов, связанных с субфреймами стен этой комнаты. Поскольку, однако, эта схема имеет серьезный дефект, то она и рассматривается здесь только лишь в качестве "первого приближения". Действительно, если вы стоите около левой стены и двигаетесь вдоль нее вперед, то предметы, расположенные вблизи этой стены, испытывают значительную трансформацию типа "двигаться-вправо", фронтальная стена подвергнется преобразованию типа "приблизиться", а правая стена испытывает трансформацию типа "двигаться влево". Таким образом, вопрос о правильном предсказании изменений перспективы оказывается далеко не таким легким, чтобы его можно было решить простым переносом типов операций более низкие уровни системы.

1.9. Проблемы заслонений

Когда мы двигаемся вправо, крупный предмет, находящийся в центре на переднем плане, вероятно, левой своей стороной будет заслонять любой более удаленный предмет. При планировании перемещений человек должен иметь возможность предвидеть некоторые из этих изменений. Часть предметов может стать невидимой, а другая их часть - видимой. В нашем исходном примере куба проблемы заслонения не существует, поскольку это тело является полностью выпуклым; исчезновение целой грани и всех ее связей может быть легко восстановлено по данным, содержащимся на верхнем уровне. Однако в комнате, которую обычно следует рассматривать как тело вогнутой формы, отдельные элементы объектов, относящиеся к различным терминалам, могут заслонять друг друга. Рассмотрим в этой связи две экстремальные стратегии.

Локальные группы предметов. Так же, как и с различными видами одиночных объектов, при рассмотрении знакомых групп частично загораживающих друг друга предметов можно воспользоваться специальной системой фреймов, соответствующей данной конфигурации объектов исходного изображения. Для примера рассмотрим сцену, состоящую из стола и стула (рис.1.10 и табл.1.1).

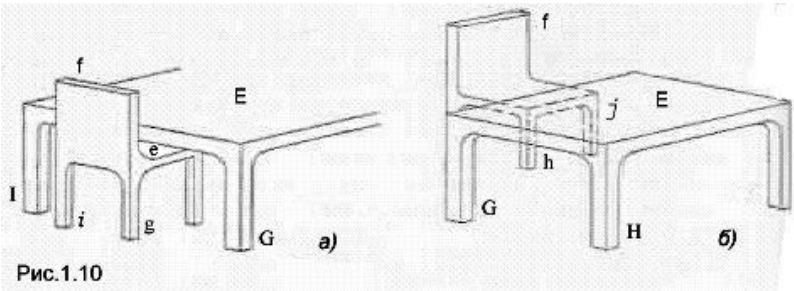


Рис.1.10

Таблица 1.1

Характер заслонения	Положение элементов относительно друг друга

Рис.1.10а

f частично заслоняет e	Под, сзади
e заслоняет j	Под, сзади
f частично заслоняет E	Слева, под
E заслоняет J	Под, сзади
E заслоняет h	Под, частично сзади
E заслоняет j	Под, сзади

Рис.1.10б

e заслоняет i	Под, сзади
E заслоняет I	Под, сзади
E частично заслоняет e	Слева, под
E заслоняет j	Под, частично сзади
E частично заслоняет h	Под, слева

Если к фрейму, представляющему эту сцену применить те же, что и ранее, трансформации перспектив, то это позволит в первом приближении решить проблему заслонения.

Такой подход хорош для компактных, заранее известных групп объектов, но он не дает нужных результатов в тех случаях, когда одни предметы загораживают от дельные детали других объектов, расположенных в углублениях комнаты. Для технических применений простота данной схемы не компенсирует частые ошибки, возникающие при ее использовании, однако, как теория человеческих действий она может быть достаточно хорошей.

Отметим, что умственная деятельность, направленная на выяснение заслонений, протекает медленно и ее не следует стремиться объяснить

на основании упрощенных представлений, связанных главным образом со скоростью выполнения операций.

Глобальная система заслонений. Более глубокая стратегия должна предусматривать, чтобы все фреймы пространственных изображений (перспектив) дополняющая единую, центральную систему фреймов зрительного восприятия образов. Терминалы такой системы должны сответствовать отдельным элементам внутреннего представления пространства среды, а трансформации - определять выявленные законы перспективы, согласно которым одни элементы по-разному заслоняют другие для различных точек наблюдения.

Если на самом деле у человека существовала бы такая макросистема, то была бы она врожденной или же сформировалась в результате обучения? Приведенная выше цитата из работы Ж.Пиаже и Б.Инельдер служит подтверждением тому, что дети до десятилетнего возраста не обладают подобными макроструктурами.

1.10. Образы и системы фреймов

"Каждый человек охотно согласится с тем, что между двумя видами ощущений - непосредственными, когда человек, например, страдает от сильной жары или ему приятно разливающееся по всему телу тепло, и ощущениями, воскрешаемыми в памяти или предвкушаемыми благодаря своему воображению, имеется значительное различие. Эти наши способности могут имитировать восприятия органов чувств, но с их помощью невозможно достигнуть силы и живости первоначального чувства... Самая яркая мысль все. же слабее самого притупленного ощущения".

Д. Хьюм (1963)

Теория зрительного восприятия должна быть, видимо, одновременно и теорией воображения, поскольку и та, и другая имеют одни и те же конечные цели: **конкретизацию заданий терминалов фреймов**. Каждый из нас согласится с Д.Хьюмом в том, что **имеется различие между зрительным восприятием и ощущениями**. Д.Хьюм утверждает, что это различие проистекает вследствие того, что **восприятие по своей природе является мгновенным и непосредственным,** тогда как вызываемые в памяти образы получаются рекомбинацией заполненных "впечатлений", а при

хранении, поиске и преобразованиях этих структур каким-то образом теряется их выразительность. **Зрительное восприятие кажется более живым, нежели воображение, ибо его задания менее подвижны**. Они более стойко сопротивляются попыткам других процессов изменить их. Для того чтобы изменить описание непосредственно видимой сцены, наша система зрительного восприятия, вероятно, будет вынуждена изменить эту сцену - ни больше и ни меньше. Для человеческой фантазии, конечно же, подобных жестких ограничений не существует. В то же время сами "зрительные" задания не являются абсолютно негибкими; любой может научиться в уме изменять, представленную на чертеже изометрию куба. Существуют так называемые "двусмысленные" фигуры, которые можно легко описать различным образом. В этом случае изменение фрейма равнозначно изменению "описательной точки зрения", причем действие или преобразование носит символьный, а не физический характер. В любом случае существуют такие психические состояния, при которых фантазии менее гибки, чем "непосредственные восприятия", а иногда даже и более "ярки".

1.11. Априорное означивание

Как мы уже упоминали, **целью работы механизмов восприятия и воображения является конкретизация заданий терминалов фреймов**, при этом воображение оставляет человеку больший простор для выбора деталей и различных вариантов этих заданий. По-видимому, фреймы никогда не хранятся в долговременной памяти с незаданными значениями своих терминалов. Каждый терминал фрейма в действительности непрочно связан со своими заданиями отсутствия, которые чаще всего бывают полезны, но иногда могут препятствовать процессу поиска нужного фрейма. Так, если вам скажут: "Джон ударил ногой по мячу", то, видимо, вы не думаете о каком-то чисто абстрактном мяче, а представите себе вполне определенные его характеристики: размер, цвет, массу, которые, однако, пока еще неизвестны. Возможно, что возникший в вашей памяти образ чем-то напоминает вам самый первый мяч или тот, которым вас больно ушибли, или, может быть, тот самый последний мяч, который вы держали в своих руках. В любом случае вашему воображению недостает остроты (эффекта "присутствия"), поскольку те процессы, которые контролируют ход согласования и оказывают влияние на непрочно связанные с терминалами задания отсутствия, только лишь

подтверждают или отвергают их пригодность и не имеют связи с реальной действительностью.

Задания отсутствия должны оказывать тонкие, идеосинкразическне воздействия на те мыслительные операции, с помощью которых человек проводит аналогии, делает обобщения и вырабатывает суждения, особенно когда внешние влияния на них достаточно слабы. При правильном выборе эти стереотипы могут служить в качестве хранилища ценных набросков планов, в противном случае они могут образовывать наборы иррациональных данных парализирующего действия.

1.12. Системы фреймов и конкретные мыслительные операции Пиаже

"Каковы в действительности условия, необходимые для формирования формального мышления? Ребенок должен в уме не только оперировать с предметами, иными словами, мысленно выполнять возможные над ним действия, но он должен также "отражать" в себе эти операции при отсутствии предметов, когда последние заменяются на словесные формулировки. Это "отражение" есть мысль, достигшая второй стадии (стадии формальных операций). Конкретное мышление является представлением о возможном действии, а формальное мышление - представлением представления о нем.
Поэтому нет ничего удивительного в том, что система конкретных операцнй должна полностью сформироваться в течение последних лет детства (к 11-12 годам) еще до того, как она получит свое представление с помощью формальных операций. По своим функциям формальные операции не отличаются от конкретных за исключением того, что они применяются к гипотезам или утверждениям абстрактной системы "вывода", которая управляет конкретными операциями".

Ж. Пиаже (1971)

Можно считать, что имеется сходство между утверждениями Пиаже о конкретных операциях и идеей использования **трансформаций между фреймами системы**. Некоторые виды логических операций могут быть легко реализованы на базе фреймов путём замены в них одних заданий отсутствия другими. Например, не вызовет затруднений

попытка аппроксимации логических транзитивностей; так, силлогизмы вида: "Все, что принадлежит А, принадлежит В, и все, что принадлежит В, принадлежит С, =>, все, что принадлежит А, принадлежит С", должны естественным образом встречаться при подстановках субфреймов в терминалы фрейма. Видимо, это общее правило не универсально, но можно считать, что вследствие транзитивности сменяющих друг друга субфреймов в некоторых случаях им будет полезно воспользоваться. В дополнение к этому можно предполагать, что к справедливости высказываний вида: Большинство, относящееся к А, относится к В, и большинство, относящееся к В, относится к С, => большинство, относящееся к А, относится к С", следует относиться с той же степенью доверия, даже если иногда это не соответствует действительности.

Ясное понимание вопроса о том, что же может быть достигнуто на базе простейших операций с фреймами, является весьма ценным и актуальным. Значительным шагом в этом направлении явилась разработанная Дж.Муром и А.Ньюэллом процедура "выравнивания и покрытия" для их программы MERLIN. Эта процедура связана с известным в математической логике алгоритмом унификации, если известны фреймы А и В, то результаты ее работы могут быть проинтерпретированы (если не слишком вдаваться в детали) следующим образом: можно рассматривать А как вид В, если имеется "отображение" или фрейм-трансформация С, указывающая, каким образом (может быть даже с помощью каких иных отображений") терминалы, относящиеся к А, могут рассматриваться в понятиях В-терминалов.

В упомянутой выше работе метод изменения точек зрения используется, чтобы предложить ряд новых интерпретаций таким основным стратегиям, как целенаправленность, индукция и использование новых знаний. Кроме того, авторы высказывают предложения, каким образом можно в машинных программах реализовать основную идею теории фреймов.

По теории Ж.Пиаже способности детей к проведению рассуждений, как с помощью трансформаций, так и о них самих проявляются с переходом мышления ребенка на стадию **формальных операций**. Для различных видов умственной деятельности эти способности могут проявляться не одновременно и не синхронно по отношению друг к другу. Чтобы проводить более сложные рассуждения и освободиться

от полезной, но малонадежной логики манипулирования с заранее заготовленными значениями, человек должен научиться оперировать с самими трансформациями, поскольку подобные преобразования содержат в себе сведения, необходимые, для более сложных форм умственной деятельности. В создаваемых для систем представления истин и информации моделях можно было бы попытаться заставить ее (систему) читать свои собственные программы. Возможная альтернатива заключается в том, чтобы представить (с избыточностью) информацию о процессах иным способом. Сотрудники лаборатории искусственного интеллекта Массачусетского технологического института, разрабатывающие программу, "понимающую программы", обычно приходили к мнению, что в них должны содержаться "комментарии" для более четкого выражения намерений, предпосылок и целей эти комментарии (в настоящее время) обычно записываются на специализированных языках. В этой связи возникает важный вопрос о цели разработки Минским теории фреймов. Мышление на базе "схем", в основе которого лежит согласование сложных ситуаций со стереотипными структурами фреймов, явно недостаточно для некоторых видов умственной деятельности. Очевидно, что взрослые люди, думая о чем-либо, используют для своих целей ранее сформировавшиеся собственные представления. Если представить себе "формальные" операции в виде процессов, которые могут изучать и критиковать наши ранее сформировавшиеся представления (в виде фреймов или любом другом), то с их помощью можно создавать новые структуры, которые будут соответствовать "представлениям о представлениях".

Та же стратегия предполагает, что непосредственное использование фреймов мы связываем (схематично, по крайней мере) с "конкретными" операциями Ж.Пиаже. Если это так, то следует согласиться с тезисом Ж.Пиаже о том, что позднее появление "формального" мышления у детей связано с парадоксальным повторным убеждением. При рассмотрении примеров применения системы фреймов к различным проблемам могут возникнуть сомнения по поводу того, что данная теория хорошо объясняет одно и плохо другое. Однако не следует ожидать, что в рамках любой отдельно взятой системы можно решить все проблемы человеческого мышления, тем более что эта система ограничена конкретными операциями, сводящимися к манипуляциям со стереотипными структурами данных.

2. Язык, понимание и сценарии

2.1. Слова, предложения и смысл

Концепция фрейма и использование заданий отсутствия, по-видимому, полезны при рассмотрении проблемы понимания смысла. Н.Хомский указывал, что такое предложение, как:

(А) Бесцветные зеленые идеи спят неистово трактуется совсем иначе, чем высказывание:
(В) Неистово спят идеи, зеленые, бесцветные.

Это происходит, во-первых, потому, что оба высказывания "одинаково бессмысленны", и, во-вторых, потому что процессы, связанные с анализом предложений, должны во многом отличаться от процессов, связанных с пониманием смысла.

Нет сомнения в существовании особых механизмов связанных с грамматическим разбором предложений. Поскольку смысл высказывания в равной мере "закодирован" как в позиционных и структурных отношениях между словами, так и в выборе самих слов, то должны существовать и механизмы, связанные с анализом этих отношений и участвующие в формировании структур, которые призваны более четко представлять смысл этого высказывания. Почему при выработке такой структуры слова в высказывании (А) производят большее впечатление и смысл их более понятен, чем в высказывании (В), если не касаться вопроса о том, следует ли называть эту структуру семантической или синтаксической? Потому что порядок слов в (А) и отношения между ними заданы с учетом (грамматических) условностей и правил, которыми люди обычно пользуются, чтобы побудить других конкретизировать определенными значениями задания терминалов своих понятийных структур. Это полностью согласуется с грамматическими теориями. Порождающая грамматика могла бы служить в качестве совокупного описания внешнего проявления этих правил (или связанных с ними процессов), а операции в трансформационных грамматиках достаточно близки к

трансформациям фреймов. Следует, однако, уточнить, насколько самостоятельно используется грамматика при работе человеческой мысли. Вероятно, неприятие высказываний (вследствие конструкций, не соответствующих правилам грамматики, или просто непонятных) указывает на более сложный характер причин, обусловливающих семантический отказ выработать какое-либо представление. Ниже приводятся доводы в пользу того, что противопоставление грамматики и смысла может осветить две крайности из этого круга вопросов, но в то же время завуалировать все остальные его важные проблемы.

Нельзя считать, что любой "логической" бессмысленности неизбежно сопутствует и бессмысленность психологическая. В самом деле, высказывание (А) может породить определенный психический образ! Центральным в этом образе, видимо, является фрейм сна, для которого система задает определенную кровать с "лежащим" на ней фреймом, представляющим мягкую, бесформенную полупрозрачную массу зеленого цвета. Во фрейме сна имеется терминал для задания характера сна (сейчас, видимо, беспокойного), а слово "неистово" кажется здесь несколько неподходящим, так как этот фрейм предпочитает не признавать в своем действии чего-либо намеренно спланированного или заранее предусмотренного. Еще больше смущает слово "идеи", поскольку в качестве субъекта мыслится какое-нибудь лицо или, по крайней мере, что-то живое. В рамках рассматриваемой структуры фрейма разрешить эти противоречия не удается.

Сказать что-либо подобное о высказывании (В) попросту нельзя, поскольку в нем нет ни одного существенного фрагмента, который можно было бы сопоставить с одним из возможных субфреймов.

Поэтому ни один более высокий по иерархии фрейм не получит каких-либо данных для согласования своих терминалов и, следовательно, ни один из фреймов верхнего уровня типа "смысл" или типа "предложение" не в состоянии сообщить о том, имеет ли высказывание (В) правильную грамматическую форму и (или) заложен ли в нем какой-нибудь смысл. Видимо, сочетание этой "гибкой" теории с градацией отбора данных для заполнения заданий отсутствия может быть положено в основу разработки такой системы, которая скорее будет несостоятельной для предложений с "плохой" грамматикой, нежели для предложений, где ее попросту нет. Если более мелкие фрагменты, т.е. фразы и части предложений достаточно хорошо удовлетворяют субфреймам, то даже, несмотря на неполную

согласованность данных на верхних уровнях могут быть построены образы, приемлемые для определенных видов понимания. Таким образом, мы получили качественную теорию грамматического разбора: *если данные удовлетворяют верхним уровням и не удовлетворяют некоторым терминалам более низких уровней, то на входе - бессмысленное предложение; если же картина обратная, то высказывание может иметь смысл, но оно облечено в грамматически неправильную форму.*

Восприятие предложений не обязательно должно сопровождаться *зрительными* образами. Некоторые люди не пытаются представить себе цвет мяча в предложении "он ударил по мячу". Но, в конечном итоге, все согласятся, что в рамках вызванного из памяти сценария используются, если не такие характеристики, как цвет или размер, то цели, отношения и другие существенные элементы. Вне рамок зрительного восприятия терминалы и их задания отсутствия могут представлять цели и функции, а не только цвета, размеры и формы.

2.2. Рассуждение

Лингвистическая деятельность человека требует от него использования образований более крупных, чем те, которые могут быть описаны с помощью грамматических правил, а это в еще большей степени затрудняет понимание вопросов, связанных с разграничением синтаксиса и семантики. Рассмотрим следующую басню (У.Чейф): *"Жили-были волк и ягненок. Однажды увидел волк, что ягненок пьет воду из реки, и появилось у него желание съесть ягненка. Решил он найти себе хоть какое-то оправдание и, несмотря на то, что сам находился выше по течению, обвинил ягненка в том, что тот взбалтывает воду и не дает ему пить…"* Чтобы понять этот отрывок, надо ясно представить себе, что волк лжет! Чтобы понять ключевую фразу "несмотря на то, что...", надо знать, что жидкость не может двигаться вверх по течению, а это, в свою очередь, требует от нас понимания самих слов "вверх по течению". В рамках декларативной, основанной на исчислении предикатов логической системы фразу "выше по течению" можно задать в виде некоторой формулы, например:

[A находится выше по течению, чем B] $\vee$ [Событие T. BA поток мутный] => [Существует [Событие U. BB поток мутный]] $\wedge$ [U позже T].

Однако более полное определение было бы гораздо сложнее. Например, как записать тот факт, что потоки воды, перемещая какие-либо предметы, обычно не изменяют их расположения относительно друг друга? Логик мог бы попытаться доказать его, исходя из достаточно сложной совокупности "локальных" аксиом и соответствующих правил индуктивного вывода. Представим эти знания с помощью особой структуры данных, которая автоматически переориентирует связи пространственных описаний с терминалов одного фрейма на терминалы другого в рамках одной и той же системы фреймов. И хотя это может рассматриваться как некоторый вид логики, здесь используются определенные механизмы такие же, как и для пространственного мышления. Во многих случаях нам приходится иметь дело с изменяющимися во времени ситуациями или причинно следственными отношениями. Так, концепции "течение реки" может соответствовать система фреймов, аналогичная той, что показана на рис.2.1, где S_1, S_2 и S_3 обозначают отдельные абстрактные участки реки.

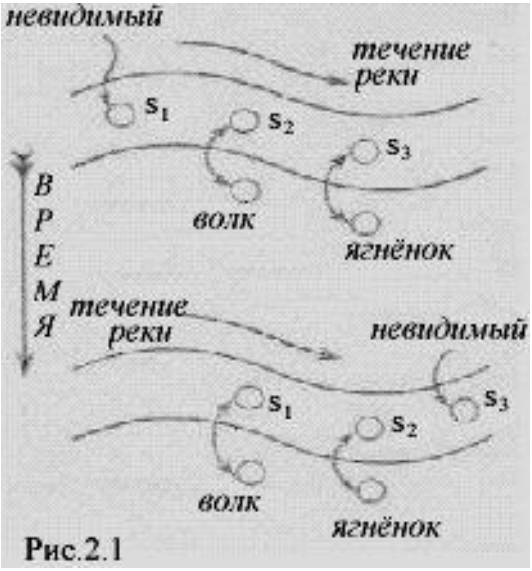


Рис.2.1

Если волк в соответствии с ранее рассмотренной схемой находится слева, а ягненок от него - справа, то S_1, S_2 и S_3 "двигаются" мимо них

также слева направо. Наше воображение именно так представляет себе эту картину, а для ее изменения требуются некоторые умственные усилия. Допустим, что участки реки S_1, S_2 или S_3 видны только тогда, когда они располагаются прямо против волка или ягненка. Если последний взбалтывал воду на участке S_3, то система фреймов укажет на то, что, поскольку этот участок перестанет быть видимым и не будет находиться вблизи волка, его претензии к ягненку безосновательны. Более подробная система могла бы иметь промежуточные фреймы, но ни для одного из них участок реки возле волка не был бы загрязнен. Существует еще много нюансов, в которых следует разобраться. Что означает слово "взбалтывал" и почему это мешало волку пить? В обычных условиях с помощью элементов S могут быть представлены просто какие-то плавающие предметы, однако здесь S_3 взаимодействует со словом "взбалтывал" и в результате получается нечто противоречащее действию "пить". Или такой вопрос: было ли доказано, что взбалтывание воды в реке обусловливает присутствие грязи в воде на участке S_3, или же это указывается в одном из заранее заготовленных значений данного действия?

Почти любое событие, действие, изменение, перемещение объектов или даже передача информации в первом приближении могут быть представлены обобщенным событием в виде системы, состоящей из двух фреймов. Система фреймов может иметь пробелы для действующих лиц (агентов), инструментов, побочных явлений, предварительных условий, обобщенных траекторий, но, в отличие от переходных глаголов в теориях формальных грамматик, здесь имеется дополнительная возможность в явном виде представлять возможные изменения. Чтобы проверить, понял ли кто-либо то или иное событие или действие, можно построить соответствующую ему **пару фреймов типа "до-после"**. Однако использование подобных пар фреймов для представления ситуационных изменений совсем небезошибочно, поскольку ссылки на эти пары фреймов не адекватны описанию различий между ними. Это вызывает неудобства при таких видах деятельности, как планирование или проведение абстрактных рассуждений, поскольку не существует явных точек включения в подобные структуры информации о трансформациях. В развитие этого варианта можно ввести пары узлов, указывающие на соответствующие терминалы: в этом случае мы получили бы структуры, подобные "записям сходства" П.Уинстона. Кроме того, можно поместить на верхний уровень системы фреймов данные, несущие в себе

информацию о различиях между фреймами в парах "до-после", выраженных в более общем виде.

В своей работе по теории "семантической зависимости" Р.Шенк пытается найти смысловое представление для сложных утверждений, таких, например, как *"Сэм верит в то, что Джон дурак"*. В этих предложениях предмет, на который направлено действие, не может быть представлен как реальный физический объект. Это, однако, не снимает вопроса о "концептуализации" этого предмета и представлении его с помощью ситуаций подобно тому, как это выглядит при разборе следующего диалога:

- *Не хотите ли Вы кусочек шоколада?*
- *Нет, я только что съел трубочку с мороженым.*

Р.Шенк считает достаточным наличие ограниченного набора "основных концептуализации" и типов связей между ними для того, чтобы с их помощью представлять смысл сложных высказываний. Трудно сказать, насколько правилен этот подход, например, можно ли с его помощью описать такое явление, как "поток". Теория Р.Шенка включает в себя идею "концептуальных атрибутов", которые напоминают некоторые из терминалов фреймов, однако в ней сделана попытка **представить результаты действий не с помощью отношений между парами фреймов, а в виде явных абстракций**. Это порождает свои проблемы; например, было бы интересно узнать, достаточно ли одной (или даже нескольких) абстрактной концепции причинам для работы системы "убеждений" или нет. Недостаточной выглядит попытка представить причинную связь с помощью некоторого условия или действия, необходимого для того, чтобы произошло какое-то событие. Не вдаваясь более в детали, видимо, что как только в рамках теории Р.Шенка будут разработаны некоторые проверки выполнимости действий, она станет мощным средством для представления знаний.

Работа Дж.Уилкса по "семантике предпочтений" тоже, видимо богата идеями относительно того, как создавать **структуры, подобные фреймам, из более простых элементов**. Его предложения в отношении предпочтений объединяют в себе особые пути, при помощи которых можно было бы представлять задания отсутствия и процедуры, согласованные с более крупными фрагментами рассуждений. Система Дж.Уилкса интересна еще и тем, что она

отчетливо показывает способы, с помощью которых можно проводить некоторые полезные неформальные рассуждения, а также раскрывает механизм псевдодедукции, основанный на шаблонном принципе построения всей системы и на текущих процессах без привлечения тщательно разработанной формальной логической системы или без излишней заботы о непротиворечивости в рассуждениях.

Р.Абельсон работал над проблемами представления еще более сложных видов деятельности. Основываясь на элементах, подобных элементам Р.Шенка, он разрабатывал схемы, в которых взаимодействие различных понятий ведет к образованию сложных "предписаний" (scripts), т. е. ажурных или решетчатых сценариев в тщательно разработанных системах "убеждений". С их помощью он пытлся отразить даже такие взаимодействия, как представление одного человека о той роли, которую он играет в планах другого действующего лица.

В своей работе Д.Макдермотт рассмотрел многие вопросы, связанные с представлением знаний. В его схеме для выработки правдоподобных заключений любые утверждения не просто принимаются на веру, но подлежат проверке, осуществляемой процессами, которые выражают "сомнения" и "убеждения"; по существу, данные, наличие которых предполагается по умолчанию (или поскольку они правдоподобны), сохраняются благодаря механизмам изменения убеждений, действующих, когда последние, зависимые предположения уже включены в работу. Д.Макдермотт особенно выделяет те вопросы, которые связаны с устранением ошибок, возникающих в любой системе в процессе поиска неформальных и правдоподобных заключений.

2.3. Смысловая структура рассуждений

Ф.Бартлетт отмечает: "Слова... могут отражать качественные и относительные черты некоторой ситуации в их наиболее общем виде, как, впрочем, и описывать, вероятно, даже более удовлетворительно, ее особенности. Именно это обусловливает внутреннюю связь языка с процессами мышления. В психологическом плане восстановление любой имевшей место в прошлом ситуации выполняется для преодоления трудностей на базе прошлого опыта... Не следует, однако, думать, что человек, попавший в затруднительное положение, воссоздает более или менее подходящую ситуацию (и на ее основе

находит для себя решение) без какого-либо поиска и выявления принципов перехода от одной ситуации к другой".

Анализ предложений с помощью "глубинных падежей", разрабатываемых, например, С.Филмором и М.Селс-Мурсиа, проводится с помощью структур, чем-то напоминающих фреймы. Сгруппированные главным образом вокруг глагола части предложения используются для того, чтобы заполнить конкретными данными пробелы в подобной фрейму структуре этого глагола с учетом различных вариантов возможного включения предлогов в это предложение. Для анализа предложений имеет смысл использовать именно те структуры, доминирующее положение, в которых принадлежит глаголам, поскольку именно это и происходит на практике. Однако при более обширном анализе, выходящем за рамки отдельного предложения, такие структуры часто теряют свою самостоятельность и включаются в более крупные фрагменты рассуждений. Темой какого-либо отрывка из повествования в равной мере могут быть картина или действие, характеристика персонажа или то, что он делает. Таким образом, при понимании рассуждения синтез глагольной структуры с конкретным заданием ее падежей может быть необходимым, но лишь преходящим этапом. По мере понимания предложения отдельные субструктуры следует объединять в расширяющих фреймах-картинах для того, чтобы построить на их основе еще более крупную картину. Действие, основное в некотором предложении, может, например, стать вспомогательным для характеристики одного из героев всего рассказа.

Не будем придерживаться концепции, утверждающей, скажем, что глаголы описывают локальные (в рамках отдельных предложений) структуры, а существительные - структуры глобальные, состоящие из отдельных параграфов или пунктов, хотя в первом приближении она могла бы оказаться полезной. Любая концепция может быть построена с помощью различных лингвистических представлений. Но обсуждаемая нами проблема не сводится к проблеме существительных или глаголов. Важно понимать, что преходящие семантические структуры, построенные во время синтаксического анализа (т.е. те структуры, которые лингвисты именуют "глубинными структурами" предложений), не идентичны более крупным (и более "глубоким") структурам, образующимся по мере того, как к ним подключаются связанные друг с другом лингвистические объекты.

Не хотелось бы, чтобы подобный акцент на имеющие лишь местное значение (или предметные) суперфреймы приводил к мысли о том, что между лингвистическими и нелингвистическими представлениями имеются коренные отличия. При более детальном рассмотрении этого вопроса станет, по-видимому, ясно, что существенная часть понимания и проведения рассуждений на основе здравого смысла во многом напоминает преобразования лингвистических структур и манипуляции ими. **Фреймы, связанные со смыслом слов, будь то существительные, глаголы или другие части речи, несомненно, являются центрами концентрированного представления знаний о том, как связаны между собой различные предметы и явления, каким образом они используются и как они друг с другом взаимодействуют.** Можно получить значительные преимущества в том случае, если бы удалось создать механизмы, в которых эти одинаковые структуры применялись бы как для реализации процессов мышления, так и для понимания естественного языка.

Представим себе ориентированный на фреймы сценарий, который представляет собой структуру для понимания смысла повествований. В начале чтения какого-нибудь рассказа о нем известно очень мало, фактически только то, что это рассказ, но даже и этих скудных сведений на первых порах оказывается достаточно. **Обычный фрейм рассказа в общем случае включает в себя пробелы, которые надо заполнить сведениями об окружающей обстановке, главных героях, основном событии, морали и т. д.** И в самом деле, в любом правильно построенном повествовании вначале идет речь о действующих лицах и той среде, в которой будет развиваться действие; так, в басне о волке и ягненке сразу же говорится о двух непримиримых субъектах, находящихся у реки (это и есть окружающая обстановка), а затем указаны мотивы для соответствующего поведения волка. Слова "найти хоть какое-то оправдание" подготавливают нас к тому, что волк, видимо, сделает какие-то ложные утверждения.

Анализ каждого предложения следует проводить только до тех пор, пока содержащиеся в нем сведения могут быть полезны для заполнения более крупных структур. Таким образом, в терминалах расширяющейся смысловой структуры накапливаются указатели и дескрипторы, которые сообщают о дальнейших путях конкретизации заданий отсутствия. Терминал, с которым связан маркер "лицо женского рода", не сможет использовать такие задания, где присутствуют местоимения мужского рода. Видимо, именно этим

следует объяснить невозможность конкретизации терминалов фрейма стены с помощью заданий типа "стул" или "стол". По мере того, как продолжается рассказ, информация (когда это возможно) передается фреймам более высоких уровней для углубления и конкретизации сценария. В некоторых случаях к терминалу фрейма удается присоединить целый субфрейм, например описание одного из героев рассказа. Это может случиться тогда, когда терминал фрейма-рассказа согласуется с указателем верхнего уровня фрейма анализируемого предложения. Данные, содержащиеся в других предложениях, могут вместе с тем приводить и к образованию противоречивых ситуаций. Но что произойдет, если не удастся осуществить передачу данных на более высокий уровень, поскольку собеседник ожидал услышать рассказ другого" типа и не располагает терминалами, с помощью которых можно было бы усвоить новую порцию данных?

Будем продолжать считать, что у нашего собеседника в действительности имеется большое количество фреймов-рассказов, связанных между собой структурами поиска информации, которые будут рассмотрены позднее. Тогда **первым шагом явится попытка включить новую информацию в текущий фрейм-рассказ**. Если это не удается, следует выработать сообщение об ошибке, например: "здесь нет места для животного", что заставит заменить используемый фрейм другим, скажем, фреймом-рассказом о животных. Все предыдущие задания терминалов могут сохраниться, если у нового фрейма типы терминалов будут теми же самыми. Если же многие задания не подходят, следует попытаться применить другой, новый фрейм-рассказ. Если и здесь мы терпим неудачу, то есть два пути: либо начать все с начала и пытаться всю структуру строить заново (видимо, это самая важная интеллектуальная задача), либо полностью отказаться от дальнейших попыток и забыть все то, что было сделано для понимания рассказа. Это обычная реакция на неизвестные ранее формы повествования! Человек не может хорошо учиться, если разрывы между известным для него и неизвестным слишком велики: **рассказы о животных не будут фактически понятны до тех пор, пока не сформированы персональные фреймы для волка, лисы, медведя и т. д.**

Таким образом, при проведении рассуждений многочисленные **фреймы и субфреймы объединяются в сети поиска информации**. **Атрибутивная или описательная информация может быть представлена с помощью простых структур, однако действия, временные последовательности, разъяснения и другие сложные**

вещи требуют более отработанных механизмов представления знаний. Следует признать, что все основные, очень трудные вопросы, относящиеся и к эпистемологии, и к лингвистике, концентрируются вокруг одной проблемы: **как отыскать те способы, с помощью которых можно будет согласовывать сведения, поступающие от различных источников информации.**

2.4. Перевод

Перевод дает возможность наблюдать механизм конкретизации заданий отсутствия в работе. При переводе басни о волке и ягненке с английского языка на японский требуется упомянуть о том месте у реки, где находятся действующие лица, хотя этого не требуется в английском варианте. В английском языке необходимо указывать время, хотя бы сказав "однажды...". В японском языке принято характеризовать как место действия (хотя бы с помощью расплывчатой фраз "в некотором месте..."), так и время его свершения. Видимо, что сведения и о месте действия, и о его времени нужны людям для их более глубоких смысловых фреймов независимо от того, на каком языке они говорят. Следовательно, как только переводчик полностью поймет какое-нибудь предложение, задания отсутствия обеих типов сразу же будут в его распоряжении. Хорошие переводчики-синхронисты говорят так быстро, что можно лишь удивляться тому, как много могут они понять еще до того, как начнут говорить. Теория фреймов доказывает, что это не такое уж большое достижение, ибо если еще до того момента, как процесс согласования *полностью* завершит свою работу, будет найден подходящий фрейм, то все его задания отсутствия становятся доступными и могут быть включены в работу. Перевод басни "Волк и ягненок" на японский язык, выполненный в соответствии с вполне приемлемой и поверхностной структурой, может выглядеть следующим образом:

Однажды в некотором месте на реке ягненок, пьющий воду, увидел волка, и этот волк этого ягненка съел под предлогом...

В японском языке более естественно сказать о том, что именно пил ягненок, чем-то, что он пил. В этом заключается один из способов, с помощью которых язык воздействует на процесс мышления: **каждое особое лингвистическое правило концентрирует внимание на заполнении терминалов определенного типа.** Если для какого-то человека привычным и связанным с действием пить является понятие

"вода", то оно является заранее заготовленным значением для структуры, представляющей это действие. Заполнение этим понятием задания отсутствия будет вполне допустимым, если, например, в разговоре встретится фрейм действия "пить", участвующий в формировании структуры выходного предложения (фразы). Конечно, если утоление жажды происходит в непосредственной близости от реки, следует позаботиться о наличии каких-то дополнительных механизмов связи понятий "пить" и "река" с помощью понятия "вода". Кажется ясным, что если в каком-либо фрейме имеется пробел для представления одного из видов жидкости, утоляющей жажду, и этот пробел слабо связан со всей структурой, то он может быть легко вытеснен данным того же типа, сильно связанным с субфреймом, который подключается к исходному фрейму.

2.5. Активная и пассивная формы интеллектуальной деятельности

Традиционно *видение* и *воображение* рассматриваются как "пассивная" и "активная" формы восприятия. Действительно, **в процессе зрительного восприятия (видения) человек анализирует картину с помощью выявленных в процессе сбора информации данных**; это ограничивает его возможности уже потому, что преобладающим здесь выступает не поиск непротиворечивого и правдоподобного описания наблюдаемой сцены, а стремление согласовать "объективную" визуальную информацию с выбранным для целей представления фреймом. *Воображение* рисует нам значительно более разнообразнее картины, поскольку в этом случае задачей является подбор таких заданий терминалов, которые удовлетворяли бы нашим внутренним и, следовательно, изменяемым целям.

В языковых формах общения напрашивается аналогичное противопоставление. Если человек *слушает* своего собеседника, что предполагает проведение грамматического разбора, то для понимания текста ему требуется выстраивать отдельные слова в конструкции, отвечающие правилам грамматики и соответствующие как общему содержанию разговора, так и намерениям говорящего; это, конечно, резко уменьшает возможное число вариантов конкретизации заданий терминалов. Значительно большую свободу выбора предоставляет нам *речь*. Существует огромное число различных способов образования из отдельных фреймов предложений. Для выражения поставленных перед

собой целей, будь то желание сообщить какую-либо информацию, убедить или ввести кого-то в заблуждение.

Подобные разграничения, однако, весьма опасны. Во многих случаях речь представляет собой прямой перевод данных, представленных в семантических структурах, в последовательность слов, тогда как слушание требует выполнения сложных и обширных построений для решения, в частности, проблемы *понимания.* Проведем аналогию между визуальным фреймом комнаты и фреймом, представляющим собой группу существительного в предложении при проведении рассуждении. И в том и в другом случае одна часть терминальных заданий является обязательной, а другая - нет. Например, стены в комнате совсем не должны быть чем-либо украшены, но каждый подвижный предмет не может не иметь соответствующих типов опор. Группа существительного может не содержать определителя числа, но не может обойтись без самого существительного или заменяющего его местоимения. Поверхностная структура оставляет человеку весьма ограниченный выбор вариантов, поскольку необходимым является в одном случае учет всех слов, включенных в предложения, а в другом - всех ключевых признаков анализируемых сцен.

При изучении языковых форм общения так же, как и процессов зрительного восприятия, совсем не следует ограничиваться рассмотрением только лишь поверхностных структур. У человека имеется множество вариантов для включения знаний в семантическую структуру или результатов действий в текущий контекст. **Любой предмет характеризуется не только формой, но и своей историей**. Присутствие его в данном месте и в данное время обычно не только чем-то обусловлено, но и несет в себе разную смысловую нагрузку, например, выступает как признак изменения отношений или как ключ к разгадке какой-то тайны. Любое предложение может быть по-разному понято. При этом речь идет не о случайных (и во многом несущественных) двусмысленностях грамматического разбора, а о преднамеренно разных интерпретациях. Также как любую комнату можно рассматривать из разных мест, так и любое утверждение может быть проанализировано с различных точек зрения по тем вопросам, которые в нем затрагиваются. Это видно из следующего примера, каждое предложение которого имеет свои структурные особенности:

Он ударил по мячу.
По мячу ударили.
Сегодня имели место некоторые удары.

Поскольку такие варианты формально напоминают синтаксические активно-пассивные операции в трансформационных грамматиках, человек может не заметить различия в их смысловых значениях. Мы выбираем нужный нам вариант в соответствии с *тематическими* вопросами, т.е. в зависимости от того, что нас интересует: то, что сделал "он", или то, куда мог исчезнуть мяч, или то, что кто-то его повредил, и т.д. Наиболее просто ответить на такие вопросы, если заострить внимание на их существе и вызвать фрейм, относящийся главным образом именно к *данной* теме. В соответствии с традиционными положениями теории трансформационных грамматик такие **альтернативные фреймы не могут существовать независимо друг от друга, ибо они являются производными от одной глубинной структуры.** Если, однако, допустить их не связанное друг с другом присутствие в долговременной памяти, можно получить определенные выгоды, подключив к каждому фрейму информацию о том, как его следует использовать. Впрочем, существуют (и на это справедливо указывают лингвисты) систематические "регулярности", которые свидетельствуют, что подобные трансформации могут практически с той же легкостью применяться и к незнакомым глаголам; все это уменьшает целесообразность отдельного существования таких альтернативных фреймов. Создается впечатление, что теоретики-трансформационисты склонны верить в наличие особых центральных механизмов для управления изменения "семантической перспективы", хотя ясно, что разнообразие особенностей, связанных с отдельными словами, делает это технически маловероятным. Теория, разработанная в соответствии с излагаемыми в данной работе взглядами, должна предусматривать, что всякий раз, когда человек встречается с неизвестным ему употреблением слова (или самим словом), он использует процесс согласования для того, чтобы догадаться, правильно оно или ошибочно, к чему оно ближе всего, а затем приспосабливает систему трансформации внимания к этому слову. Сложно сейчас указать, с помощью каких экспериментов можно было бы выявить различие между рассмотренными выше предположениями, но оно кажется важным.

Важно помнить следующее: на то, чтобы что-то понять, часто требуются минуты, часы или даже столетия.

2.6. Сценарии

"Мышление... в биологическом плане возникает вслед за развитием процессов формирования образов. Оно возможно только тогда, когда будет найден способ разрушить "массированное" влияние прежних стимулов и ситуаций, когда будет познан механизм подавления тирании прежних реакций. Но хотя мышление является более сложной формой умственной деятельности, оно не заменяет собой метода восприятия образов. Мышлению присущ ряд недостатков. По сравнению с воображением оно несколько теряет в живости и разнообразии. Основным инструментом мышления выступают слова, и не только потому, что они социальны, но и потому, что при употреблении они обязательно выстраиваются в цепочки и входит в привычные действия даже более легко, нежели образы. Благодаря мышлению мы подвергаемся риску оказаться все более и более втянутыми в поиск утверждений общего характера, имеющих мало общего с реальным и конкретным опытом. Если нам не удается придерживаться методов мышления, то реальна опасность ограничить себя частными примерами и стать игрушкой воли случая".

Ф.Бартлетт

В языке и мышлении мы кратко выражаем или условно представляем сложные ситуации и целые эпизоды с помощью слов и символов. Далеко не все слова, видимо, могут "определяться" с помощью простых и изящных структур; например, лишь весьма незначительный в своей полноте смысл понятия "торговля" может быть установлен исходя из следующих выражений:

Первый фрейм Второй фрейм

А имеет X, В имеет Y ---> В имеет X, А имеет Y

Торговые сделки производятся на основании определенных социальных законов, в обстановке доверия и обоюдного согласия. Если эти факты не получат должного отражения, то многочисленные торговые операции будут почти бессмысленными. Важно знать, что стороны, участвующие в сделке, обычно не против иметь обе обмениваемые вещи, но вынуждены пойти на компромисс. Счастливым, но необычным является такой вариант, когда каждый из

торговцев стремится избавиться от своего товара. Чтобы представить стратегию торговли, можно включить в вышеприведенный сценарий, который состоит из двух фреймов, основное правило торговли; для того, чтобы А смог заставить В пожелать приобрести больше X (или отдать больше Y), ему следует придерживаться одной из следующих тактик:

Предложить больше за Y.
Объяснить, почему X очень полезен для B.
Создать благоприятный для B побочный эффект от наличия у него X.
Демонстрировать свое пренебрежение конкурирующими с X товарами.
Заставить B думать, что C хочет получить X.

В этом, однако, заключается только лишь поверхностное рассмотрение проблемы. Торговля обычно производится в рамках сценария, действия которого связаны друг с другом значительно сложнее, нежели просто последовательной цепью событий. Одного из таких сценариев будет явно недостаточно; когда возникает мысль о торговле, важно знать, какой из имеющихся сценариев будет, по всей вероятности, наиболее полезен.

Е.Чарняк рассмотрел вопросы, связанные со сделками: сделки на первый взгляд достаточно просты для понимания, тем не менее, они могут быть представлены только с помощью структур, богатых, заданиями отсутствия. В учебниках для младших школьников можно найти такие рассказы, как, например:
Джейн была приглашена к Джеку на день рождения.
Она подумала, понравится ли ему воздушный змей.
Она пошла в свою комнату и потрясла копилку.
Из копилки не донеслось ни звука.

Большинство читателей понимает, что Джейн нужны деньги, чтобы купить Джеку в подарок воздушный змей, что в ее копилке нет для этого денег. Е.Чарняк предлагает ряд способов для облегчения вывода подобных фактов, например: использовать "демон" *подарок* с помощью которого может выполняться поиск вещей приобретаемых за деньги, или демон, связанный с копикой, который "знает", что если ее трясут и из нее не доносится никаких звуков, то она пуста, и т.д. **Демон** (Е.Чарняк) - **это утверждение процедурального типа, ассоциируемое с некоторым понятием семантической сети**. Например, демон

"дерево" может быть связан с понятием *весна* и отражать следующую причинно-следственную связь: "Когда *весна* полностью вступает в свои права, начинают цвести плодовые *деревья* ". Несмотря на то что теперь слово *подарок* активирует понятие *деньги,* сами они в рассказе не фигурируют и это может вызвать удивление. Подарок определенно связан с днем рождения, а деньги - с копилкой, но каким образом строятся эти длинные цепочки рассуждении? Это еще одна из проблем, затронутых в работе Е. Чарняка. Приятель говорит Джейн:

У него уже есть воздушный змей
Он заставит тебя отнести его обратно.

Какой воздушный змей надо будет отнести обратно?! Мы не хотели бы, чтобы Джейн вернула старый змей Джека. Чтобы разобраться в том, какое слово заменено здесь местоимением "его", нужно, в рамках предполагаемого сценария понять очень многие вещи. Ясно, что слово "его" относится к *новому* воздушному змею, который должна купить Джейн. Но каким образом мы об этом узнаем? (Заметьте, что совсем не обязательно требовать единственного объяснения). Обычно местоимения относятся к последнему из упомянутых ранее объектов, но, как показывает данный пример, это отношение не определяется просто лишь локальным синтаксисом. Предположим, что мы пытаемся заполнить конкретными заданиями пробелы в субфрейме "покупка подарка". Само по себе слово "его" недостаточно информативно, но группа слов "взять его обратно" вполне подошла бы для согласования соответствующего терминала сценария *покупка.* Поскольку задания этого терминала должны одновременно удовлетворять маркерам терминала "подарок", мы убеждаемся в правильности вхождения слова "его" (подарок) в словосочетание "отнести X обратно". Таким образом, тот факт, что отнести следует новый воздушный змей, устанавливается автоматически. Конечно, так же, как и у всех других, у искомого терминала имеются свои собственные ограничения. Субфрейму "покупка подарка" для идиомы "отнести его обратно" должно быть известно, что действие "отнести X обратно" предполагает следующее:

X куплен недавно.
Возврат производится по месту покупки.
У вас должен быть товарный чек и т.д.

Если текущий сценарий не содержит терминала "отнести его обратно", следует найти и использовать другой сценарий, сохранив по

возможности наибольшее числе прежних заданий. Отметим, что при нормальном выполнении процесса согласования вопрос о старом воздушное змее здесь попросту не возникнет. Ощущение двусмысленности появляется только тогда, когда рассматриваются отклонения, близкие к неправдоподобным.

Е. Чарняк предложил аналогичное решение; он, однако, подчеркивает важность понимания того факта, что поскольку у Джека уже есть воздушный змей, ему вряд ли нужен будет еще один. Е. Чарняк предлагает использовать следующие правила, связанные со словом "подарок":

(А) Если мы видим, что подарок X не нравится лицу P, то следует ожидать возврата X в магазин, где он был куплен.
(В) Если возврат X в магазин предполагается или является установленным фактом, то причина этого в том, что X не нравится P.

Автор этих правил считает, что подобные "советы" процедурального типа должны реализовываться как дополнение к сведениям декларативного характера, хранящимся в базах данных, и вызываться в ситуациях, соответствующих определенным контекстам. Если все предшествующие условия для таких "советов" выполнены, то включенные в них действия привнесут то количество информации (о Джеке, о воздушном змее), которое достаточно, например, для правильного разрешения вопроса с местоимением "его".

Е.Чарняк придерживается того мнения, что **система должна следить за определенными типами событий и ситуаций и использовать в своей работе сведения о возможных причинах, мотивах и объяснениях происходящего.** Он предполагает, что дополнительные взаимосвязи между фрагментами рассказа должны помочь процессу согласования в тех случаях, когда "правдоподобные" предположения заводят в тупик формальные логические процедуры. Допуская, например, что утверждение "X не нравится" является следствием того, что "X возвращён назад". Чарняк надеется таким путем получить возможность имитации понимания обычных или повседневных событий. Пока еще не ясно, насколько должны быть сложны и разнообразны механизмы выработки правдоподобных заключений для понимания действий заданного уровня общности, работа Е.Чарняка также не дает ответа на этот вопрос, поскольку в ней рассматриваются лишь относительно простые вопросы моделирования мыслительных

процессов. Е.Чарняк в большинстве случаев предлагает ограничиваться установлением причины или мотива поведения и не проводить более углубленный анализ, если в том нет необходимости. К примеру, если надо установить, почему Джек может возвратить второй воздушный змей, достаточно выяснить то, что он ему не нравится. В то же время более глубокий анализ мог бы показать, что Джек хотел бы получить в подарок разные вещи, поскольку их ценность в данном случае (когда предполагается только один подарок - от Джейн) значительно выше ценности одинаковых вещей.

2.7. Более сложные сценарии

Смысл празднования дня рождения ребенка весьма приближенно передается тем определением, которое можно найти, например, в толковом словаре: день рождения - это "прием гостей, устраиваемый по случаю дня рождения". Прием здесь может, в свою очередь, толковаться как "люди, собравшиеся по случаю праздника". В этом определении отсутствуют следы какой-либо деятельности, т. е. то, без чего невозможна сама жизнь. Дети знают, что такое определение должно включать в себя большее число различных конкретных данных; это может быть реализовано с помощью таких заданий отсутствия, как например:

Одежда	Воскресная, самая лучшая
Подарок	Должен понравиться хозяину, должен быть куплен и красиво упакован
Игры	Жмурки, салочки
Украшение	Воздушные шары, гирлянды цветов, банты, креповая бумага.
Угощение	Пирог, мороженое, лимонад, бутерброды с горячими сосисками.
Развлечения	Свечи, задуть свечи, высказывать пожелания, петь песни, посвященные дню рождения.
Мороженое	Обычное, ассорти

Эти компоненты типично американского дня рождения следует включать в более крупную структуру. Поскольку день рождения обычно празднуют в течение одного дня, его следует поместить во

фрейм соответствующего дня. Обычно день включает в себя такие основные события, как

Подъем, Одевание, Завтрак, Поездка на работу. Обед и т.д.

День школьника содержит более определенные данные:

Подъем, Одевание, Завтрак, Поездка в школу, Пребывание в школе. Класс, Сбор, Английский язык. Математика, Обед, Естественные науки, Перерыв, Физкультура. Возвращение домой. Игры. Ужин. Домашняя работа. Приготовление ко сну. Сон.

Очевидно, что день рождения не вписывается во фрейм дня школьника. Родители знают, что угощение во время дня рождения приурочивается к обеду.

Возвратимся к проблеме с воздушным змеем. Первая фраза, которую мы слышим, несет в себе информацию о том, что Джейн приглашена на день рождения к Джеку. Без разработанного сценария дня рождения или, по крайней мере, без сценария "пригласить в гости", второе предложение:
Она подумала, понравится ли ему воздушный змей, оказалось бы несколько загадочным. Чтобы попытаться объяснить, каким образом человек может быстро сообразить, в чем здесь дело, предлагается в состав фрейма, имеющего структуру сценария, явным образом включать указатели на те вопросы, которые обычно тесно связаны с этим фреймом.

Допустим, что терминалы фреймов представляют собой именно такие вопросы, и рассмотрим эту идею более подробно. Для приема гостей в день рождения лица X получится следующая схема:

Y должен приобрести P для X	*Выбрать P!*
P должно понравиться лицу X	*Будет X рад видеть P?*
Купить P	*Где купить P?*
Достать деньги для покупки P	*Где достать деньги?*
(Подвопросы для данного фрейма)	
Y должен быть одет	*Что следует надеть Y?*

Именно эти вопросы будут в первую очередь одолевать человека, который приглашен в гости.

Читателю предоставляется возможность подумать, насколько приемлемо сделанное выше предложение о включении указателей в состав фрейма. Действительно, вопрос "Будет ли X рад видеть P", вне всякого сомнения, согласуется с фразой "Она подумала, понравится ли ему воздушный змей" и правильно соотносит параметр P с его значением "воздушный змей". Вместе с тем важно понять, можно ли считать окружающий нас мир настолько систематизированным, чтобы, составив подобные группы вопросов, ожидать от предложенного механизма хорошей работы? Видимо, ответ здесь должен быть двояким. И в самом деле, с одной стороны, мы убеждены в существовании большого числа подобных вопросов, а с другой - не в силах предвидеть все возможные в такой ситуации варианты. Специальные проверки или "экспертизы" предназначены для того, чтобы получить представление о характере основных проблем и их взаимосвязях в малоизвестных нам ситуациях. Обратим внимание на то, что в нашем фрейме-сценарии "день рождения" *нет* заданий отсутствия для тех событий, которые могут происходить в какой-то определенный момент времени. Это акцентирует наше внимание на тематических вопросах и вопросах включения во фреймы соответствующих заданий терминалов. В любом случае, видимо, не обойтись без более действенного механизма для понимания слова "подумала", механизма, который сможет, располагая информацией, сосредоточенной в данном фрейме, выработать предположения о том, что же могла подумать Джейн.

Третье предложение рассказа ("Она пошла в свою комнату и потрясла копилку") также должно согласовываться с одним из вопросов фрейма. Здесь имеется указанная явным образом связь между понятиями "деньги" и "копилка", поскольку она, видимо, содержится во фрейме "копилка", а не во фрейме "день рождения". Будучи обнаруженной, она будет сопоставлена с вопросом: "Где достать деньги?"

Основная функция копилки состоит в накоплении денег с тем, чтобы их в нужный момент можно было взять обратно. Последнее действие может быть осуществлено тремя основными способами:

1. Использовать ключ (что в большинстве копилок не предусмотрено).
2. Разбить копилку (чего дети обычно не любят).

3. Вытрясти деньги или использовать тонкий предмет, по которому монеты могут выскользнуть из копилки.

Рассмотрим, наконец, четвертое предложение ("Из копилки не донеслось ни звука"). Знает ли человек о том, что та копилка, из которой при тряске не доносится звуков, пуста (видимо, это именно так), или же он пользуется общими сведениями, именно, что пустым является любой пустотелый ящик, не издающий звуков при тряске? Существует многое людей, которые предпочитают второй вариант. Использование этого логического "универсального" принципа было бы достаточным; однако, при этом может быть упущено из виду то важное обстоятельство, что конкретный сценарий такого характера крепко врезается в память каждого ребенка. Более того, наш рассказ сразу же становится понятным большинству читателей, чего нельзя было бы столь же категорично утверждать, если бы они использовали более сложные рассуждения, базирующиеся на общих принципах. Можно привести примеры, значительно более трудные, нежели предыдущий, например:

Коза забрела в огород, где что-то красил Джек, и опрокинула краску на себя. Когда мать увидела козу, она спросила: "Джек, это ты сделал?"

Здесь нет ни одного слова или фразы, к чему бы вносилось местоимение "это". Е.Чарняк отмечает, что оно, видимо, относится к "той причине, по которой и коза оказалась измазанной краской", и, не указывая, каким образом следует пытаться решать проблемы такого рода, отмечает лишь то, что для подобных слабоограниченных микромиров его концепция "демонов" должна быть значительно расширена. Обратите внимание, как много должен знать человек об окружающем его мире, чтобы понять, что "это" относится не к "козе во дворе", а к "козе, измазанной краской". В той же работе Е.Чарняка изучаются вопросы активации демонов в соответствии с имеющимися в них ожиданиями и знаниями в виде заранее заготовленных значений. Многие из его идей нашли отражение в настоящей работе.

Попытаемся провести параллель между тем, как Е.Чарняк трактует проблему понимания естественного языка, и зрительным восприятием образов. Существенные для проведения рассуждений тематические структуры или сценарии вызываются из памяти с помощью ключевых слов или ценных для этих рассуждении идей. Отдельные утверждения

способствуют возникновению временных представлений, которые, по всей вероятности соответствуют тому, что лингвистика именует "глубинными структурами", и которые могут быстро изменяться или совсем исчезать по мере уточнения и развития основной идеи в рамках выбранного сценария. Среди ингредиентов такой структуры можно выделить в порядке увеличения масштабности охвата событий следующие уровни:

1. *Поверхностные синтаксические фреймы* - главным образом, это структуры с глаголами и существительными. Для них необходимы соглашения о предлогах и порядке следования слов в предложении. Напомним, что английский язык отличается строгим порядком следования слов в предложении, отклонениями от которого подчёркивается его особая семантическая окраска.

2. *Поверхностные семантические фреймы* - группы слов, объединенные вокруг действий. Необходимы определители и отношения для действующих лиц, инструментов, траекторий, стратегий, целей, последствий и побочных явлений.

3. *Тематические фреймы - это* сценарии для видов деятельности, окружающих условий, изображений кого-либо или чего-либо, наиболее важных проблем, обычно связанных с данной темой.

4. *Повествовательные фреймы* - это скелетные формы для типичных рассказов, объяснений и аргументации. Необходимыми здесь являются соглашения о формах построения повествований, о развитии действий, о главных действующих лицах, основных событиях и т.д., призванные помочь слушателю строить в своем уме новые тематические фреймы и конкретизировать задания отсутствия. В том случае, когда задание не согласуется более в той степени, в какой это необходимо, отдельные предложения могут переозначивать задания терминалов, присоединять другие субфреймы, использовать трансформационные механизмы или вызывать замену фреймов верхнего уровня.

В *видении* трансформации между фреймами по своей природе просты, они заключаются в переходе от одной группы данных к другой; при изучении языковых форм общения следует предполагать потребность в более сложных и менее регулярных системах фреймов. Тем не менее, поскольку *время, причина* и *действие* чрезвычайно важны для понимания смысла, мы часто применяем связанные трансформации для

замены ситуаций их временными или причинными последовательными цепочками событий.

Изучение лингвистических структур, по всей видимости, поможет нам лучше понять, как построены наши системы фреймов, по той причине, что синтаксические структурные правила управляют отбором и компоновкой фреймов переходных предложений. С их помощью можно искать эти особым образом связанные с "задающими" терминалами структуры, для чего следует выбирать наиболее важные точки зрения (трансформации), вводить структуры предложений в тематические структуры и изменять крупные тематические представления в целом.

Наконец, по аналогии с известными "основными сюжетными линиями" для рассказов должны существовать основные суперфреймы и для проведения рассуждений, аргументации, изложения фактов и т.д. Следует ожидать, что удастся найти особые лингвистические указатели для операций, касающихся этих более крупных структур. Не следует ограничиваться изучением грамматики предложений, если мы хотим найти и систематизировать лингвистические соглашения, действующие, в более широких масштабах и связанные с компоновкой и трансформациями сценариев и планов.

2.8. Вопросы, системы и концептуальные случаи

Как отмечает А.Ньюэлл, "Вопросы являются следствием определенных точек зрения, они проистекают из чего-то такого, что помогает выяснять неясные моменты, формулировать ответы и уточнять проблематичные вещи. Не следует думать, что наши взгляды на окружающий мир предопределяют его; они выражают только то, что мы принимаем из действительности, и то, как мы ее создаем." Область лингвистики подводит нас к созданию такой концепции фрейма, в которой терминалы служат для хранения наиболее вероятных (в некоторой ситуации) вопросов. Разберем более детально эту интерпретацию фреймов.

Фрейм - это множество вопросов, которые необходимо задать относительно предполагаемой ситуации; на их основе происходит уточнение перечня тем, которые следует рассмотреть, и определяются методы, требуемые для этих целей. Терминалы фрейма соответствуют, видимо, "концептуальным случаям" Р.Шенка, хотя, их не следует

ограничивать столь малым числом типов, как это предлагает Р.Шенк. Чтобы понять действие, о котором идет речь или за которым человек имеет возможность наблюдать, ему часто приходится искать ответы на такие вопросы, как:

В чем причина этого действия (агент)?
Какова цель действия (намерение)?
Каковы последствия этого действия (побочные явления)?
На кого (или что) это действие влияет (получатель)?
С помощью каких средств оно выполнено (инструмент)?

Число таких вопросов или, по Р.Шенку, "случаев", которые полезны для понимания действий, проблематично. Хотя мы стремимся сокращать число "примитивов" до весьма малого их количества (по аналогии, видимо с традиционным лингвистическим анализом), однако, в этом случае нет оснований серьезно рассчитывать на успех. Видимо есть смысл встать на сторону таких исследователей, как У.Мартин, который исходит из наличия значительного числа "примитивов", снабженных комментариями относительно того, каким образом они связаны между собой. Только время может показать, какая из этих двух концепций лучше. Что касается сущностей, отличных от действий, то относительно них задают самые различные вопросы, тематически значительно менее локализованные, например:

Почему они говорят мне об этом?
Как я могу выяснить об этом больше данных?
Каким образом это поможет мне в реальной обстановке? и т. п.

В рамках любого рассказа обычно интересуются его темой, отношением к ней автора, главным событием, главными героями и т. д. По мере того, как на каждый подобный вопрос дается предварительный ответ, к фрейму-рассказу присоединяются соответствующие субфреймы и вопросы, которые возникают при этом, становятся, в свою очередь, активными.

"Маркеры", предложенные нами для фреймов зрительного восприятия, приобретают здесь вид более сложных конструкций. Если встать на позицию А.Ньюэлла и рассматривать понятие "точка зрения" более широко, чем мы это делали до сих пор, то, помимо поиска ответа на какой-либо вопрос, нужно указать, *каким образом* на него следует отвечать. Следовательно, **в терминале должны содержаться данные**

или предположения о том, где и как следует искать ответы для конкретизации заданий терминалов. В этом случае введенные ранее **задания отсутствия** играют роль наиболее простых вариантов подобных рекомендаций. Человек обладает целой иерархией таких правил, которая управляется внешними ситуациями и соответствует структурам "предпочтений" Дж.Уилкса.

Для синтаксических фреймов характерно стремление к полной конкретизации заданий своих терминалов, но люди вообще придерживаются более гибких правил. Как говорит Р.Шенк, "Люди обычно не разъясняют всех подробностей той мысли, которую они стремятся сообщить другим, поскольку стараются быть краткими и опускают вследствие этого предполагаемую и несущественную информацию... Концептуальный процессор использует незаполненные пробелы для поиска определенной информации в предположении или в более крупной единице рассуждения, полезной при их конкретизации". При зрительном восприятии ситуация аналогичнаю. Наблюдатель не в силах охватить взглядом коробку сразу со всех четырех сторон, и, несмотря на "стремление к краткости", при необходимости может потребоваться осуществление движения вокруг этого предмета. В соответствии с этой точкой зрения *системы* фреймов представляют собой точки выбора, соответствующие, (на концептуальном уровне) системам взаимно исключающего выбора Т.Винограда. **Различные системы фреймов представляют собой различные варианты использования одной и той же информации, локализованной в общих терминалах.** Как и в языке, в каждый момент времени следует выделять только один из множества имеющихся вариантов. На концептуальном уровне этот выбор может осуществляться в соответствии с тем, *какие вопросы следует задать относительно данной ситуации.*

Откуда берутся *вопросы?* Рассмотрение этой проблемы выходит за рамки настоящей работы, однако, кем бы и как бы ни создавались фреймы, они должны использовать некоторые общие принципы. Методы, лежащие в основе порождения вопросов, в конечном счете, образуют общий интеллектуальный стиль каждого человека. Люди, конечно, во многом отличаются друг от друга, в частности, в том, например, какие вопросы они предпочитают задавать: "Почему?" или "Каким образом я бы мог выяснить кое-что еще?", "Что мне это даст?" или "Каким образом это мне может помочь в достижении более общих целей?" и т. п.

Аналогичные проблемы не могут не возникнуть в отношении стиля даваемых *ответов.* В простейшей форме стремление к заполнению конкретными данными пробелов терминалов может показаться чем-то похожим на удовлетворение человеком голода или ликвидации неудобств, ибо воспринимается как поиск любого задания, не противоречащего наложенным ограничениям.

Хочется представить себе множество систем фреймов, которое включает, с одной стороны, простые шаблонно заполняемые структуры, а с другой - конструкции для реализации "точек зрения" А. Ньюэлла со всей их причастностью к согласованной работе генераторов вопросов, с методиками оценок предлагаемых ими решений, способами их исследования и т.д. Представляются не совсем правильными попытки использования одних и тех же видов теоретических построений для работы на различных понятийных уровнях. Нам следует предполагать, что существуют весьма различные механизмы, которые оперируют как нашими стереотипами нижнего уровня, так и всеобъемлющими стратегическими концепциями.

3. Обучение, память и парадигмы

"Природа дает ребенку различные средства для исправления всяческих ошибок, которые он может совершить по отношению к окружающим его предметам. При каждой возможности его взгляды корректируются опытом; **неудача и боль являются непременными спутниками ошибочных взглядов, тогда как удовольствие и наслаждение возникают при правильных суждениях**. Имея таких учителей, нам ничего другого не остается, как стать хорошо информированными людьми и быстро научиться рассуждать правильно...
В научной практике наблюдается нечто совершенно противоположное: ошибочные суждения, которые мы вырабатываем, не влияют ни на наше существование, ни на наше благополучие, и никакая физическая необходимость не заставляет нас их изменить. Напротив, **воображение, которое вечно старается выйти за рамки <u>истины</u>, в сочетании с самовлюбленностью и самонадеянностью, которые мы так склонны лелеять, побуждают нас делать заключения, которые непосредственно из фактов не следуют...**" *А.Лавуазье*

Парадигма (от греч. Paradeigma - пример, образец) – система форм; изменяющегося слова или целой конструкции в предположении. Здесь этот термин используется **для обозначения совокупности различных представлений об одних и тех же объектах или процессах**.

Как удается обнаружить фрейм для представления новой ситуации? Очевидно, что мы не можем разрабатывать какую-либо законченную теорию вне рамок глобальной схемы для организации знаний вообще. Если же допустить, что мы работаем в относительно узкой области, то у нас появится возможность обсудить некоторые важные вопросы:

ПРЕДВОСХИЩЕНИЕ: как сделать первоначальный выбор такого фрейма, который бы, удовлетворял некоторым заданным условиям.

ДЕТАЛИЗАЦИЯ: как выбирать и конкретизировать субфреймы, чтобы иметь возможность представлять дополнительные подробности.

ИЗМЕНЕНИЕ: где и как искать другой фрейм, если предыдущий подходит недостаточно хорошо.

ОБНОВЛЕНИЕ: что делать, если приемлемого фрейма найти не удается. Можно ли модифицировать один из существующих или надо строить совершенно новый?

ОБУЧЕНИЕ: какие фреймы должны запоминаться в долговременной памяти, а какие модифицироваться в процессе накопления опыта?

Широко распространено мнение, что память - это нечто совершенно отличное от остальной части мышления; однако для того, чтобы создать полезную и эффективную модель памяти, требуются те же стратегические принципы, которые используются при анализе других аспектов мышления.

Мы говорим о человеке, что он умен, если он может **быстро находить весьма подходящие фреймы**. Система извлечения информации у этого человека работает лучше, чем у других при выработке правдоподобных гипотез, при определении условий, которым должен удовлетворять новый фрейм, и в части использования тех знаний, которые были приобретены на "неудачных" ветвях процесса поиска.

Организация удачной модели памяти является задачей не менее сложной, чем решение любых других задач того же класса. Поэтому хороший механизм извлечения информации лишь частично может базироваться *на* основных "врожденных" механизмах. Он в значительной степени должен зависеть от (приобретенных в процессе обучения) данных о структуре своих собственных знаний. Наше предложение по этому вопросу объединяет в себе ряд компонентов, включая **методы распознавания образов, теорию групп и сети подобия**.

При зрительном восприятии, например, комнаты или процессе понимания рассказа **человек собирает сеть, состоящую из фреймов и субфреймов**. В этой сети представляется все, что было замечено, или существует в предположении, будь то правильным или ошибочным. Мы уже отмечали, что активный фрейм, т.е. участвующий в процессе "понимания" системой представления истин и информации (в данном случае) внешней для нее ситуации, не может оставаться таковым, если не выполнены условия, определяемые маркерами его терминалов. Теперь мы вводим **постулат о том, что все фреймы, задания которых конкретизированы в некоторой ситуации, обязательно должны выступать в качестве значений для терминалов фреймов более высоких уровней.** Это же относится и ко всем существенным фрагментам "данных", собранным при наблюдении и хранимым в памяти системы представления истин и информации. Конечно, здесь не может не быть исключений. Мы должны предусмотреть возможность существования определенного числа групп данных, связанных с чем-то подобным регистрам "кратковременной памяти". Следует, однако, помнить, что лишь незначительная часть информации может длительно храниться в памяти, если она не связана с соответствующими фреймами. Предложенная схема концептуальна по своей сути, но, тем не менее, в определенных областях следует допустить существование других типов "зацепок" в памяти и специальных сенсорных буферов.

3.1. Требования к памяти

Мы можем представить себе, что память - это система, управляемая двумя дополняющими друг друга потребностями. С одной стороны, это потребность в правильном представлении объектов (сообщений, новых предметов и пр.), которое осуществляется с помощью подключения их к более крупным фреймам; с другой - потребность в

конкретизации заданий, терминалов. Остальная часть этой системы будет стремиться сбалансировать их требования, но не столько на базе "общих принципов", сколько в соответствии со специальными знаниями и теми условиями, которые определяются текущими целями.

Если применение какого-либо фрейма окажется неудачным (например, нельзя подобрать нужное задание по какому-то важному условию), следует что-то предпринять, но что? На наш взгляд, существуют следующие возможные пути приспособления фреймов

СОПОСТАВЛЕНИЕ ОБРАЗЦОВ. Когда ничего более *конкретного найти не удается, можно воспользоваться одним из основных механизмов - механизмом ассоциативного поиска. Это может принести - успех в относительно простых ситуациях, но должно являться лишь вспомогательным средством при использовании других тактик.*

ОПРАВДАНИЕ. Довольно часто может быть найдено объяснение или оправдание неудаче при согласовании. Например, маленький по своим размерам стул может относиться к классу предметов "игрушки".

СОВЕТ. Фрейм содержит четкие указания относительно того, что делать при его неудачном согласовании.
Ниже мы рассмотрим обширную "сеть подобия", получаемую в результате обучения и хранящую подобные знания.

РЕЗЮМЕ. Если фрейм нельзя приспособить к реальной ситуации или заменить другим, от него следует отказаться. Но прежде надо четко указать причину отказа или составить резюме так, чтобы в будущем помочь любому процессу в решении задач, связанных, с аналогичной переориентацией субфреймов.

Все эти четыре варианта жизненно необходимы, поэтому рассмотрим их более подробно.

3.2. Сопоставление образцов

При замене одного фрейма другим не имеет смысла начинать весь процесс согласования заново. Но каким образом можно сохранить то, что было уже ранее установлено? Будем рассматривать лишь тот

случай, когда при отсутствии специальных знаний система обращается за помощью к некоторой "общей" стратегии. Ни на один из универсальных методов мы не можем в данном случае возлагать очень больших надежд, но если удастся подыскать такой фрейм, который использует достаточное число общих со **старым фреймом терминалов**, то некоторые из общих заданий можно будет сохранить и это, видимо, будет лучшим выходом из положения.

Данная проблема может быть сформулирована следующим образом: допустим, что E есть штраф за потерю очного полностью согласованного терминала, а F - потери от того, что какой-то другой терминал не может конкретизировать свои задания. Тогда, если E больше F, то любой новый фрейм должен сохранять старый субфрейм. При наличии какой-либо приоритетности среди терминалов типичный запрос на вызов нового фрейма должен включать:

1. *Поиск фрейма с возможно большим числом терминалов, общих с конкретизированными терминалами (a, b,… z) старого фрейма и расположенных в порядке убывания их приоритетности.*

Следует помнить, что заменяемый фрейм является обычно субфреймом некоторого более крупного фрейма и ПОЭТОМУ должен удовлетворять маркерам того терминала, к которому он подключен. Это предопределяет наличие другой формы обращения к памяти, направленной в ее иерархии скорее вверх, нежели вниз.

2. *Поиск или создание нового фрейма, обладающего свойствами (a, b,…, z).*

В том случае, когда мы в большей степени подчеркиваем различия, а не их общие характеристики, можно объединить оба правила в одно.

3. *Помимо нового фрейма, во всем подобного старому фрейму, за исключением различий (a, b, …, z) между ними.*

Реализация правил 1 и 2 может быть выполнена с помощью процесса поиска в памяти с параллельной выборкой или хаш-кодированием (метод функции расстановки), если терминалы или свойства (a, b , ..., z) - **простые атомарные понятия**. (В любом случае должен существовать какой-то механизм для поддержания работы

генерирующих программ или для одного из видов сопоставления образцов). К сожалению, для осуществления всего этого имеется так много различных способов, что нельзя указать на какие-нибудь характерные для данных целей требования к конструктивным особенностям этих механизмов.

Хотя правила 1 и 2 формально являются частными случаями правила 3, на практике они различны, поскольку для работы со сложным правилом 3 требуются знания о том, каковы же эти различия (a, b , ..., z). Действительно, последнее правило слишком сложно чтобы его можно было использовать по той схеме, как это предполагалось выше: приведем доводы в пользу того, что следует в большей степени полагаться на особые, выявленные в процессе обучения различия между парами фреймов, чем на общие принципы.

Нужно еще раз подчеркнуть, что оснований считать реальной возможность быстрого достижения успеха практически нет. Для решения новых и трудных задач необходимо построить новую структуру представления, а это потребует применения как общих, так и специальных знаний. Работа П.Фримэна и А.Ньюэлла, в которой рассматривается проблема конструирования структур, дополняет данную работу в одном важном направлении. В ней рассматривается вопрос о том, каким образом следует создавать структуры, удовлетворяющие наборам функциональных требований, т. е. условиям, связанным с достижением целей, в дополнение к тем требованиям, которые определяются необходимостью использования определенных субфреймов и символов.

3.3. Оправдание

Можно допустить, что фрейм представляет собой "идеальный" образ реального предмета или явления (истины). В том случае, когда какой-то фрейм не может быть согласован с действительностью (**по своей природе <u>все идеальное</u> <u>ошибочно</u>)**, он должен быть заменен, другим фреймом. Особенность "идеальных" образов заключается в том, что они являются превосходными **упрощениями действительности**; они привлекательны своей простотой, но реальные возможности и преимущества таких структур зависят от дополнительных знаний об их взаимодействии между собой. Следовательно, у нас нет особых причин отказываться от такого "идеального" образа только потому, что не удалась попытка заполнения конкретными данными всех его пробелов,

правда, если при этом подобные расхождения можно объяснить, прибегнув к помощи межфреймовского взаимодействия. Ниже приводятся примеры, в которых "оправдание" такого рода может помочь при неудаче процесса согласования.

ЗАСЛОНЕНИЕ. Мы знаем, что у стола четыре ножки. Однако стоящий рядом стул может заслонять одну из них; чтобы убедиться в правильности такого "оправдания", можно попытаться найти Т-образные соединения элементов конструкций или проанализировать тени от этих предметов.

ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ. Геометрически ножка стула представляет собой обыкновенный брусок, но что более важно, в функциональном отношении она представляет собой опору. Поэтому в качестве приемлемой замены для всех ножек можно предложить конструкцию, состоящую из центрального стержня и прикрепленной к нему сверху пластины. **Многие предметы могут использоваться для самых различных целей, поэтому в их описании должен преобладать <u>функциональный, а не физический подход</u>.**

ПОЛОМКА. Отсутствие необходимых деталей можно объяснить либо тем, что их просто нет, либо тем, что они поломаны. В реальной жизни существует огромное число причин - ниспровергателей наших идеальных представлений.

ПАРАЗИТНЫЕ КОНТЕКСТЫ. Предмет, отличающийся от стула только лишь размерами, может быть (и, видимо, является) игрушечным стулом. Сетование на то, что кто-то или что-то является "слишком маленьким", может быть неправильно интерпретировано с помощью контекстов, содержащих другие предметы из мира детей и их игр.

В большинстве этих примеров для исправления заданий и сохранения текущего фрейма используются знания достаточно общего типа; они могут быть присоединены к фреймам более высоких уровней. В дальнейшем упор будет сделан на более локальные типы знаний, которые естественно разместить в самом фрейме в целях выработки рекомендаций относительно его собственной замены.

3.4. Суждения и сети подобия

"Подтверждение правильности утверждения Наполеона (если оно вообще принадлежит ему) о том, что люди, пытающиеся представить себе любое событие во всех его будущих подробностях, не имеют данных для того, чтобы стать командирами, следует искать именно в первой части этого высказывания. Командир, ожидающий, что ход сражения будет протекать точно так же, как и раньше, через две минуты после его начала обнаружит нечто совсем иное. Его ожидания не оправдаются, а в запасе будет в лучшем случае еще одна схема, да и та не сможет быть ему полезной длительное время. Не менее правдивым может оказаться и такой вариант, когда после первой неудачи этот горе-командир вспомнит сразу так много различных примеров, что не будет знать, как поступить в данном конкретном случае. Слишком конкретизированный опыт прошлого создает такие же затруднения, как и его полное отсутствие. **Для правильного удовлетворения требований постоянно меняющейся обстановки мы должны не только уметь выделять те объекты и явления, которые имеют отношение к переменам, но и знать, какие их части (свойства) могут развиваться и изменяться, не нарушая при этом общего смысла и функциональной значимости".**

Ф.Бартлетт

При перемещении по знакомой квартире нам известна структура поиска информации, основанная на фрейме комната. Когда мы проходим через дверь D комнаты X, то ожидаем, что очутимся в комнате Y (конечно, если D не является наружной дверью). Этот факт может быть представлен с помощью трансформации простейшего типа, состоящей из указателей между двумя фреймами комнат в рамках системы фреймов типа дом. Если дом, где происходит действие, ранее не был знаком, то в своих рассуждениях логично переместиться вверх на один классификационный уровень и использовать следующее правило: когда вы выходите из одной комнаты, обычно следует предполагать лишь то, что вы очутитесь в какой-то другой комнате и не более. Недостаточность конкретной информации может быть преодолена использованием обобщений типа классов ситуаций; если же вернуться к незадачливому командиру Ф.Бартлетта, то описанная у него дилемма разрешается применением некоторой формы абстракции или обобщения.

В некотором смысле использование обобщений (классов) является неизбежным: если конкретная информация отсутствует, человек обращается к классам как к теории "первого порядка", лежащей в основе любых сложных моделей. К счастью, нет необходимости использовать классы явным образом, что могло бы повлечь за собой определённые неприятности. Если рассматривать класс в буквальном или строго математическом смысле, то это вовлекает нас в проблему иерархии, основанную на включении в нее различных понятий; порядок в такой системе будет неудовлетворительным из-за различным образом связанных между собой концепций, находящихся в разных контекстах. Этот вывод справедлив также для процедур и фреймов. Нам следует стремиться к тому, чтобы не оказаться связанными с подобными негибкими классификаторами знаний.

П.Уинстон предложил способ создания системы поиска информации, в которой могли бы быть представлены классы и обеспечен ряд дополнительных возможностей. Указатели поиска информации в такой системе можно приспособить для выражения целевых условий и результатов действий, а также для определения принадлежности понятий к определенным классам. Поскольку эта идея известна еще далеко не всем, попытаемся объяснить ее на примере, взятом из работы П.Уинстона.

Что означает ожидать увидеть стул? Обычно то, что он состоит из четырех ножек, нескольких перекладин, сидения и спинки, находящихся друг с другом в определенных отношениях. Например, ножки должны опираться на пол и находиться ниже, а спинка выше сидения; само сидение должно располагаться горизонтально, спинка - вертикально и т. д. Предположим теперь, что система зрительного восприятия не смогла обнаружить спинку: все есть (четыре ножки, ровная поверхность - сидение), а спинки нет. "Различие" между тем, что мы видим, и тем, что мы ожидали увидеть, состоит в отсутствии требуемого числа спинок, а это свидетельствует скорее о наличии не стула, а скамьи или стола.

П.Уинстон считает целесообразным снабдить каждое описание, находящееся в памяти, указателями к другим описаниям, причем **каждому указателю должен соответствовать свой <u>маркер отличия</u> между связанными им парами понятий**. Если в процессе согласования фрейма встретятся какие-либо затруднения, то они должны согласовываться с указателями, выходящими из данного

фрейма; это может снабдить систему предложениями по выбору более подходящего фрейма. П.Уинстон назвал образующуюся при этом структуру **сетью подобия**. Он предлагает поручить компьютеру вести упорядоченное сопоставление находящихся в памяти моделей во время его "холостого хода" и при выявлении между ними существенных различий вводить соответствующие указатели.

Та же информация может быть получена и в процессе согласования какой-либо реальной ситуации с хранимыми в памяти данными, ибо следующие одна за другой попытки дают результаты далеко не во всем отличные друг от друга, а различия эти всегда можно зафиксировать. Таким образом, в процессе обычного использования содержащихся в памяти данных можно получать дополнительные сведения и использовать их для расширения сети подобия. Если процедура образования новых указателей чувствительна к записи различий, относящихся к достижению цели, то результат будет еще более ценным, ибо появится возможность реализации механизма обучения на основе собственного опыта.

Можно ли на практике создать сети подобия? На первый взгляд кажется, что это может привести к неограниченному росту требуемого объема памяти. И в самом деле, если имеется N фреймов и К типов различий, то общее число указателей может достигнуть величин KNN и следует опасаться того, что:

1) если N велико, скажем, равно 107, то NN будет слишком велико и объем требуемой памяти, по крайней мере, для людей, может оказаться попросту нереальным;

2) для устранения какого-то различия или согласования фрейма может потребоваться такое большое число указателей, что система окажется малоэффективной ввиду ее плохих избирательных свойств;

3) само К может быть весьма велико в том случае, когда параметры системы вариативны.

Фактически же ни одна из этих проблем не представляется достаточно серьезной, если рассматривать их по отношению к возможностям человеческой памяти. По современным представлениям (правда, еще недостаточно обоснованным) скорость накопления информации в

долговременной памяти человека такова, что вопрос о ее насыщении затрагивать вообще не следует.

Реальность состоит в том, что нам не только не грозит опасность насыщения, но, как это ни парадоксально, связей может просто не хватить! Более того, нельзя рассчитать, во-первых, что мы получим достаточно времени, чтобы установить все требуемые связи, и, во-вторых, что каждое нуждающееся в указателе различие будет в действительности им обладать. Ниже мы рассмотрим вопрос о том, как следует хотя бы частично обойти эту проблему.

3.5. Группы, классы и аналогии

"Несмотря на то, что рассмотрение некоторых признаков, свойственных ряду таких категорий, как игры, стулья или листья, часто помогает нам узнавать, каким образом следует использовать то или иное понятие, не существует каких-то определенных совокупностей характеристик, одновременно применимых ко всем элементам данного класса и только к ним одним. Если мы сталкиваемся с ранее неизвестным нам видом деятельности, то используем для его представления тот термин, например "игра", который соответствует группе похожих действий и, как мы знаем, носит это название. Короче говоря, игры, стулья и листья являются естественными семействами, каждое из которых определяется сетью перекрывающихся, перекрещивающихся одних и тех же признаков. Существованием такой сети объясняется наша успешная идентификация объектов или действий".

Т.Кун

Чтобы организовать более "полное" функционирование сети подобия, рассмотрим следующую аналогию. В городе каждый человек может посетить любого другого жителя; однако мы не строим пути между всевозможными парами домов, а объединяем группы домов в "кварталы". Мы не соединяем один квартал с другим своей собственной дорогой, ибо знаем, что должны существовать улицы "совместного пользования". Мы не связываем дорогами один город с другим, а прокладываем магистрали между наиболее крупными областными (или другими) центрами. В пределах такой организации каждый ее член непосредственно соединен с другими на своем собственном "уровне", в первую очередь с теми, кто по смыслу

наиболее им близок; кроме того, каждый индивидуум связан, по крайней мере, с небольшим числом ключевых членов групп более высоких уровней. Таким образом, между любыми двумя элементами сети существует (если вообще существует!) весьма короткий путь, состоящий из небольшого числа последовательных связей.

При решении задач поиска подобных структурах используется иерархия связей, похожая чем-то на ту, что заложена в каждом почтовом адресе. Любой человек обладает определенными сведениями о каких-то обобщенных понятиях, например, он знает, где в его стране расположены наиболее крупные города. Населению крупного города обычно известны многие пригородные центры, а их жителям - близлежащие деревни. Ни одному из нас неизвестны все возможные в городе пути между любыми двумя домами, но зато человек может без труда запомнить самый короткий путь к своему другу, живущему в другом городе; этот путь будет лучше (короче), нежели тот, который согласно общему правилу должен проходить через главный населенный пункт его района. Для решения таких проблем используются адресные книги, с помощью которых выбираются стандартные маршруты между основными узлами в сети. Индивидуальные маршруты могут пролегать в обход главных узлов через хорошо знакомые пункты. И поскольку наши разветвленные службы транспорта успешно справляются со своими задачами, те же стандартные маршруты обеспечивают сравнительно небольшое число пересадок при движении из одного произвольного пункта в другой.

На каждом уровне имеются свои центры или капитолии. Они составляют основу групп следующих иерархических уровней. Например, между городами Нью-Хейвен и Сан-Хосе нет беспосадочной авиалинии, поскольку более рационально использовать магистральную воздушную трассу между Нью-Йорком и Сан-Франциско. Оба этих крупных города являются капитолиями на данном уровне объединения.

По мере роста сети следует ожидать, что появится необходимость в объединении тех элементов, на которые направлены наши указатели подобия. **Решения относительно того, что считать <u>главными отличительными признаками</u>, а что второстепенными, будут оказывать значительное влияние на возможности всей системы в целом. Эти решения, накапливаясь, <u>формируют концепты наших представлений об окружающем мире</u>.**

Таким образом, дивергенции и конвергенции указателей подобия, которые носят неслучайный характер и связаны с каждым D-различием, объединяют наш концептуальный мир вокруг d-групп и d-капитолиев. Отметим, что идеальным является вариант, когда в одной группе сосредоточены такие капитолии, для которых не существует ни одного более общего для них всех атрибута. В этом случае перекрестные "сходства" являются следствиями локальных соединений в нашей сети подобия. Они, несомненно, достаточны для объяснения того, каким образом мы можем предполагать, что же такое стул или игра, хотя не всегда при этом способны "логически" определить это понятие как элемент в некоторой иерархии классов. Для отражения в сети четких определений не требуется явная согласованность концептуальных групп, однако, она может понадобиться для преодоления затруднений при ориентированном на различия поиске путей, ведущих к поставленной цели.

Выбор капитолиев соответствует выбору таких стереотипов или типичных элементов, чьи заранее заготовленные значения (**задания отсутствия**) чрезвычайно полезны. Например, формы стульев могут быть самыми разными, поэтому следует тщательно **отбирать те фреймы, которые могут стать главными капитолиями в "мире" стульев**. Они будут использоваться для быстрого согласования и установления приоритетности среди самых разных различий. Те черты, который относятся к центральному элементу группы и имеют низший приоритет, либо отражают несущественные и в большинстве случаев отсутствующие свойства для всех типов стульев, либо, если требуется большая точность, служат в качестве указателей на локальные "города" и "деревни" и мире стульев. **Указатели различия могут носить функциональный, а не только геометрический характер**. Поэтому после первой неудачной попытки отнести заданный предмет необычной формы к классу "стулья", человек может проверить применимость к нему следующего функционального правила: "стул есть нечто такое, на чем можно сидеть". Это потребует проведения более глубокого анализа с привлечением таких понятий, как сила и сопротивляемость действию силы. Конечно, этот анализ не будет охватывать игрушечные стулья и стулья с таким утонченным украшением, что о применении их для сидения не может быть и речи. Их лучше рассматривать с помощью **метода "оправданий", при котором причина ищется не в виде геометрических или функциональных объяснений, а на основе контекстов, связанных, например, с миром искусства пли играми.**

Важно подчеркнуть, что нет оснований ограничивать структуру памяти иерархией одного и того же вида, а понятие "уровень" объединения может не совпадать для разных типов различий, d-капитолии могут существовать не только в виде явных деклараций, но и в неявной форме, определяемой посредством сходящихся d-указателей, которые задают их местонахождение в структурах данных.

В системе GPS (General Problem Solver - универсальная программа для решения задач) А.Ньюэлла и Г.Саймона "различия" упорядочены в иерархию неизменной структуры. Если же приоритетность различий определять в зависимости от решаемых задач, то можно на базе тех же структур памяти достигнуть значительно большего; получающийся при этом решатель задач потеряет свое изящество и простоту, что явится своеобразной платой за отход от используемой в системе GPS логики предикатов первого порядка.

Следует, наконец, отметить и тот факт, что нет надобности создавать какой-то особый механизм для организации групповых структур. **Можно предположить, что в процессе эволюции ранее образовавшиеся фреймы будут стремиться стать капитолиями для своих более поздних собратьев** (если, конечно, это не очень противоречит опыту), поскольку всякий раз, когда использование одного стереотипа оказывается успешным, его центральное положение подчеркивается присоединением еще одного указателя. Иначе говоря, выделение новых центров знаний происходит в значительной степени под влиянием внешних факторов: словарного запаса, поведения окружающих нас объектов, знания общечеловеческой культуры, полученного нами в школе и семье. На каждом этапе структура уже приобретенных знаний оказывает решающее влияние на весь последующий ход их расширения и углубления. Описанные выше группы и формы представления знаний должны возникать в результате взаимодействия механизмов памяти с внешним миром.

3.6. Аналогии и альтернативные описания

Мы рассмотрели вопросы применения различных фреймов одной и той же системы для описания одной ситуации с помощью различных средств: **для изменения положения при восприятии зрительных изображений и для смещения акцентов при использовании языка.** Например, в эпизоде с волком и ягненком два фрейма используются в ситуационной паре типа "до - после". Иногда при поиске решений мы

применяем два или большее число описаний, если ищем какие-то аналогии или используем разные виды анализа для оценки одной и той же ситуации. Для решения сложных задач использование одной проблемной области обычно оказывается недостаточным! Предположим, что в вашем автомобиле стал разряжаться аккумулятор. Вы полагаете, что произошло короткое замыкание или неисправен генератор. Генератор можно представить себе как механическую систему: ротор снабжен ведомым колесом, приводимым в движение ремнем от двигателя. Тут же возникают вопросы: достаточно ли туго натянут ремень и имеется ли этот ремень вообще? С точки зрения механика выходным элементом является кабель, идущий от аккумулятора к каким-то другим устройствам: исправен ли он, хорошо ли затянуты болты, прижимаются ли щетки к коллектору? С точки зрения электрика генератор должен описываться иначе. Ротор рассматривается не как вращающееся устройство, а как катушка, индуцирующая магнитный поток. Щетки и коллектор выполняют роль электрических переключателей. Выходным параметром является электрический ток, протекающий по паре проводов, которые соединяют щетки, схему управления и аккумуляторные батареи.

Таким образом, ситуация описана нами в двух различных системах фреймов. В одной из них якорь является механическим ротором со шкивом, а другой - проводником в постоянно меняющемся магнитном поле. Одни и те же (или аналогичные) элементы совместно ИСПОЛЬЗУЮТСЯ терминалами разных систем фреймов, и эти системы могут трансформироваться одна в другую.

Различия между двумя этими системами фреймов существенны. В "электрической" системе фреймов шасси автомобиля всегда соединено с одной из клемм аккумулятора. Тот, кто ставит диагноз о неисправности, должен использовать в своих рассуждениях и действиях оба вида представления. Если ток не течет, то часто это является следствием того, что проводник его не проводит. Для этого случая трансформация приводит к фрейму, с помощью которого можно установить, что прочное механическое соединение исключает обрыв в электрической цепи. Поэтому любое нарушение проводимости, выявленное путем электрических измерений, заставит нас искать неисправность в "механической" системе фреймов. В нашем обычном понимании "исправность" чего-либо есть синоним понятия "механическая исправность" и поэтому проводимый диагноз должен заканчиваться именно в "механической" системе. В конечном итоге мы

можем обнаружить неисправное механическое соединение, выявить разболтавшееся крепление проводника, коррозию, износ и т. д.

Но почему генератор должен быть представлен двумя отдельными системами фреймов, а не одной объединенной структурой? В рамках таких сложных задач человек не может сразу представлять себе очень большое число различных подробностей. В каждый момент времени человек должен мыслить в пределах разумно простой структуры. Видимо, любая проблема, которую вообще может решить человек, вырабатывается в каждый момент времени в рамках небольшого контекста и ключевые операции, используемые при поиске решений, связаны с формированием соответствующей рабочей среды. Действительно, поиск неисправности в предыдущем примере требует одновременной работы сразу с **тремя фреймами: визуальным, электрическим и механическим.** Если данные электрических измерений указывают на неисправность механического соединения, то визуальный фрейм используется человеком при ее отыскании.

Существуют ли общие методы построения компетентных систем фреймов? На этот вопрос можно ответить и положительно, и отрицательно. В самом деле, нам известны некоторые во многом очень ценные стратегии для приспособления уже существующих фреймов к новым целям; вместе с тем следует подчеркнуть, что люди не обладают каким-то единым и таинственным механизмом, позволяющим решать *все* сложные проблемы. Мы не должны впадать в крайность и требовать существования узких теорий человеческого поведения, которые объясняли бы невозможные вещи. Не следует ожидать, что для любой ситуации можно будет отыскать или, в крайнем случае, построить в точности соответствующий ей фрейм. Нам, однако, предстоит много поработать над этим вопросом; важно напомнить здесь и о том вкладе общечеловеческой культуры, который был сделан ею в разработку вопросов оценивания сложности различных проблем. Опытному механику при выполнении своей обычной работы нет нужды что-либо изобретать; у него уже сформировано представление, например, о двигателе как совокупности систем зажигания, смазки, охлаждения, синхронизации, смешивания топлива, трансмиссии, сжатия и т.д. Система охлаждения подразделяется на схему циркуляции жидкости, воздушного потока и др. Большинство из таких "обычных" проблем решается систематическим применением аналогий, которое обеспечивается трансформациями между парами фреймов-структур. Элементы огромной сети знаний, приобретенных в школе, из книг, в процессе профессионального обучения, а также из

других источников, накрепко связаны друг с другом **указателями различия и релевантности**. Нет сомнения в том, что культура оказывает большое влияние на формирование такой сети благодаря обычному использованию одинаковых слов при объяснении различных мнений об одних и тех же предметах или явлениях.

Что можно сказать о взаимодействии между элементами фреймов-структур, которое тесно связано со всеми рассматриваемыми здесь вопросами? Гештальт-психолог потребовал бы выполнения такого синтеза (на базе имеющихся у человека сведений), при котором двигатель будет представляться как единое целое. Неплохо, однако, для начала вспомнить, что автомеханик сможет выявить неисправность (если вообще сможет ее отыскать), возникшую как результат взаимодействия трех или более элементов двигателя только после изнуряющей замены многих его исправных частей. Следует признать, что стремление к полным синтетическим представлениям нецелесообразно и не может выступать в качестве требования для теоретических построений. В действительности должна существовать некоторая структура, объединяющая различные концептуальные типы фрейма "двигатель". Но она тоже может быть относительно простой. К уже рассмотренным типам фрейма "двигатель" следует, видимо, добавить еще один - **суперфрейм, терминалы которого будут указывать на первые три типа фрейма (электрический, механический и визуальный) и, будучи связанными друг с другом указателями, сообщать, как и когда именно следует использовать эти субфреймы.** Каждая сложная "понимающая" система, по всей вероятности, содержит подобные структуры суперфреймов, которые управляют использованием субфреймов.

Привлекательной кажется идея о том, что более крупные понятия формируются на подсознательном уровне человеческого мышления. По наблюдениям Х.Пуанкаре, после периода умственной активности, за которым следует более длительный период ее спада, на человека находит внезапное озарение. **Подсознательную деятельность** (если следовать взглядам Х.Пуанкаре) следует рассматривать как **<u>комбинаторный эвристический поиск</u>, в котором вероятность успеха зависит в первую очередь от качества ингредиентов, введенных во время активной умственной работы**; эти элементы объединяются самыми разными способами до тех пор, пока не будет получена такая конфигурация, которая выдержит определенный вид проверки. "Я говорил о чувстве абсолютной уверенности, сопровождающем наше вдохновение..., оно часто вводит нас в

заблуждения, но мы обнаруживаем это только тогда, когда пытаемся привлечь на свою сторону доказательства", Х.Пуанкаре.

Итак, продуктом вдохновения является не полностью детализированное решение, а "точка отсчета" или план, доведенный до сознательного уровня мышления, потому что он прошел через порог "эстетической разумности".

Что же касается взглядов Х.Пуанкаре, то он, видимо, придерживался (по крайней мере, в данном вопросе) концепции холизма, ибо в его понимании "элегантные" математические объекты есть не что иное, как объекты, "элементы которых расположены настолько гармонично, что ум в состоянии охватить их всех сразу, целиком и в то же время представлять себе все возможные подробности". Из нерассмотренных остался еще вопрос о том, требуется ли для работы фильтров, которые передают на сознательный уровень новые описательные комбинации. проведение активного их анализа или действия таких фильтров смогут получить объяснение на основе более простых операций согласования и поиска информации. К несчастью, математики, за исключением Х.Пуанкаре, Д.Пойа и некоторых других, не сумели внести значительный вклад в понимание механизмов решения задач. Видимо, это не было следствием их приверженности концепции "элегантности", которая передавалась от одного поколения другому как почитаемое, но не объяснимое и не проанализированное качество. В любом случае не видно оснований считать, что подсознательная деятельность отличается либо мощными параллельными вычислениями, либо необычным синтезом, предусматриваемым философией холизма. Более правдоподобно выглядит предположение о проводимых на этом уровне быстрых и неглубоких исследованиях с привлечением тех данных, которые были подготовлены в процессе активного анализа материала. Аспект подсознательного может только указать на отсутствие "аннотации" и какой-либо регистрации мыслительных операций: в противном случае мы получили бы возможность рассмотрения и анализа этих процессов. Вопрос относительно сложности фильтров остается, по-прежнему, открытым.

3.7. Использование фреймов в эвристическом поиске. Резюме

Широкое распространение получила идея о том, что важны все аспекты **представления информации с помощью "пространства**

задачи"; однако мысль о том, что описания могут быть полезны и для самих программ, и для авторов этих программ, не стала столь же популярной. Прогресс в понимании этого ключевого момента был фактически задержан остроумными схемами, созданными для того, чтобы избежать явных манипуляций описаниями.

Основным устремлением, особенно при доказательстве теорем и моделировании игр, была разработка средств, ведущих к систематическому уменьшению протяженности поиска в пространстве задачи. Иногда простая задача может быть решена при помощи последовательного перебора допустимых методов решения: перебор производится до тех пор, пока один из методов не даст положительного результата. В более сложных ситуациях используются усовершенствованные локальные правила поведения, а также варианты "восхождения к вершине" в пределах пространства задачи. Однако если нам и удается таким способом решить определенную задачу, мы получаем мало сведений о пространстве задачи и, следовательно, не повышаем свою квалификацию, что весьма пригодилось бы нам в будущем. **Наиболее разработанными в эвристических методах являются игровые методы поиска решений, в которых используются различные стратегии для уменьшения дерева перебора, оценки терминальных вершин и выработки разумного хода**. Однако даже в тех системах, где применяются различные способы - организации иерархий символьных целей, отсутствует "осознанный" самой системой подход к процессу поиска решений и не совершенствуется качество представления информации. Рассмотрим следующее более совершенное и мощное правило.

Главной целью при решении задач должно быть стремление лучше понять пространство задачи и найти такие представления, в рамках которых данная задача решается довольно просто. Цель поиска состоит в том, чтобы **получить информацию для формирования надлежащего представления**, а не для нахождения решения, как это обычно предполагается; после того, как удастся соответствующим образом представить это пространство задачи, решение найти будет значительно легче. В частности, Мински является противником того, что значимость интеллектуального эксперимента должна оцениваться либо относительно категорий "неудача - частичный успех - успех", либо с помощью таких понятий, как "улучшение ситуации" и "уменьшение различия". Применение какого-то метода или изменение представления могут быть ценными лишь в том случае, когда они ведут к совершенствованию стратегии проведения последующих

экспериментов. В более ранних формулировках роли стратегий в эвристическом поиске эти возможности не были выделены, хотя в неявной форме они содержались в рассуждениях о задачах планирования.

Каким образом можно объединить новое правило с классической стратегией минимакса? Пусть мы находимся в определенном узле A дерева решения (играя в какую-нибудь игру, например, шахматы) и исследуем два (или более) возможных хода, скажем B и C. Каждый из этих ходов получает значения оценок V(B) и V(C). Затем оба этих значения объединяются с помощью функции M для того, чтобы выработать одну общую оценку

S(A)=M(V(B),V(C)).

По существу, с помощью функции M должны подводиться итоги поиска на всем дереве ниже узла A и определяться оценка позиции A.

Посмотрим, в чем заключается цель подобных действий. Если можно было бы произвести поиск на всем дереве перебора, то мы смогли бы использовать найденную в каждом узле оценку S для принятия решения о том, какой следующий ход лучше всего сделать. Если, однако, оценка S дается просто в виде числа, то на этой основе невозможно будет провести значительные рассуждения, требуемые для анализа существующей ситуации. Если значение S(B) невелико, то можно предположить, что B - неудачная позиция. Но если мы хотим, чтобы генератор ходов не повторял своих прежних ошибок, сообщение S должно включать некоторую дополнительную информацию относительно того, чем нас не удовлетворяет позиция B или как поступить в данном случае. Нам фактически требуется итоговое объяснение того, что обнаружено в процессе поиска; поскольку мы работаем с деревом перебора, нам требуется также рекурсивное суммирование подобных резюме.

Рассмотрим проблему, названную нами "расхождением резюме". Если резюме для ситуации A содержит (в общем случае) любое явное описание для B и C, то существует опасность того, что любая схема рекурсивного описания будет повторять дерево ходов; это приведет к столь же длительному поиску итогового сообщения, как и поиску самого решения. Чтобы этого не произошло, можно воспользоваться довольно простым способом - ограничить размеры самого резюме. В

этом случае следует позаботиться о том, чтобы избежать сильного уменьшения информативности сообщений. Во фреймах-описаниях важные черты и отношения, находящиеся на верхних уровнях, могут служить в качестве резюме, а вспомогательные описания становятся доступными лишь по необходимости. Вопрос о том, какая часть проанализированного дерева должна оставаться в долговременной памяти, а какая отбрасываться после того, как сделан очередной ход, зависит от других аспектов использования участником игры всего накопленного им опыта.

Какие принципы должны лежать в основе образования резюме? И в этом вопросе концепция фреймов демонстрирует свою гибкость. Вместо того чтобы попытаться ограничить сообщения какими-то жесткими форматами, мы может построить набор "резюме"-фреймов для каждого данного случая; любой фрейм будет вызываться, когда его терминалы подходят к описаниям ситуаций более низких уровней, а маркеры согласуются с текущими целями. Таким образом, каждый из этих фреймов выполняет свою работу только тогда, когда он соответствует текущей ситуации. Например, у человека могут быть самые разные фреймы типа "шахматной вилки". Если конь занимает такую позицию, что угрожает одновременно и шахом, и взятием ладьи, то активируется фрейм вилки, соответствующий следующему условию: при любом из двух возможных ходов теряется та фигура, которая не изменит своей позиции. Как только будет активирован этот фрейм, он может дать конкретную рекомендацию, вероятно, следующего содержания: генератор ходов того игрока, который попадает под вилку, должен выяснить, не может ли какой-нибудь ранее сделанный ход обеспечить защиту того поля, откуда исходит угроза вилки.

3.8. Фреймы в качестве парадигм

"До тех пор, пока не была создана эта парадигма схоластов (средневековая теория "первого толчка"), маятники как таковые не были известны людям, а ученые видели в них только качающиеся камни. Существование маятников было открыто благодаря изменению парадигмы, очень напоминающему переключение гештальта. Следует ли нам описывать то, что отличает взгляды Галилея от воззрений Аристотеля или Лавуазье от Пристли с позиций трансформации зрительных образов? Действительно ли они видели разные вещи, когда смотрели на одни и те же предметы? Имеются ли у

нас какие-то основания утверждать, что они проводили свои исследования, находясь в разных мирах? Я отчетливо предвижу трудности, которые могут возникнуть в том случае, если предположить, что когда Аристотель и Галилей смотрели на качающийся камень, первый видел в этом лишь несвободное падение тела, а второй - маятник. Тем не менее, я убежден, что нам следует научиться находить смысл в утверждениях, подобных данному утверждению".

Т.Кун

Согласно предложенной Т. Куном эволюционной модели **наука развивается с помощью установленных описательных схем.** Крупные открытия являются результатом новых парадигм, новых способов описания вещей, которые приводят к новым методам и методикам. В конце концов, изменяется содержание научного знания. Т.Кун предпочитает применять свою весьма эффективную схему нового описания на уровне крупных научных революций; видимо, эта идея применима и к проблемам повседневного мышления. Действительно, последнее процитированное предложение Т.Куна подтверждает ту его точку зрения, что в визуальном восприятии парадигмы должны играть не метафорическую, а, скорее, самостоятельную роль, а это именно то, что предлагается в концепции фреймов.

Когда обычно наши воззрения не пригодны, когда не удается отыскать в своей памяти эффективные системы фреймов, нам следует построить новые системы, которые позволят правильно отразить новые реалии. По всей видимости, обычным следует считать способ построения новой, системы из двух или более старых систем с последующим редактированием или "доводкой" ее до такого совершенства, когда она во всем будет соответствовать имеющимся обстоятельствам. Но каким образом можно это сделать? Можно сформулировать эту задачу так: **построить систему фреймов с наперед заданными свойствами**. Подобная постановка задачи может упростить решение, поскольку позволяет разбить его на **два этапа: вначале формулировка требований, затем само решение проблемы.**

Этот путь, однако, несвойственен процессу человеческого мышления, ибо требования никогда не формулируются все сразу, а новая система не строится по заранее и полностью построенному сценарию. В

действительности неудовлетворенные требования осознаются нами последовательно в процессе видоизменения непригодного для нас представления в виде тех или иных недостатков или "дефектов".

"Доводка", фундаментально важный его компонент, обладает своими особыми методами и процедурами. Каждый нормальный человек имеет возможность пользоваться ими в полной мере; в противном случае он не научился бы ни видеть, ни говорить. Обратимся теперь к работам И. Голдштейн и Дж.Суссмана, в которых рассматриваются вопросы явного использования знаний о доводке при обучении символьным представлениям. В этих работах строятся новые процедуры, которые должны удовлетворять многочисленным требованиям с помощью простых и вместе с тем мощных методов. Перечислим некоторые из них:

1. *Первую попытку сделайте с помощью простого объединения процедур, каждая из которых позволяем достигнуть одну определенную цель.*

2. *Если что-то получается не так как нужно, попытайтесь представить один из дефектов как особый (и нежелательный) тип взаимодействия двух процедур.*

3. *Примените тот "метод доводки", который согласно информации, имеющейся в памяти, дает хорошие результаты при исправлении этого особого вида взаимодействия.*

4. *Составьте резюме по итогам выполненных операций и включите его в хранящуюся в памяти "библиотеку методов доводки".*

Эти методы могут показаться несколько наивными, однако в тех случаях, когда новая проблема не слишком отличается от старых, имеются значительные шансы на успех, особенно при правильном подборе процедур-кандидатов на совместную работу. Коли же новая проблема резко отличается от всех предыдущих, то не следует ожидать, что вообще должна существовать такая теория обучения, которая будет хорошо работать в этих условиях. Без структурированного познавательного плана, без "почти промахов" П.Уинстона и без хорошей подготовки к решению проблем данного типа нельзя ожидать появления совершенно новых парадигм, как бы остро мы в них ни нуждались.

Что представляют собой "виды взаимодействий" и "методы доводки"? Самым простым для них, видимо, следует считать тот случай, когда результат достижения первой цели оказывает влияние на некоторые условия, необходимые для достижения второй цели. В этом случае можно предложить использовать эту предпосылку в качестве нового условия; известны, однако, такие ситуации, в которых применение только этой методики не приведет к успеху, ибо это новое условие несовместимо с первой целью.

Если задать вопрос о наиболее важных задачах в области искусственного или естественного интеллекта, то среди них следует отметить, во-первых, проблему взаимодействия между рассмотренными выше идеями и, во-вторых, использование многочисленных вариантов представления для изучения одной и той же ситуации с нескольких точек зрения. Проведение исследований в этих направлениях потребует новых идей относительно взаимодействий между трансформируемыми элементами. Здесь проявляется определенная ограниченность представления о системе фреймов, взятой в изолированном виде. Образование новых представлений на базе старых является сложным процессом, в рамках теории фреймов эта задача может быть решена лишь с помощью сложного предварительного ввода данных (конечно, если ее вообще можно решить). Более того, требуется особое искусство при разработке методов решения данной задачи, которую можно считать одной из основных в теории интеллекта.

4. Управление

4.1. Централизация управления

Выше уже затрагивался вопрос о процессах, которые манипулируют системами фреймов. Далее мы не будем касаться тех аспектов, которые связаны с длительным управлением процессами мышления, включая такие проблемы, как многоцелевое управление, распределение времени между задачами, распределение памяти, энергетических и других видов ресурсов.

В значительно более короткие промежутки времени - назовем их **эпизодами** - деятельность механизмов человеческого мышления и понимания, видимо, направлена на то, чтобы отыскать подходящий в

данной ситуации (будь то планирование или распознавание образов) **фрейм и заполнить задания отсутствия его терминалов конкретными данными. Это дает нам возможность представить крупные проблемы в виде совокупности более мелких**, а также затрагивает все обычные вопросы эвристического программирования, например:

ПОИСК СВЕРХУ - ВНИЗ ИЛИ ГОРИЗОНТАЛЬНЫЙ. Следует ли вначале обойти все терминалы или же попытаться осуществить полное заполнение пробелов одного, центрального терминала? В действительности, ни тот, ни другой вариант не следует считать достаточно хорошим. Человек обычно стремится "семь раз отмерить и один раз отрезать", однако всегда должна существовать возможность немедленной обработки субфреймов, вызванных интересным или неожиданным событием.

ЦЕНТРАЛЬНОЕ УПРАВЛЕНИЕ. Должен ли фрейм после своей активации взять на себя управление и руководить заполнением своих собственных пробелов, или же эта операция должна вестись под руководством какого-то центрального процесса? И здесь ни одна из этих двух стратегий не является наилучшей. Ни демон, ни любой другой локальный процесс не может обладать знаниями обо всей ситуации в целом, достаточными для принятия правильных решений; однако ни один "руководитель" верхнего уровня не может знать требуемое количество подробностей.

Видимо, оба вопроса следует попытаться решить на основе, предложенной У.Мартином в противовес идее о "поддержке" и задуманной как стратегия обращения с ошибками и неудачами. Нельзя ни передавать управление подчиненным структурам, ни полностью сосредоточить его на верхнем уровне; поэтому нам требуется такой интерпретатор, который имел бы доступ и к целям верхнего уровня, и к работе отдельных демонов. **Терминалы различных типов нуждаются в различных типах процессов, поэтому одной стратегией здесь не обойтись.** Заполнение пробелов терминала стены фрейма комнаты предусматривает поиск и заполнение конкретными данными субфрейма "стена" более низкого уровня, в то время как конкретизация терминала "дверь" предусматривает присоединение фрейма комнаты к фрейму дома. Для включения в каждый фрейм данных относительно действий подобного типа каждый терминал мог бы указывать интерпретатору на те инструкции, где сказано, как

собирать нужную информацию и как реагировать в случае трудностей и различного рода неожиданностей.

Итак, процесс конкретизации фреймов должен объединять в себе элементы поиска на дереве решений и активации демонов: управление поиском на дереве решений зависит от результатов проверок, которые могут выполняться с помощью демонов. После того, как фрейм комнаты будет включен в работу, он может проверить, например, основное свойство стены. Такие проверки будут производиться на дереве, узлы которого образованы всевозможными фреймами стены, а его структура обеспечивает удобный нелинейный порядок для выяснения того, какие задания отсутствия могут быть сохранены, а какие требуют дополнительного рассмотрения.

В модели, использующей демоны, предполагается, что определенные терминалы вызванного фрейма активируют связанные с ними демоны с целью наблюдения за развертыванием событий во внешнем мире. Круглый предмет, находящийся высоко на центральной стене (а на боковой - имеющий вид эллипса), по предположению, должен быть часами, и это должно получить свое подтверждение в вид найденной цифры или радиальной линии (стрелки). Если такое подтверждение не будет получено, то "наблюдатель" всё же "увидит" часы, но описать их подробно не сумеет. Четырехугольник, расположенный на уровне глаз, может представлять собой картину или окно; в таких случаях дальнейший анализ, как правило, необходим.

Цель работы системы зрительного восприятия заключается не в том, чтобы постоянно отыскивать все находящиеся вокруг нас предметы; ее главной задачей является помощь в выработке ответов на вопросы путем объединения визуальной информации с предположениями, вырабатываемыми внутренними процессами. Однако в любом случае мы должны иметь возможность правильно ориентироваться в пространстве относительно нашего ближайшего окружения, что, кстати говоря, требуется для ответа на большинство из встающих перед нами вопросов. Поэтому определенная часть процесса конкретизации будет выполняться независимо от каких бы то ни было специальных вопросов или целей. Ясно, что нам требуется такой механизм, который умел бы "идти на компромисс" и позволял бы легко заменять "слабые" задания отсутствия при выявлении демонами непредвиденных обстоятельств.

Структура управления "продукциями" А.Ньюэлла и Г.Саймона образуется последовательным расположением (в некоторой памяти) локальных правил поведения. В системах, подобных языку CONNIVER, существуют явные структуры управления высших уровней; однако и здесь многое зависит от того, какие утверждения (аналогичные "продукциям") активны в данный момент; такой вид управления полностью явным уже не назовешь. Обе эти системы характеризуются высокой степенью локального процедурального управления. Все, что удается заметить, сопоставляется со своим "образцом-предшественником", который вызывает другой субфрейм, подключает его к процессу поиска и выполняет некоторые предписанные им функции.

Здесь существует еще одна проблема: процессы, являющиеся общими для многих систем, должны быть централизованы. Это способствует и экономии ресурсов, и возможности их усовершенствования, что достигается в процессе отладки. Слишком большая автономия мешает системе быстро и правильно реагировать при появлении новых целей высокого уровня. Ниже предлагается один из вариантов, с помощью которого, по всей вероятности, можно будет преодолеть подобные затруднения. Фрейм представляется в виде "пакета" данных в процедурах, в таком же виде представляются и цели высокого уровня. Когда вызывается какой-либо фрейм, его пакет добавляется к "среде" текущей программы и определяемые им процессы получают непосредственный доступ к тем данным, которые им нужны, не ухудшая при этом возможности работы с остальными знаниями системы. Теперь следует рассмотреть два вопроса: как в деталях реализуется эта идея и насколько она хороша.

4.2. Фреймы и процесс согласования (по С.Фальману)

Рассмотрим базу данных, в которой множества фактов и демонов объединены в пакеты; любое их число может быть немедленно активировано или к ним в любой момент можно организовать доступ. Пакет может (рекурсивно) содержать любое число других пакетов это означает, что если один пакет активируется, то и все, содержащиеся в нем пакеты, также активируются, это открывает доступ к любым данным, за исключением тех которые были особым образом модифицированы или аннулированы. Таким образом, активация небольшого числа соответствующих пакетов приводит к созданию в

системе той среды, которая требуется для проведения вычислений и которая содержит только необходимые для достижения заданной цели данные и процедуры. Конечно, в некоторых случаях может оказаться необходимым дополнить активную группу новыми пакетами для того, чтобы иметь возможность найти выход из какой-то особой ситуации, но неудобство такого рода во много раз меньше, чем бремя постоянного перебора ненужных знаний или бесполезной активации демонов

Фрейм начинает процесс согласования с проверки любых сведений, которыми он уже располагает и которые могли быть получены в процессе его активации или проверки предыдущих гипотез. После этого, если разрабатываемая гипотеза еще не принята, но и не отклонена, **фрейм начинает задавать вопросы**, чтобы получить больше сведений о текущей ситуации. Характер этих вопросов будет меняться в зависимости от проблемной среды: программа, работающая в области медицины, может потребовать проведения некоторых лабораторных исследований, **визуальная программа - дать указание более внимательно изучить некоторую область пространства**. Иногда один вопрос может положить начало целому процессу распознавания: "Это может быть коровой - посмотреть, есть ли у нее вымя". **Последовательность, в которой задаются вопросы, определяется дополнительной информацией, хранящейся во фрейме**. Эта информация указывает, какие основные черты следует выявлять в рамках данной проверки, каким образом на нее может повлиять уже имеющаяся информация и во что обойдется ответ на каждый вопрос. При выявлении каждой новой черты ее описание добавляется к пакету информации вместе с указанием на то, откуда получена эта информация и насколько она надежна. Этот пакет может быть использован и при переходе к другой гипотезе. Когда встречается незатребованная информация, она проверяется и пускается в дело. Конечно, на практике невозможно добиться для такой системы идеального согласования. Для каждого возможного вида нарушений в дополнительных данных фрейма содержатся указания на то, следует ли это нарушение рассматривать как тривиальное, серьезное или **фатальное (т.е. отклоняющее возможность использования этого фрейма).** Такие индивидуальные черты, как размер обуви, пропорции тела или давление крови будут иметь перечни с указанием диапазона своего нормального изменения, а также данные других возможных значений с указанием на возможные последствия. Иногда какая-то черта может не способствовать ни принятию, ни опровержению выдвинутой гипотезы, но сама она может быть объяснена с помощью

этой гипотезы; это также должно быть отмечено во фрейме. Если анализируемая ситуация содержит нечто необычное, не предусмотренное текущим фреймом (например, оленьи рога), то система будет рассматривать этот факт как серьезное нарушение, а сами не вписывающиеся в обычную схему данные будут оцениваться в соответствии с информацией того пакета, который связан с этими данными, ибо ясно, что фрейм гипотезы не может содержать сведений о том, что делать со всякими не вписывающимися в него деталями.

Изредка какая-либо деталь будет получать для себя сильное подтверждение: если подобный факт удается заметить, то не нужно будет волноваться относительно правильности выбранного пути. Однако подобное случается крайне редко, поэтому обычная процедура состоит в том, чтобы выявленные детали накапливать до тех пор, пока либо не будет достигнут некоторый *уровень приемлемости* и гипотеза сможет перейти в разряд принятых, либо пока какое-то явное нарушение или совокупность более мелких нарушений не укажут на необходимость поиска другой гипотезы. (Представляется *уровень приемлемости* в виде простой "копилки": каждая согласованная черта увеличивает счет в этой "копилке", а каждая не согласованная, но тривиальная - его уменьшает. Возможно, что здесь может понадобиться и более сложная схема). В зависимости от ситуации уровень приемлемости может колебаться в значительных пределах: лишь один беглый взгляд может убедить нас в том, что стол по-прежнему находится на своем месте, тогда как счет в тысячу долларов заслуживает тщательной проверки, прежде чем его следует оплатить. Иногда может случиться так, что, за исключением двух-трех серьезных нарушений, вся собранная нами модель довольно хорошо согласуется с реальной ситуацией. В таком случае система должна попытаться объяснить имеющиеся разногласия. Возможно, корова красная потому, что ее кто-то вымазал краской. Возможно, у больного не наблюдается высокое давление, какое обычно бывает при подобных заболеваниях, поскольку он принимает соответствующие лекарства. Если какое-то несоответствие может, в конце концов, получить удовлетворительное объяснение, то данную гипотезу следует принять. Иногда две модели будут настолько близки друг к другу, что их можно различить только с помощью особого теста или по ряду малозначительных деталей. Наиболее простым выходом из положения будет включение в оба родственных фрейма сведений о подобии, а также инструкций для правильного выбора нужного фрейма. В медицине подобное тестирование именуется дифференциальным диагнозом.

Отметим, что такое использование фреймов придает системе значительную гибкость, особо ценную в путаных и непредвиденных ситуациях. Формально корова может быть представлена как крупное четвероногое, но наша система не встретит особых затруднений, если у коровы не будет одной ноги, хотя во всем остальном эта корова достаточно хорошо вписывается в свой образ. (Заметим, что отсутствие ноги объяснить легко, а вот присутствие лишней - намного труднее.) Если такой системе предъявить нечто, не вписывающееся ни в одно из известных ей понятий, то она может, по крайней мере, указать, к чему близко это нечто, а также его основные отличия от понятия, предложенного в качестве первого приближения. Визуальная система, организованная в соответствии с этими принципами, может легко сориентироваться при встрече с такими высказываниями, как, например, "похожий на человека, только ростом 25 метров и зеленый". При определенных обстоятельствах такие описания могут образовывать ядра новых фреймов распознавания, представляющих собой законные, хотя и не имеющие наименования, концепты.

Важной чертой фреймов распознавания (и тех категорий, которые они представляют) является то, что они могут образовывать <u>иерархические структуры</u>. Благодаря этому система может вырабатывать гипотезы на многих уровнях, от весьма общих до очень конкретных, например: животное некоторого вида, четвероногое средних размеров, собака, колли, кличка Лесси. Каждому уровню соответствует свой фрейм распознавания, однако фреймы, с помощью которых порождаются конкретные гипотезы, включают в себя пакеты фреймов более высоких уровней; так, например, если в системе активирован фрейм "собака", то ей доступна информация фрейма "животное". Конечно, конкретный фрейм может содержать такие сведения, которые будут исключать из рассмотрения некоторые более общие данные: фрейм "утконос" будет включать в себя информацию фрейма "млекопитающее", но должен исключить сведения о живородящем варианте появления своего потомства. Часто общий фрейм будет использовать в качестве образца одно из своих конкретных проявлений; фрейм "млекопитающее" может скорее призвать на помощь фреймы "собака" или "корова", а не пытаться обеспечить соответствие входной фразы некоторой схематической модели идеального, но неконкретного животного. В подобном случае единственное различие между использованием понятий "млекопитающее" и "корова" будет заключаться в том, что во втором варианте переход к какому-либо иному конкретному представлению

будет более сложным; в целом же проверке будут подвергаться одни и те же признаки.

Отметим, что подобная организация системы допускает существование большего числа различных иерархических сетей, которые могут перекрывать друг друга в самых различных (и интересных!) сочетаниях; так, например, с точки зрения зоосистематики дракон "комодо" должен быть пресмыкающимся, однако у него имеется четыре ноги и по своим повадкам он ближе к повадкам собаки, а не змеи. Чтобы решить, как представлять эти запутанные ситуации и что с ними делать, требуются дальнейшие изыскания. Некоторые фреймы следует считать фреймами-"паразитами", поскольку единственное их назначение состоит в том, чтобы прикрепиться к другим фреймам и тем самым изменить эффект от их применения. (Может быть, здесь более подходит термин "вирусный фрейм".) К фрейму "корова" может прикрепиться фрейм "статуя" и тем самым исключить такую его черту, как способность двигаться, изменить вид материала (мясо, скажем, на гипс), а формы оставить нетронутыми. Можно к животному добавить понятие "мифический" и сделать более правдоподобными его возможности самостоятельно летать, перевоплощаться, рассказывать сказки на латинском языке и менее правдоподобным его физическое существование. Тот же механизм может использоваться для более практических целей, например, чтобы учесть возможные осложнения для различных видов болезней. И еще одно замечание: нет ничего необычного в том, если к одному фрейму прикрепятся несколько фреймов-паразитов, взаимно не исключающих друг друга; например, вполне может существовать изваяние мифического животного.

5. Пространственные образы

5.1. Местоположение и ориентация

Обычно мы представляем себе, что наше движение происходит в неподвижном пространстве: когда мы поворачиваемся, мир не вращается вместе с нами, когда мы продвигаемся вперед, мир не отступает. Сидя за письменным столом, я считаю, что видимая из окна река течет на север, хотя на самом деле она сильно отклоняется от точного направления на Северный полюс. Такое ощущение направления относится ко всей окружающей обстановке; тот же "север" существует в любом доме, своем и соседнем, а любой

неподвижный предмет также характеризуется своим направлением (ориентацией) в пространстве.

Кроме ориентации каждый предмет характеризуется определенным местоположением. Мы менее уверены в существовании каких-то связей между позициями, находящимися в разных комнатах. Частично это происходит от того, что **определение местоположения любого предмета всегда требует вычислений, тогда как установление связей между ориентированными объектами - дело более простое (в прямоугольных комнатах направления просто переносятся из одного замкнутого пространства в другое).**

В незнакомой обстановке одни люди ориентируются значительно легче, чем другие. Есть люди, которые постоянно сверяют свои ощущения с компасом и никогда не теряются в новом для себя городе. Лишь небольшая часть их умения ориентироваться базируется на правильном использовании данных о проделанных в ходе движения по улицам поворотах. Они используют разные средства: карты, тени, время дня, ориентиры (включая отражения от окон) и т. д. Вначале этот способ кажется громоздким, но на самом деле он не требует слишком больших усилий. Весь фокус состоит в том, что надо выработать в себе привычку подмечать и должным образом представлять подобные вещи.

Сформировавшиеся представления об ориентации объектов довольно устойчивы и их трудно менять, даже когда для этого имеются веские основания. Такие трудности указывают на" то, что нами используются как глобальные эталонные фреймы, так и более мелкие локальные структуры. Трудности при перестройке представлений свидетельствуют, что **локальные фреймы не являются полностью трансформируемыми структурами, а при уточнении межпредметных связей зависят от места их присоединения к глобальным фреймам**. Ниже рассматриваются некоторые вопросы использования глобальных эталонных систем: в принципе это должно предполагать наличие более мощных и общих процедур для частичного изменения сложных представлений; на практике возможности людей в этом плане сильно ограничены, особенно когда они находятся в условиях жесткого дефицита времени.

5.2. Глобальная система пространственных фреймов

***Глобальный пространственный фрейм* (GSF) представляет собой постоянный набор "типичных позиций" в абстрактном трехмерном пространстве, копии которого используются как каркасы для сборки сложных сцен.** Такой каркас можно представить себе в виде расположенной в горизонтальной плоскости решетки (матрицы) размером (5x5), каждый узел ("позиция") которого имеет три вертикальных уровня. Центральные ячейки служат для представления сведений, наиболее близких к основному в GSF понятию, а периферийные представляют собой менее значительные понятия. (В сущности, человек всегда представляет себе, что он находится в универсальной воображаемой комнате, в которой происходят реальные события). Люди, вероятно, в жизни своей используют более сложные и математически менее строгие структуры, например, чтобы подчеркнуть простоту доступа к объектам, находящимся вблизи рук или лица, или чтобы представлять пространство не в чисто метрических категориях, а по отношению к своим манипуляционным возможностям.

GSF связан с системой *видовых фреймов;* каждый видовой фрейм описывает визуальные характеристики GSF со своей "колокольни". Таким образом, этот подход не противоречит одновременно ни системе Коперника, ни системе Птолемея: перемещения наблюдателя никак не влияют на присутствие в GSF образа видимой им сцены, однако активация видового фрейма, соответствующего данному конкретному местоположению наблюдателя, представляет его воображению именно ту картину. которую он и должен видеть.

<u>Видовой фрейм</u>, соответствующий определенной позиции наблюдателя, получается путем проектирования на это место ячеек GSF. Результатом является массив *видовых перечней,* каждый из которых представляет собой упорядоченную последовательность тех ячеек GSF, которые должны пересекаться лучом, исходящим от наблюдателя. Таким образом, **видовой фрейм подобен обычному визуальному фрейму**, за исключением того, что его элементы получены из GSF, а не в результате наблюдений за отдельными элементами и связями реальных объектов. Поскольку видовые перечни соответствуют участкам сетчатки, нам они представляются

трехмерными зонами, вытянутыми вдоль одного общего для данного перечня направления.

Заслонения объясняются или представляются с помощью предписании для видовых перечней; нам не следует ожидать, что удастся увидеть целиком тот объект, который не находится на первом месте в видовом перечне. (Аналогично, более близкие к началу перечня предметы препятствуют выполнению манипуляций с другими предметами, находящимися дальше в этом списке. Заслоненные ячейки перечней видов предоставляют процессу согласования большую свободу, ибо они устраняют часть ограничений соответствующих терминалов.)

Для усвоения визуальной информации, полученной из разных точек наблюдения, нам необходимо нечто наподобие схемы "косвенной адресации", в которой визуальные черты приписываются видовым перечням посредством каркасных конструкций GSF. Ниже приводится набросок такой схемы.

ЗРИТЕЛЬНОЕ ВОСПРИЯТИЕ. Разнообразные типы визуальных "черт" распознаются с помощью демонов на уровне сетчатки. Каждая обнаруженная черта автоматически связывается с видовым направлением текущего видового перечня в соответствии с его положением на общем визуальном поле.

АКТИВАЦИЯ ФРЕЙМОВ. Одновременно производится предварительное присоединение некоторого предметного фрейма или некоторого вида ожидания к узлам решетки GSF в соответствии с данным направлением текущего видового перечня. Это означает, что каждый терминал связывается с видовым направлением некоторого вшивного видового перечня. (Иными словами, терминалы визуального фрейма содержат пространственно-сориентированные данные, что оказывается возможным благодаря наличию соответствующих указателей в структуре GSF). В рамках одной системы различные визуальные фреймы выбираются согласно текущему видовому фрейму, а направления всех объектов должны быть соответственно откорректированы.

КОНКРЕТИЗАЦИЯ. Когда мы смотрим в определенном направлении, то, во-первых, в соответствии с информацией активного фрейма ожидаем увидеть определенные визуальные черты в определенных зрительных областях, соответствующих данным GSF, и, во-вторых, *на*

самом деле видим их там. Поэтому естественно предложить такую теорию зрительного восприятия (первого порядка), в которой каждый маркер каждого терминала фактически задает некоторый класс **визуальных демонов - "<u>признаков</u>"** так же, как и предполагаемое местоположение соответствующего узла в GSF. В такой системе наблюдатель может быть тоже представлен как объект и это позволит ему "увидеть" себя из разных мест в качестве полноправного элемента сцены. При наличии всего этого довольно легко получить информацию, требуемую для означивания терминалов и конкретизации фреймов. Системе остается лишь сопоставить "перцептуальные" пары (демон, видовой перечень) со "схематическими" парами (маркер, узел). Если бы терминалы предметных фреймов можно было непосредственно присоединять к узлам GSF, и если бы они автоматически проектировались, и образовывали видовые перечни, то это бы почти полностью избавило систему от необходимости проведения повторных вычислений для представления тех объектов, которые уже наблюдались, но только из других положений.

5.3. Совершенствование системы

В первой формулировке предполагалось, что терминалы визуальных фреймов связаны некоторым образом с узлами каркаса GSF. В этой связи возникает вопрос: почему бы не отказаться от всей идеи создания системы визуальных фреймов и не построить **трехмерные предметные фреймы**, которые непосредственно трансформировались бы в определенные пространственные позиции? В этом случае **предметный фрейм** почти без всяких ухищрений мог бы представлять **трехмерную символьную структуру**, а GSF-система автоматически порождать различные видовые фреймы для любого объекта.

Для систем, ориентированных на ЭВМ, это могло бы принести хорошие результаты, но для психологической модели породило бы слишком много серьезных проблем: каким образом можно справиться с трансформациями, поворотами и изменениями масштаба; как следует проводить переориентацию субструктур и др. Для моделирования поворотов первое и весьма несовершенное решение может состоять в том, чтобы каждый объект характеризовался небольшим числом стандартных видов с указанием различных размеров и ориентации. Прежде чем отвергать эту идею, отметим, что она может быть весьма

полезной для представления некоторых видов действий, а также при моделировании действий на их предварительных этапах.

Поскольку, однако, образ любого предмета базируется на опыте его использования в различных ситуациях, требуется, по всей видимости, некоторый более общий тип операций, основанных на трансформациях. Представление изменений в местоположении и масштабе может быть выполнено на основе следующей промежуточной структуры: каждый предметный фрейм следует включить в некоторый пригодный для изменения местоположения "портативный" мини-GSF, который можно поворачивать и присоединять к любому узлу глобального GSF с соответствующими "примечаниями", указывающими, каким образом трансформирован исходный образ. Наличие такой структуры влечет за собой не просто усложнение самой операции встраивания. Оно требует наличия в GSF "однородных структур"; это позволит упорядочить прежние, полезные, но идиосинкразические преувеличения, касаемые всего того, что расположено вблизи основного пространства, и потому нам более всего знакомого. Как бы привлекательна ни была подобная модель, верится с трудом, что она реально существует в механизмах человеческого восприятия. Люди не очень хорошо представляют себе различные видоизменения сцен.

Итак, у нас имеется ряд теоретических механизмов пространственного видения. Не будем выделять какой-либо из них. Дело здесь в важности того положения, что каждый индивидуум, очевидно, развивается с помощью последовательных все более и более усложняющихся механизмов. Нам следует уяснить, какие механизмы восприятия будут достаточны для различных уровней манипуляции зрительными образами; только после этого можно ожидать появления теории, совместимой с отмеченной концепцией развития. Следует также позаботиться и о том, чтобы располагать значительно более полной и точной психологической картиной, указывающей, как же в действительности используются пространственно-визуальные образы.

Некоторые читатели могут поинтересоваться, почему, подойдя столь близко к вопросам создания трехмерного аналогового механизма, не сделать это некоторым более простым, изящным и систематизированным способом. Такое предложение весьма естественно, но следует заметить, что из тех, кто стремился найти практическое решение подобного типа, еще никто не продвинулся

дальше первых и несовершенных гештальт-моделей. Можно себе представить также и нейронную структуру несимвольной трехмерной системы, однако, проблемы построения на ее основе предполагаемых образов твердых тел вновь заставят нас решать те же нетривиальные с вычислительной точки зрения и в основе своей символьные вопросы. Поэтому представляется неизбежным создание некоторого аналога рассмотренного ранее видового перечня, а это ставит под сомнение саму цель организации промежуточной аналоговой пространственной модели.

5.4. Эволюция

Теория фреймов предполагает наличие большого числа разнообразных механизмов для манипуляции визуальными и символьными образами. Видимо, что большинство этих механизмов не может возникнуть в процессе самоорганизации системы; скорее, они зависят от того, что было заложено в систему с самого начала. Какие этапы в эволюционном развитии способствовали появлению подобной первоосновы? Приведенные ниже доводы показывают, что совершенствованию фреймо-представлений в целом, видимо, способствовали требования пространственного зрительного восприятия.

На ранних стадиях эволюционного развития узловые моменты были, видимо, связаны с совершенствованием детекторов отдельных визуальных черт, что диктовалось необходимостью в удовлетворении первых жизненных потребностей (питание, воспроизводство, самооборона). По мере того, как все более сложными становились зрительная и двигательная системы, росли требования к правильному установлению отношений между видимыми предметами и их местонахождением во внешнем мире, т. е. между предметами и теми позициями, которые можно достигнуть или к которым можно подойти. Особо нужными становились те преобразования, которые позволяли бы компенсировать изменения в своем собственном местоположении. Это было важно, например, на охоте или в каких-то иных критических условиях. На охоте или во время полета определенным преимуществом обладает тот, кто способен координировать информацию, получаемую во время своего движения; если даже видение все еще базируется на последовательном распознавании простых визуальных черт, то и в этом случае способность к правильному объединению различных

признаков, замеченных в разное время, предоставляет субъекту определенные преимущества.

Простое, линейное, горизонтальное упорядочение визуальных черт позволяет создать большое число полезных схем "распознавания". Еще большего можно достигнуть, если пользоваться данными, получаемыми, во-первых, в процессе движения объекта относительно наблюдателя и, во-вторых, как результат изучения двигательного параллакса. Вследствие этого нам **нужно заниматься как схемами распознавания на базе согласования линейных фреймов с отдельными частями упорядоченных совокупностей, так и схемами объединения для выработки и развития представлений (пусть даже несовершенных) об окружающем нас мире.** Не следует думать, что мы сразу получим глобальную картину мира; вначале мы будем располагать эгоцентрическими полярными представлениями, основанными на связях между парами различных объектов или между объектом и каким-либо опорным направлением, скажем, направлением на солнце. На ранних этапах, по-видимому, еще не должны существовать усложненные механизмы для анализа связей типа "фигура-фон" и построения трехмерных сцен. Пока не известны какие-либо серьезные доказательства того, что живые существа, помимо человека, могут вырабатывать реальные представления о том, как устроен наш мир, и, хотя по поведению отдельных животных это можно было бы предположить, таким фактам можно дать более простое толкование.

Построение и использование глобальных представлений требуют развития тех же трансформаций движения, которые необходимы для задания соответствующим узлам различных видовых данных. Чтобы объяснить, в свою очередь, эволюцию таких механизмов, нужно попытаться представить себе возможные пути этого развития, начиная с эгоцентрического углового пространства, которое способствует реализации визуально-моторной деятельности. Построение системы фреймов на базе общих терминалов - близкая и несколько более простая задача; для ее решения могут оказаться полезными те схемы и структуры, о которых сейчас идет речь. Чтобы создавать другие варианты памяти для хранения зрительных образов, нужно разработать способы включения совокупностей заданий в элементы долговременной памяти: кто-то, например, хочет получить представление о квартире товарища, кто-то - о местах гнездования птиц или районах хорошей охоты и т. д. Было бы ценным получить

нужные и интересующие вас как охотника сведения для уверенной ориентации в том районе, где намечается охота.

Потребности зрительного восприятия отчетливо указывают на необходимость проведения определенных манипуляций с символьными представлениями, основанными на теории фреймов, однако же, они в полной мере не обусловливают необходимости в механизмах воображения. Последние весьма полезны в любой деятельности, направленной на решение задач и требующей использования принципов планирования.

Нам следует разделять индивидуальное и эволюционное развитие. Текущий видовой фрейм взрослого человека обычно определяется тем, что он знает о своем местонахождении; это требует учета всех поворотов тела, поворота головы и направления взгляда. Было бы неудивительно обнаружить в лобных участках коры головного мозга "врожденные" механизмы, ответственные за зрительное восприятие, т. е. механизмы, с помощью которых параметры, характеризующие положение человека в пространстве, управляют переадресацией демонов характерных визуальных признаков. Гипотеза о врожденных механизмах подтверждается хорошей визуально-моторной координацией, наблюдаемой в раннем возрасте у многих видов позвоночных. Однако существование иных механизмов *индивидуального* развития человека могло бы уменьшить требования к формированию врожденных механизмов зрительного восприятия.

Такая система восприятия, характерная для взрослых людей, может рассматриваться как система Коперника, тогда как у детей следует ожидать присутствия иных схем. У ребенка развитие системы зрительного восприятия начинается, вероятно, с той схемы, в центре которой находится лицо (а не ноги) и главная функция которой состоит в том, чтобы увязать зрение с движением рук. После этого наступает очередь создания несовершенного еще образа двигательных возможностей своего тела, и лишь значительно позже появляется глобальная система с "постоянным" чувством ориентации, в пределах которой наблюдатель мысленно может свободно перемещаться. Подобная эволюция системы, в центре которой последовательно располагаются голова, тело и затем свой собственный пространственно-ориентированный образ, требует очень больших усилий, и поэтому у ребенка этот процесс значительно растянут во времени. Такой процесс, но в значительно более ограниченных

масштабах, можно было бы, по всей видимости, изучить, наблюдая за тем, как люди учатся ориентироваться по карте. Вначале человеку необходимо сопоставлять карту с видимой картиной, затем это становится все менее и менее необходимым. Искусство, вероятно, заключается в том, чтобы представлять себе и картину, и карту одинаково ориентированными относительно некоторого внутреннего направления, например на север. Часть появившихся в процессе тренировки новых способностей состоит в том, что человек по мере стабилизации и уменьшения амплитуды колебаний между теми или иными возможными решениями улучшает механизм перспективных преобразований данных на основе отбора наилучших для каждой конкретной ситуации ориентиров.

В любом случае наша задача состоит не столько в том, чтобы обосновать преимущества "врожденной" или "развиваемой" модели восприятия, а, скорее, в том, чтобы построить хорошие сценарии возможных действий промежуточных систем. Относительная беспомощность младенца по отношению, скажем, к жеребенку, конечно же, не означает, что у него отсутствует врожденный пространственно-моторный механизм; это свидетельствует о том, что его проявление задерживается до тех пор, пока не появятся предпосылки для образования образов и построения на этой основе более сложной в интеллектуальном плане системы.

5.5. Вопросы измерений и количественных оценок

Большинство людей ощущают противоречие между объяснением мыслительных процессов с помощью дискретных символьных описаний и естественными представлениями, в которых наш внутренний мир постоянно связан с понятиями различной интенсивности (краски, усилия и др.), т.е. понятиями со свойствами непрерывности. В этой области выявление истины с помощью самоанализа или интуиции пользы не приносит. Видимо, что **символьные модели являются более глубокими по своим возможностям, тогда как (и это может показаться парадоксальным) непрерывные структуры ограничены и могут явиться тормозом при проведении исследований.** Мы уже проиллюстрировали эту точку зрения на примере оценочных функций в шахматах. Разумеется, аналоговая техника весьма полезна. Однако многие аналитики недооценивают мощь знаковых систем. Их стремление к непринятию идеи символьного описания объектов и

явлений проистекает из чувства "непрерывности" сознания: не следует ли нам не замечать каких-либо гипотетических процессов, в которых одно символьное описание внезапно исчезает, чтобы уступить место другому. **Такое непрерывное представление не может обладать реальной силой, ибо существен только тот процесс, в котором может быть отражено, зафиксировано и проверено то, что уже произошло.** Точно так же, как наша способность к отладке программ для вычислительных машин зависит от характера и качества соответствующих проверок, самосознание должно зависеть от прежних состояний человека и выработанных для них итоговых оценок. В этом случае феноменологически "гладкая" или "грубая" последовательность психических состояний должна отражать только стиль описания, используемого для представления этой последовательности.

Точные количественные измерения параметров могут лежать в основе выполнения различных прогнозов в системах роботов, связанных с вычислительными машинами. При работе над созданием теории восприятия зрительной информации человеком нам следует уяснить, **насколько хорошо качественные символьные методы могут имитировать наши способности к воображению и манипуляции образами.** Люди очень плохо воспринимают абсолютные значения размеров, расстояний и интенсивностей; они не могут с достаточной точностью устанавливать принадлежность размера, громкости, высоты тона, массы к одной из, скажем, десяти категорий. При сопоставлении различных суждений друг с другом многие заключения, для которых, казалось бы, требуются числовые данные, во многом предопределяются наличием простой упорядоченности величин. Рассмотрим три предмета A, B и C, видимые последовательно на фоне центральной стены комнаты. Если мы сдвинемся вправо к обнаружим, что B теперь находится левее A, то сделаем вывод, что B расположен ближе к прежней точке наблюдения и его надо задать как элемент переднего плана. В подобных "грубых" рассуждениях может содержаться даже большее количество информации, если пользоваться данными о расстояниях между линией воображаемого перемещения наблюдателя и объектами сцены. Таким образом, человеку вряд ли часто требуются точные количественные данные: дифференциальные измерения вполне подходят для близлежащих предметов, тогда как более общие суждения достаточны для тех объектов, которые находятся на значительных расстояниях друг от друга. Для большинства практических целей достаточно установить небольшое число связей между соседними предметами. Число их не должно увеличиваться быстрее, чем растет число предметов; если два предмета

находятся у противоположных стен комнаты, этот факт следует представить во фрейме "комната" верхнего уровня и этой информации человеку обычно вполне достаточно; если два предмета расположены вблизи друг друга, это отмечается в менее крупном фрейме, который содержит и другие данные о связях между этими двумя объектами. Таким образом, будет правильно полагать, что человеку трудно вспоминать взаимное расположение предметов, информация о которых содержится в разных фреймах, поскольку это требует поиска дополнительных данных, которых нет в памяти, а это всегда сложно и утомительно.

Против схемы GSF имеется ряд существенных возражений. В самой природе перспективы заложено, что любой близлежащий элемент будет заслонять ряд более удаленных элементов; в тех случаях, когда невидимой будет граница объекта, картина станет еще менее ясной, ибо нельзя будет сказать точно, какие части удаленного предмета от нас заслонены. (Поэтому идея видовых перечней не совсем хороша, если, впрочем, обратиться к вопросам человеческого воображения, то проблемы здесь будут те же самые.) Чтобы улучшить свойство предвидения, присущее системе, **видовые перечни можно преобразовать в видовые структуры с целью представления специальных отношений, более сложных, чем пары вида "ближе - дальше"**. Можно полагать, что измерительные возможности данной системы можно значительно улучшить, используя "символьную интерполяцию". Если рассмотреть совместно или последовательно видовые перечни двух (или более) близких друг другу позиций, то можно отыскать компромиссный вариант для согласующихся в отдельности прогнозов. Использую движение (изменение точки наблюдения), человек, таким образом, может значительно точнее определять невидимые границы предметов.

Эта идея интерполяции или - в своей простейшей форме - суперпозиции во многих случаях позволяет улучшить общую применимость используемых стратегий. Усреднение или иное комбинирование прогнозов приводит к получению лучших, нежели можно того ожидать, результатов. Следовательно, расчеты для манипуляций образом тела (которые, видимо, требуют проведения сложных векторных и матричных преобразований) могут выполняться путем суммирования ожиданий или прогнозов, исходящих от достаточно близких к требуемым "стереотипных положений". Весьма перспективно распространить это правило на абстрактные виды

деятельности, например на процессы, которые могут многократно использовать символьные представления.

Поиск и извлечение информации из памяти - еще одна область, где важны, по крайней мере, на первый взгляд, количественные методы. Здесь нужны механизмы для управления допустимым диапазоном изменения заданий терминалов. Что лучше: принцип "полного согласования", использование некоторого порога пригодности или что-либо еще? Ни одна стратегия в отдельности не принесет желаемых результатов. Рассмотрим следующее высказывание:

"Возьмите этот самый большой красный кирпич". Чтобы уяснить смысл слова "самый большой", надо сопоставить различные по своим размерам тела. Если для подобных целей разработать одну неизменную процедуру, она сможет правильно работать лишь в простых ситуациях. Поэтому следует обратиться к цели решения стоящей перед нами задачи. Если кого-то интересует масса, то следует принять, что *самый большой* - это *самый тяжелый.* Если человек придерживает окно и для этого ему нужен шест, тогда самый большой - это *самый длинный.*

Положение может сильно осложниться, если выбор •предмета не будет оговорен в тексте: *"Возьмите какой-нибудь большой красный кирпич".* В этом случае следует использовать те же принципы; разделить мир на классы, уместные в данной ситуации, затем взять из этого класса то понятие, которое наиболее подходит к слову "большой". Обычно слово "большой" означает "самый большой", но это правило не применимо в том контексте, где употребляется слово "громадный". В последнем случае следует, определив цель высказывания, произвести выбор соответствующего метода группировки понятий и далее действовать аналогичным образом. Количественные признаки и здесь могут найти себе применение, но они, естественно, будут подчиненными, второстепенными, ибо в противном случае можно упустить наиболее важные аспекты рассматриваемой проблемы. Д.Макдермотт рассматривает многие вопросы, касающиеся дискретных представлений пространственных структур. В его работе приведено довольно много различных аргументов против использования количественных моделей. Каждый из них в отдельности не слишком весом, и поэтому, видимо, следует остановиться на тех общих положениях, которые лежат в основе негативного в целом отношения, к количественным моделям. Исходный тезис таков: выходные данные такого механизма независимо от того, являются ли они цифровыми,

аналоговыми, физическими (несимвольными) или статистическими, слишком бесструктурны и неинформативны, чтобы на их основе можно было проводить дальнейший анализ. Данные в виде чисел позволяют принимать решения о немедленном выполнении каких-то действий или мускульных сокращений, о выделении и объединении стимулирующих признаков и т. д. Но поскольку каждое такое **данное** по природе своей **является оценкой, а не резюме**, то для целей планирования и проведения дальнейших рассуждении все они непригодны. В числовом показателе нельзя отразить те соображения, на основании которых он был получен. Поэтому, **хотя количественные результаты полезны для достижения непосредственных целей, они во многом ограничивают возможности дальнейшего и более глубокого развития систем.**

Это, конечно, не означает, что люди не должны и не пользуются такими методами. Учитывая, однако, те препятствия, которые они создают для проведения дальнейших рассуждений, мы можем сказать, что количественные методы будут концентрироваться и использоваться функциональными элементами типа *терминалов.* В более общем плане они могли бы обусловливать деятельность, наиболее близкую к бихейвористическим концепциям, и это могло бы в какой-то мере объяснить традиционный интерес ученых этого направления к количественным методам исследований. Опасность заключается в том, что основанные на них теории, пригодные для расчета вероятностных характеристик, составления расписаний и др., видимо, не способны объяснить сложную познавательную деятельность. В качестве психологических теорий они, по-видимому, не могут не быть ошибочными.

5.6. Критика логистического подхода

"Когда кто-либо пытается описать реальные мыслительные процессы с помощью традиционно-формальной логики, результат часто оказывается неудовлетворительным; в этом случае, несмотря на существование целого набора правильных операций, теряется смысл процессов, а то, что было жизненно важным, значительным, творческим как-то исчезает из формулировок".

М.Вертхаймер

Ниже будет приведено разъяснение, почему, по мнению М.Мински, большинство логических подходов малоэффективными.

Еще со времен Аристотеля предпринимались серьезные усилия, чтобы представить рассуждения с помощью **логистической системы, т.е. такой системы, в которой полностью отделены друг от друга предложения, которые содержат конкретную информацию, и силлогизмы или общие законы построения правильных высказываний.** На протяжении веков никому не удалось продемонстрировать успешное функционирование такой системы на реальном и значительном множестве высказываний. М. Мински полагает, что подобные попытки будут и впредь заканчиваться неудачно, но не вследствие дефектов логических формализмов, а из-за самого характера логистического метода. (Последние опыты базировались на логике предикатов первого порядка, но не в ней корень зла).

Типичная попытка имитации рассуждений на уровне здравого смысла начинается с выбора "микромира" ограниченной сложности. С одной стороны, задаются цели высокого уровня, например: "Я хочу попасть из дома в аэропорт"; с другой - множество небольших высказываний - аксиом, например: "Автомобиль находится в гараже", "Никто не выходит из дома раздетым", "Чтобы попасть в какое-то место, человек должен двигаться в том направлении" и т. д. Для работы системы используются процедуры эвристического поиска, с помощью которых должна быть доказана достижимость поставленной цели и найден соответствующий перечень необходимых действий.

Не будем останавливаться и анализировать историю всех имевших место попыток получить из набора аксиом целевые высказывания, а приведем по этому вопросу такое суждение: в простых случаях можно добиться, чтобы подобные системы могли "действовать", но по мере приближения выбранного микромира к реальному трудности становятся непреодолимыми. Проблема поиска подходящих аксиом или, иначе, проблема "задания фактов" на основе всегда логически правильных допущений оказалась значительно более трудной, чем это ранее предполагалось.

ФОРМАЛИЗАЦИЯ ЗНАНИЙ. Главной проблемой в изучении интеллекта является проблема создания основы знаний. Мы слишком мало знаем о содержании и структуре обычных знаний независимо от того, какую цель преследуют наши исследования: создание логистических систем или что-либо иное. **Самая простая система здравого смысла должна быть информирована о таких категориях,**

как причина и следствие, время, цель, местоположение, процесс, вид знаний; ей также необходимы сведения о том, как приобретаются, представляются и используются знания. В этой области необходимы серьезные эпистемологические исследования. Работы Дж.Маккарти и Е.Сандуолла ценны именно в этом плане.

РЕЛЕВАНТНОСТЬ. Ключевой является проблема выбора из избыточного множества релевантной информации. Современная эпистемология во многом отличается от прежних теорий познания. **Необходимы новые и общие представления о вычислениях**. **Наиболее ценная по своему характеру часть знаний не может передаваться нам извне, скорее, она внутренне должна быть доказана.** Для каждого факта человеку требуются метафакты, сообщающие о том, каким образом и когда их можно использовать. В пределах ограниченного микромира можно установить способы взаимодействия между **ситуациями, действиями и случайными явлениями.** И хотя данная система сможет на базе заданных аксиом выполнять дедуктивные построения, она не сможет определить, когда ей следует это делать, а когда нет. Например, человек может пожелать сообщить системе следующее: "Не переходи дорогу, если приближается автомобиль". Но он не может потребовать того, чтобы система доказала, будто автомобиль не приближается, поскольку подобное доказательство обычно будет совсем не тем, что нам нужно. Системе PLANNER (С.Хьюитт) можно дать указание попытаться доказать, что автомобиль приближается, и сообщить, что только в том случае, если эта (ограниченная) попытка дедуктивного вывода окажется безуспешной, можно переходить улицу. Чисто логистическая система ничего подобного сделать не позволяет. Первой реакцией должно быть: "Посмотреть налево, посмотреть направо". Но если сообщить системе данные о скоростях, тупиковых переулках, вероятностях обгона на повороте и др., доказательство становится необозримым и потому невозможным. Нам следует представить и сделать понятным системе слово "обычно". В конечном счете, потребуется понять компромисс между гибелью и деятельностью, ибо нельзя ничего сделать, будучи парализованным страхом.

ЕДИНООБРАЗИЕ. Даже сформулировав ограничения на использование релевантной информации, в логистических системах нам все равно придется столкнуться с проблемой её правильного использования. В таких системах все аксиомы обязательно должны быть "разрешенными", ибо с их помощью вырабатываются новые заключения. Любая дополнительная аксиома ведет к появлению новых

теорем, и поэтому ни одну из аксиом потерять нельзя. Вся сложность в том, что *нет явного способа указать системе, какие выводы следует делать, а какие - не следует.* Если мы зададим достаточно аксиом, чтобы на их основе вывести все требуемые нам следствия, то, кроме того, мы докажем значительно большее число других вещей. Если, однако, попытаться изменить это положение, задав ряд аксиом о релевантности данных, то это приведет лишь к росту числа нежелательных теорем: к старым добавятся такие теоремы, которые будут содержать утверждения относительно их нерелевантности.

Логиков обычно интересуют сами процедуры доказательств, они не обращают внимания на возможный рост дедуктивных систем и поэтому могут получать те утверждения, которые их интересуют. При развитии интеллекта ситуация будет иной. Субъект должен научиться определять, во-первых, какие из признаков в каждой ситуации основные, а какие нет, и, во-вторых, какие виды дедукции не должны восприниматься слишком серьезно. Обычная реакция на рассказы лгунишки смех, из чего следует сделать вывод, что отклонять следует не его исходную посылку-аксиому, а его дедуктивные построения. В этой связи возникает следующая проблема.

ЗНАНИЯ ДЛЯ УПРАВЛЕНИЯ ПРОЦЕДУРАМИ. Отделение аксиом от процесса вывода делает невозможным использование классифицированных знаний об имеющихся в системе утверждениях или фактах. Мы также не можем включить в нее знания об управлении процессом дедукции. Проблема состоит в том, чтобы аксиоматизировать наши представления об аппроксимации и близости объектов друг к другу. Человеку привычно свойство транзитивности, скажем:

(А около В) /\ (В около С) => (А около С),

но неограниченное применение такого правила приведет к тому, что все предметы окажутся расположенными по соседству друг с другом. Можно применить нечто вроде технической шутки:

(А (около)^1 В) /\ (В (около)^1 С) => (А (около)^2 С),

допустив при этом, скажем, только пять степеней для понятия "около": около, (около)^2, ... (около)^5. Можно изобрести какие-то аналоговые величины или параметры. Но в логистической системе нельзя

ограничиться применением, например, трех правил транзитивности подряд, если на то нет серьезных оснований. Пока затруднительно ответить, как же следует разрешить эту проблему, и, по имеющимся сведениям, никто еще не предложил в этом плане чего-либо делающего ему честь. Следует отметить тот факт, что, поскольку логистический подход достаточно распространен, никто непредвзято не исследовал подобный тип процедурных ограничений.

КОМБИНАТОРНЫЕ ПРОБЛЕМЫ. Логическим системам, видимо, не удастся избежать комбинаторного взрыва в том случае, если будет найдена возможность представления более обширных знаний. Хотя время от времени мы получаем сведения об успешной работе подобных систем в ограниченных микромирах, следует иметь в виду, что для исследований по искусственному интеллекту это обычная ситуация: система высокого качества, решающая трудные головоломки, часто оказывается непригодной для работы в более крупных проблемных областях.

СОВМЕСТИМОСТЬ И ПОЛНОТА. В процессе своей умственной деятельности человек критически оценивает имеющиеся у него планы и перечни целей, пересматривая свои знания и правила их использования. Некоторые из этих действий можно непосредственно внести в саму программу доказательства теорем и использовать их для последующего самоанализа, но человек в действительности хочет представлять их себе более естественным образом, в виде свода декларативных правил. Почему же тогда ученые стремятся, чтобы именно логистические системы выполняли эту работу? Действительная причина заключается в том, что такие системы весьма просты и элегантны; если бы они еще были и эффективны, было бы просто замечательно. Чаще указывают на другую причину, неверную по своей сути, именно, что подобные системы математически строги, поскольку они обладают свойствами:

(1) **полноты**, т.е., "**можно доказать все истинные утверждения**", и

(2) **совместимы**, т.е. "**нельзя доказать ни одно ложное утверждение**".

По всей видимости, люди часто не понимают, что **полнота** - это достоинство не такое уж редкое. **Оно является тривиальным следствием любой процедуры исчерпывающего поиска, поэтому**

всякая система может быть переведена в категорию "полных", если к ней подсоединить любую другую полную систему и после этого чередовать этапы вычислений. Совместимость - понятие более тонкое, оно предполагает отсутствие противоречивости в наборах аксиом. М. Мински считает, что в системах искусственного интеллекта подобного требования не следует придерживаться, ибо **ни одна система естественного интеллекта не является полностью совместимой.** Важно то, каким образом человек разрешает парадокс или находит выход из конфликтной ситуации, каким образом человек учится на своих и чужих ошибках, как распознает и отбрасывает всевозможные несоответствия.

Подобные неправильные представления привели к тому, что теорема неполноты Гёделя стимулировала появление совершенно беспочвенных утверждений о различиях между человеком и машиной. Никто, видимо, не заметил ее более "логичной" интерпретации, именно, что стремление к совместимости налагает определенные ограничения.

Конечно, есть и будут различия между людьми (которые доказуемо несовместимы) и машинами, конструкторы которых создавали их на основе этого принципа. **Но для машин вовсе не является необходимым программирование на основе только совместимых логических систем.** Те же философские рассуждения, которые выше не были нами приведены, но, тем не менее, подразумевались, использовали это ненужное допущение. (Полученные результаты, показывающие совместимость современной теории множеств, рассматриваются М. Мински не как доказательство потенциальной возможности ее использования в системах искусственного интеллекта, а, наоборот, как подтверждение ее вероятной неприменимости для наших целей.)

Когда одного известного математика предупредили, что, сделав еще один логический шаг в своем доказательстве, он придет к парадоксу, тот совершенно серьезно ответил: "А я не буду делать этот шаг". Значительная часть наших обычных (и даже математических) знаний напоминает знания людей - представителей опасных профессий, которые должны очень хорошо знать, когда и какие действия следует считать неразумными. В наших условиях нужно дать ответы на следующие вопросы: в каких случаях возможно применение тех или иных видов аппроксимаций; когда различные критерии могут

предопределить получение различимых оценок; какие утверждения и какие типы ссылок допустимы и другое. Концепции, основанные на свойстве транзитивности, представляют значительный интерес и от них вовсе не следует отказываться лишь потому, что еще не найдена удовлетворительная система аксиоматизации. Подводя итоги, отметим следующее.

1. Логические рассуждения недостаточно гибки и не могут служить основой для мышления; они представляются М. Мински в виде набора эвристических методов, эффективных только тогда, когда применяются к упрощенным схематическим планам. Совместимость, требуемая логикой, в иных аспектах обычно не обеспечивается и, вероятно, даже нежелательна, поскольку совместимые системы по своим возможностям будут, видимо, недостаточно мощными.

2. М. Мински сомневается в возможности эффективного представления обычных знаний в виде совокупности простых, независимых, "истинных" утверждений.

3. Стратегия полного отделения конкретных знаний от общих правил вывода слишком радикальна. Мы нуждаемся в разработке более непосредственных способов соединения фрагментов знаний, позволяющих дать совет, *каким образом* их следует использовать.

4. **Декларативная форма представления информации**, которую долгое время считали наиболее подходящей для проведения дедуктивных выводов, оказалась не столь уже необходимой, ибо мы нашли способы манипуляции структурными и процедуральными описаниями.

М. Мински не собирается утверждать, что *мышление* во многом может самостоятельно развиваться без чего-либо подобного *рассуждениям.* Мы, как говорит М. Мински, без сомнения, нуждаемся и используем элементы силлогистической дедукции, однако их применение должно подчиняться процессам "согласования" и "конкретизации", вызванным к жизни другими функциональными потребностями. **К традиционной формальной логике следует подходить как к <u>техническому инструменту</u> для уточнения всего, что может быть выведено из некоторого множества данных или для подтверждения того, что данное следствие можно получить совершенно определенным образом; формальная логика совершенно непригодна для**

обсуждения того, *какая* информация требуется и *что должно выводиться* при обычных обстоятельствах. Подобно абстрактной теории синтаксиса **формальная логика нуждается в мощной процедуральной семантике**, без которой она попросту бессильна в сложных проблемных ситуациях.

Не следует категорически утверждать, что принцип совместимости, столь важный для математической логики, оказал губительное влияние на исследования в области моделирования мышления. Однако в общем плане он привел к роковой концепции о потенциальных возможностях машин вообще. На "логическом" уровне были заблокированы попытки представления обычных знаний, ибо все работы предполагали поиск набора таких истин, которые бы не зависели от контекста и были почти всегда сами по себе справедливы. На уровне моделирования интеллекта был задержан процесс осознания того факта, что *мышление всегда начинается с наводящих на мысль, но несовершенных планов и образов, которые (если это вообще имеет место) постепенно совершенствуются и заменяются лучшими вариантами.*

6. Суть проблемы представления знаний

Проблема представления знаний является тем ключевым пунктом, через который проходят пути к достижению успеха, пожалуй, во всех направлениях исследований по искусственному интеллекту, начиная от проблем понимания естественного языка и кончая проблемами машинного восприятия зрительных образов и речи.

Очевидная первопричина сложности создания машинной модели реального мира кроется в бесконечном многообразии этого мира. Действительно, представим себе, что такую модель необходимо создать для робота, предназначенного для выполнения неограниченного набора приказов человека и функционирующего в среде неограниченной вариативности. Если попытаться решить задачу "в лоб", путем представления модели мира в виде набора программ. каждая из которых соответствует последовательности возможных действий робота при выполнении одного из приказов, осуществляемых с учетом конкретных условий внешнего мира робота (состояния внешней среды), то возникают по крайней мере две непреодолимые трудности, обусловленные бесконечным многообразием реального мира. (Под условиями внешнего мира понимается не только мир,

воссоздаваемый сенсорами робота, но и априорные знания о закономерностях реального мира).

Первая из них связана с необходимостью наличия в памяти ЭВМ неограниченного набора программ, каждая из которых рассчитана на выполнение приказа при определенном состоянии внешней среды. Вторая трудность порождена необходимостью соотнести данный приказ оператора и состояние внешней среды робота с конкретной программой из этого неограниченного набора, ответственной за выполнение данного приказа в конкретной внешней ситуации, и выбрать эту программу из памяти за приемлемое весьма ограниченное время. Иными словами, **речь идет о машинном "понимании" за ограниченные отрезки времени языка приказов и "языка сенсоров", с помощью которых воспринимаются те особенности внешнего мира, которые важны для выполнения приказа.** И если в отношении "понимания" языка приказов можно несколько уменьшить трудности за счет, например, использования приказов однозначного толкования, то в отношении понимания "языка" сенсоров это сделать невозможно.

Приведенные трудности объясняют нереальность создания модели реального мира таким путем. Несмотря на это, такой подход всё же несет одно ценное свойство. По крайней мере, на интуитивны уровне ясно, что каждую из программ, ответственную за выполнение данного приказа, можно построить так, чтобы учесть все необходимые для выполнения приказа особенности внешнего мира робота, причем реального внешнего мира со всем его многообразием и сложностью, что невозможно при использовании известных подходов из-за принципиальной ограниченности средств описания внешнего мира, характерных для этих подходов.

Конечно, ограниченность машинной памяти не позволит иметь большого количества таких программ, но в случае маловариативных сред и невысоких требований к диапазону функциональных возможностей робота реально получение приемлемых технических решений.

Очевидный путь, позволяющий, по-видимому, несколько уменьшить трудности создания машинной модели реального мира, лежит через кардинальное уменьшение числа входящих в модель программ при сохранении общего объема знаний о мире, содержащихся в этой

модели. Это можно было бы осуществить, если бы удалось определенным образом упорядочить, структурировать программы, образующие модель. С этой целью, во-первых, полезно найти и выделить в разнообразных возможных действиях общие **универсальные фрагменты**, с помощью которых можно компоновать эти действия. Очевидно, из *m* таких универсальных фрагментов в пределе можно скомпоновать (1!+2!+3!+…+ m!) разнообразных действий. Если даже ограничить число используемых для описания действия фрагментов величиной *m*, т.е. имеет место колоссальный выигрыш в использовании памяти, тем больший, чем больше *m*. Во-вторых, необходимо стремиться так построить программы, входящие в модель реального мира, чтобы каждая из них была способна формировать широкий набор (в пределе бесконечный) разнообразных действий, варьируемых, например, в зависимости от характера информации, собираемой определенной группой сенсоров, или от модификации приказов.

7. Характерные особенности фрейм-подхода к проблеме представления знаний

Одним из возможных путей организации машинной модели реального мира является подход, развиваемый М.Минским. В соответствии с этим подходом знания о мире - **машинная модель реального мира** - должны быть представлены в памяти ЭВМ в виде достаточно большой совокупности определенным образом структурированных данных, представляющих собой стереотипные ситуации. Эти **структуры запомненных данных получили название "фреймы"**. В случае возникновения конкретной ситуации, например, необходимости совершить роботом, управляемым ЭВМ, определённое действие, воспринять с помощью сенсоров, связанных с ЭВМ, какой-то зрительный образ и т.д., из памяти ЭВМ должен быть выбран фрейм, соответствующий данному классу ситуаций и согласован с рассматриваемой конкретной ситуацией из этого класса путем изменения подробностей, т.е. путем конкретизации данных из набора, которые могут удовлетворить выбранный фрейм.

Так как фрейм можно представить себе в виде сети, состоящей из узлов и связей между ними, то каждый узел должен быть заполнен своим

"заданием", представляющим собой те или иные характерные черты ситуации, которой он соответствует. В общем случае во фрейме можно выделить несколько уровней, иерархически связанных друг с другом. **Узлы фрейма, принадлежащие к верхним уровням**, представляют собой более общие вещи, которые всегда справедливы в отношении предполагаемой ситуации. Эти узлы уже заполнены своими заданиями. Например, узел самого верхнего уровня фрейма обычно заполнен **названием ситуации**, т. е. **названием зрительного образа** (это может быть, например, "куб"), **названием действия** (например, "уборка комнаты"). **Узлы нижних уровней по большей части не заполнены своими заданиями. Такие незаполненные узлы называют терминалами. Они должны быть заполнены конкретными данными, представляющими собой их возможные задания в процессе приспособления фрейма к конкретной ситуации, из того класса ситуаций, который представляет данный фрейм.** Каждый терминал может устанавливать условия, которым должны отвечать его задания. **Простые условия устанавливаются "маркерами"**, которые могут потребовать, например, чтобы **заданием терминала было какое-то лицо, какой-то предмет достаточной величины, какое-то элементарное действие или "указатель" на какой-то другой фрейм**, представляющий собой другую, обычно более частную ситуацию и называемый субфреймом. Более сложные условия могут устанавливать связи между заданиями для нескольких терминалов.

Группа фреймов может объединяться в систему фреймов. Результаты характерных действий отражаются с помощью трансформаций между фреймами системы. Они используются, чтобы ускорить вычисления определенных видов при представлении типичных изменений одной и той же ситуации.

В случае зрительного образа различные фреймы системы описывают картину с различных точек наблюдения, а трансформация одного фрейма в другой отражают результаты перемещения из одного места в другое. Для фреймов невизуальных видов различия между фреймами системы могут отражать действия, причинно-следственные связи и изменения понятийной точки зрения. Различные фреймы системы используют одни и те же терминалы. Это важное обстоятельство, благодаря которому, в частности, экономится объем памяти ЭВМ, используемой для построения модели реального мира. Характерной чертой описываемого подхода является возможность использования различных видов прогнозов, ожиданий, предположений. **В соответствии с этим терминалы фрейма, выбираемого для**

представления ситуации, обычно уже заполнены заданиями, которые наиболее вероятны в данной ситуации. Эти задания называются "заданиями отсутствия".

Таким образом, фрейм может содержать большое число деталей, которые могут и не подтвердиться данной ситуацией. Задания отсутствия "непрочно" связаны со своими терминалами, поэтому они могут быть легко "вытеснены" другими заданиями, которые лучше подходят к текущей ситуации.

После того как выбран **фрейм для представления ситуации**, процесс согласования фрейма с данной конкретной ситуацией состоят в **нахождении таких заданий для терминалов фрейма, которые совместимы с маркерами терминалов**. Процесс согласования частично контролируется информацией, связанной с фреймом (в которую входит и информация относительно того, как действовать в случае появления необычных ситуаций, "сюрпризов"), а частично знанием текущих целей. Если выбранный фрейм не удается согласовать с реальностью, т. е. если невозможно найти задания для терминалов, которые соответствующим образом согласуются с условиями маркера, то происходит обращение к так **называемой сети поиска информации, с помощью которой соединяются между собой системы фреймов**. Эта сеть позволяет найти другие способы представления знаний о фактах, аналогиях и другой информации, которую можно использовать для согласования с реальностью.

Теория представления знаний с помощью фреймов, развиваемая М.Минским, претендует на объяснение ряда характерных особенностей человеческого мышления. По мнению автора, она позволяет охватить единой концепцией такие, казалось бы, разные теории, как понимание естественного языка, машинного "восприятия" зрительных образов, поиска решений, планирования. Она объединяет многие классические и современные идеи психологии, лингвистики, а также искусственного интеллекта. В частности, эта теория обобщает идеи, высказанные в ряде известных работ по искусственному интеллекту, например в работах А.Ньюэлла, Г.Саймона, в которых знания о мире представляются с помощью пространств подзадач, в работах Р.Шенка, Р.Абельсона, где модель мира представляется пространством "сценариев", наконец, в работах С.Пейперта и самого М.Минского, в которых предлагается подразделить знания на "микромиры".

В своей теории М.Минский не проводит границы между теорией человеческого мышления и теорией построения "думающей" машины (искусственного интеллекта). Он считает, что и процесс мышления человека основан на наличии в его памяти каким-то образом материализованного огромного набора разнообразных фреймов, с помощью которых человек осознает зрительные образы (фреймы визуальных образов), понимает слова (семантические фреймы), рассуждения, действия (фреймы-сценарии), повествования (фреймы-рассказы) и т.д. Процесс понимания при этом сопровождается выбором из памяти соответствующего фрейма, у которого терминалы уже заполнены заданиями отсутствия, и приспособлением его к текущей ситуации. Если это не удается, то из памяти выбирается новый более подходящий к ситуации фрейм. В случае, когда и этот фрейм не достаточно хорошо согласуется с ситуацией и поиски нового не приводят к удаче, происходит приспособление наиболее отвечающего ситуации фрейма, который был обнаружен в процессе поиска.

7.1. Фрейм - визуальный образ

В качестве простейшего примера, иллюстрирующего представление знаний с помощью фреймов, рассмотрим приведенную в работе М.Минского возможную систему фреймов для элементарного зрительного образа - куба. В соответствии с использованным в работе А.Гузмана символическим представлением тел правильной формы с помощью "областей" и "связей" между ними можно допустить, что результатом разглядывания куба является структура, подобная показанной на рис. 7.1а.

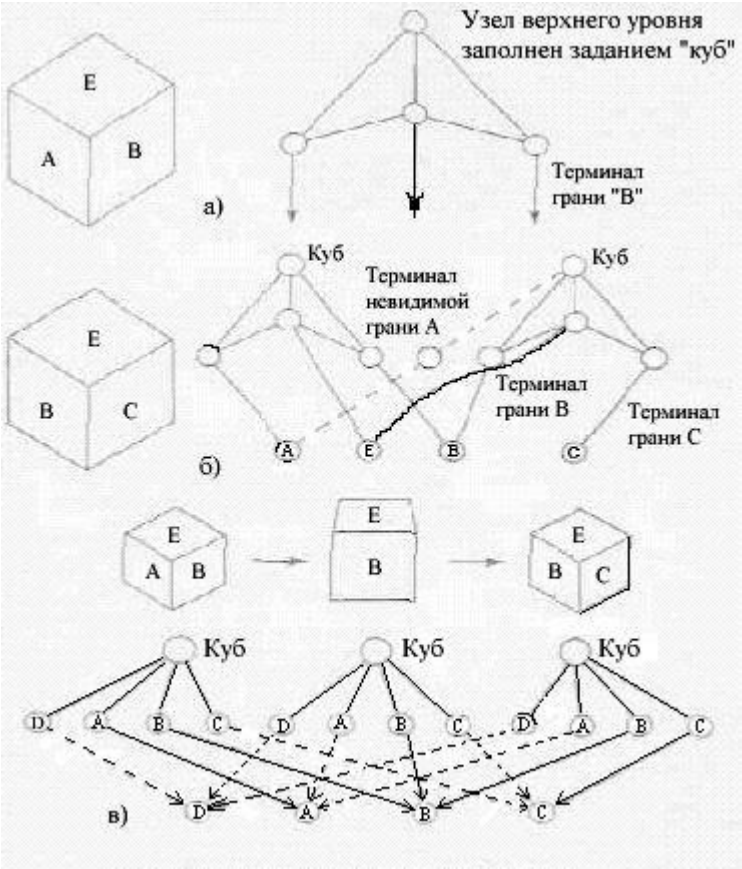


Рис. 7.1

Эту структуру можно идентифицировать с фреймом куба при разглядывании его с соответствующей позиции. Области A, E и B являются терминалами фрейма, задания для которых соответствуют возможным деталям или обозначениям на видимых с данной позиции гранях куба. Если позиция наблюдения куба перемещается вправо, то грань A исчезает из поля зрения и становится видимой грань C.

Если бы потребовалось провести полный анализ этого нового визуального образа, необходимо было бы:

1) утратить знания о грани А,
2) повторно воспринять (с помощью соответствующих "вычислений") образ грани В,
3) воспринять образ новой грани С.

Однако, поскольку известно, что произошло перемещение позиции наблюдения вправо, можно сохранить знания о грани *B* в виде задания терминалу левой грани нового фрейма куба, соответствующего новой позиции наблюдения. Кроме того, чтобы сохранить знания и о грани А, можно ввести дополнительный терминал невидимой грани, относящийся к этому новому фрейму, как это показано на рис.7.1б.

При возвращении на начальную позицию наблюдения оказывается возможным восстановить визуальный образ куба без каких-либо новых "вычислений". Для этого достаточно "вызвать" из памяти первый фрейм. Очевидно, полезно сохранить знания и о грани С, для чего можно ввести дополнительный терминал этой невидимой грани в первом фрейме (рис.7.1б).

Можно продолжить эту процедуру построения системы фреймов, соответствующую перемещению точки наблюдения вокруг куба. Это привело бы к получению более широкой системы фреймов, в которой каждый фрейм соответствует своей позиции наблюдения куба. На рис.7.1в показана система фреймов, состоящая из трех фреймов, каждый из которых представляет визуальный образ, получающийся в одной из трех позиций наблюдения. Две из этих позиций соответствуют перемещению вправо и влево на 45° относительно третьей позиции; указатели между фреймами соответствуют перемещениям точки наблюдения. Важно обратить внимание на выявленное в этом примере важное свойство представления ситуации с помощью системы фреймов. Оно состоит в том, что различные фреймы, входящие в систему, используют один и тот же терминал, соответствующий одной и той же физической черте, которая видна из различных позиций наблюдения. Это позволяет заранее сосредоточить в одном месте информацию о свойствах известных объектов независимо от позиций наблюдения, которых, особенно для предметов сложной формы, может быть очень много. В результате экономится память и сокращается процесс восприятия при изменении позиций наблюдения, так как память уже располагает необходимой информацией и время затрачивается лишь на "извлечение" ее из памяти.

Сами же системы фреймов, по-видимому, сформированы в памяти не для визуальных образов каждого возможного предмета, а для обычно используемых "основных форм", которые, вступая в различные комбинации, образуют системы фреймов для новых случаев. Это создает дополнительные возможности экономии памяти. Так же, как и в случае отдельных заранее сформированных терминалов, принадлежащих фрейму, наличие в памяти заранее заготовленного набора систем фреймов ускоряет процесс восприятия, так как новый образ не приходится строить заново, а только извлекать его из памяти и "приспосабливать" к действительности.

7.2. Фрейм-сценарий

Отмеченные свойства, позволяющие экономить память и время восприятия, очевидно, могут проявляться и при представлении невизуальных знаний о мире. Действительно, системы невизуальных фреймов, например описываемых ниже фреймов-сценариев, можно представить как одно из возможных понимании предмета обсуждения; например, электрогенератор можно представить как механическую и как электрическую системы. Любые знания о мире можно представить себе в виде некоторых общих универсальных фрагментов, которым соответствуют свои системы фреймов и из которых можно формировать новые системы фреймов, соответствующие некоторым новым представлениям.

Представление знаний о мире с помощью фреймов оказывается весьма плодотворным при объяснении механизмов понимания человеком естественного языка, рассуждений, повествований, наблюдаемых действий другого лица и т. д. По-видимому, оно будет полезным и при разработке искусственных механизмов реализации вышеперечисленных функций с помощью ЭВМ. В работе М.Минского в этом случае предлагается строить знания о мире в виде фреймов-сценариев. **Фрейм-сценарий по М.Минскому представляет собой типовую структуру для некоторого действия, понятия события и т. п., включающую характерные элементы этого действия понятия, события**. Например, фрейм-сценарий для события, состоящего в праздновании дня рождения ребенка, включает следующие элементы, которые можно трактовать как узлы фрейма, заполненные заданиями отсутствия:

Одежда: воскресная, самая лучшая;
Подарок: должен понравиться.

Для объяснения быстрого понимания человеком ситуации, представляемой сценарием, в работе Р.Шенка, Р.Абельсона предлагается отождествлять терминалы фрейма-сценария с наиболее характерными вопросами, обычно связанными с этой ситуаций. Ответы на эти вопросы полезно получить для понимания данной ситуации. По существу фрейм-сценарий в этом случае является собранием вопросов, которые необходимо задать относительно некоторой гипотетической ситуации, и способов ответа на них.

Для фрейма-сценария - дня рождения ребенка в число таких вопросов войдут следующие:
Что должны надеть гости?
Выбран ли подарок для ребенка?
Понравится ли ему подарок?
Где купить подарок?
Где достать денег? и т. д.

Для того чтобы попять действие, о котором рассказывается или которое наблюдается, человек часто вынужден задать такие вопросы:
"Кто осуществляет действие (агент)?",
"Какова цель действия (намерение)?",
"Каковы последствия (эффект)?",
"На кого это действие влияет (получатель)?",
"Каким образом оно произведено (инструмент)?".

Относительно понимания вещей, отличающихся от действий, задают несколько иные вопросы, и эти вопросы могут быть значительно меньше локализованы, чем в случае понимания действий, например: "Почему они говорят это мне?", "Каким образом я могу выяснить больше об X?", "Каким образом это поможет в решении проблемы?" и т.д. По рассказу спрашивают, какова тема, каково отношение автора, какое главное событие, кто является главным героем и т. д. По мере того, как на каждый вопрос дается предварительный ответ, из памяти могут вызываться новые фреймы, соответствующие ситуациям, которые возникают в результате ответов на вопросы. Вопросы - терминалы этих новых фреймов становятся в свою очередь активными.

Следует отметить, что число вопросов, связанных с фреймом неопределено, и на первый взгляд кажется, что для понимания ситуации их может быть очень много. Однако на практике оказывается достаточным задать весьма мало вопросов, чтобы разобраться в ситуации.

Разные люди могут задавать разное число вопросов относительно одной и той же ситуации. Число и характер этих вопросов в большой степени зависят от базы знаний относительно обсуждаемого объекта у того или иного индивидуума. Может оказаться, что полное понимание не будет достигнуто из-за отсутствия у человека необходимой системы фреймов, объединяющих знания об обсуждаемом предмете. Однако, когда необходимая база знаний существует, а относительно обыденных понятий, действий, рассуждений она есть у любого человека, то для понимания ситуации обычно достаточно не очень большое количество вопросов. Это весьма принципиальное обстоятельство, благодаря которому на интуитивном уровне мы приходим к весьма обнадеживающему для практики положению, состоящему в том, что большое количество реальных ситуаций, возникающих при понимании языка, рассуждений, действий можно понять, получив ответы на ограниченный круг вопросов. Это косвенно подтверждает возможность построения относительно простых моделей мира в данном случае с помощью фреймов, достаточных для принятия решений в этих ситуациях.

В случае фреймов-сценариев маркеры терминалов фрейма становятся более сложными, чем это было в случае фреймов визуальных образов, и определяют рекомендации относительно того, каким образом надо отвечать на вопросы, т.е., заполнять терминал заданием. Каждый терминал должен содержать рекомендации относительно того, каким образом найти его задание - ответ на вопрос. Задания отсутствия или перечень возможных ответов на вопросы являются самыми простыми особыми случаями таких рекомендаций. По-видимому, человек может иметь иерархический набор таких рекомендаций подобно схемам предпочтения, предложенным в работе Дж.Уилкса.

В соответствии с подходом М. Минского полное и всестороннее представление каждой ситуации типа событие, действие, рассуждение и т.п. подобно представлению визуального образа и осуществляется с помощью не одного, а системы фреймов. Каждый фрейм системы соответствует одной из возможных точек зрения на ситуацию,

представляемую системой фреймов, подобно тому, как один из визуальных фреймов системы представляет визуальный образ из некоторой одной точки наблюдения. Различные фреймы системы представляют различные пути использования одной и той же информации, находящейся на общих терминалах. Как и в "визуальной ситуации", человек, понимая или передавая мысль, "выбирает" один из фреймов. Этот выбор "по существу" состоит в выборе вопросов, которые, нужно задать относительно рассматриваемой ситуации.

В интерпретации Р.Шенка, Р.Абельсона **сценарий представляет собой последовательность действий, которые описывают часто встречающиеся ситуации.** В этой последовательности действий используется принцип **каузальной связи, т. е. результатом каждого действия являются условия, при которых может произойти следующее действие.**

Каждый сценарий имеет исполнителей ролей. Он имеет различные интерпретации, отражающие точки зрения различных исполнителей. Таким образом, его можно рассматривать как систему фреймов. Количество сценариев, отражающих возможные ситуации, встречающиеся в реальной жизни, огромно. Сюда относится и упомянутый выше день рождения ребенка и игра в футбол, занятия в школе и т. д. Ниже приводится сценарий посещения ресторана с точки зрения посетителя.

Сценарий: ресторан
Роли: посетитель, официантка, шеф, кассир
Цель: получить пищу, чтобы утолить голод

Сцена I. Вход
Войти в ресторан
Глаза направить туда, где есть пустые столы
Выбрать, где сесть
Направиться к столу
Сесть

Сцена II. Заказ
Получить меню
Прочитать меню
Решить, что хочешь заказать
Сделать заказ официантке

Сцена III. Еда
Получить пищу
Съесть пищу

Сцена IV. Уход.
Попросить счет
Получить чек
Направиться к кассиру
Заплатить деньги
Выйти из ресторана

В каждом сценарии средства выполнения действий могут варьироваться по обстоятельствам. Например, в сцене II заказ можно сделать письменно, устно и даже (в другой стране при незнании языка) жестами. В сцене IV выплата денег может быть осуществлена кассиру, официанту или словами "Включите в мой счет".

Возможно также, что обычная последовательность действий может нарушаться. Имеется по крайней мере **три характерных случая такого нарушения.** Первый - это **отклонение**, которое представляет собой прерывание последовательности действий сценария другим сценарием. Другие два случая называются препятствием и ошибкой. **Препятствие** имеет место тогда, когда кто-то или что-то мешает обычному действию ила отсутствует какое-то условие, необходимое для выполнения действия. **Ошибка** появляется тогда, когда действие завершается не так, как требуется. В принципе после каждого элементарного действия сценария могут возникнуть препятствия и ошибки, поэтому в сценарий вводятся различные наборы вопросов типа "а что если...", ответ на которые необходимо получать после каждого элементарного действия. При положительном ответе на один из них в сценарии предусматриваются новые действия, устраняющие препятствия и ошибки. Например, в сцене II сценария "ресторан", если официантка не замечает посетителя, он попытается встретиться с ней взглядом или окликнуть её.

Таким образом, **<u>сценарий</u> - не просто цепь событий, а скорее связанная каузальная цепочка действий**. Он может разветвляться на множество возможных путей, которые сходятся в особо характерных для сценария точках - элементарных действиях. Для сценария в

ресторане такими действиями являются "прием пищи" и "уплата денег".

Для того чтобы знать, когда пользоваться сценарием, нужны заголовки. Эти заголовки определяют обстоятельства, при которых обращаются к данному сценарию.

Из приведенного описания сценариев нетрудно установить аналогию между пониманием сценария по М. Минскому и по Р.Шенку. Действительно, как в том, так и в другом случае **сценарии описывают стереотипные, обычно встречающиеся ситуации.** Правда, в случае Р.Шенка сценариями охватывается несколько более узкий класс ситуаций, описываемый последовательностями действий. Как в том, так и в другом случае со сценарием связан определенный круг вопросов. Однако в случае Р.Шенка имеет место более конкретный и узкий круг вопросов типа "а что, если ...", а также вопросы, ответы на которые определяют выбор средств, определяющих действия сценария.

Поскольку в соответствии с высказанным ранее положением процесс понимания человеком реальной ситуации есть выбор из памяти и приспособление к этой ситуации соответствующего фрейма, возникает вопрос о возможном механизме этого акта.

8. Способ формализации фреймов

Одни из множества возможных способов формализации фрейма-сценария предполагает представление его в виде сети следующей иерархической структуры (рис.8.1).

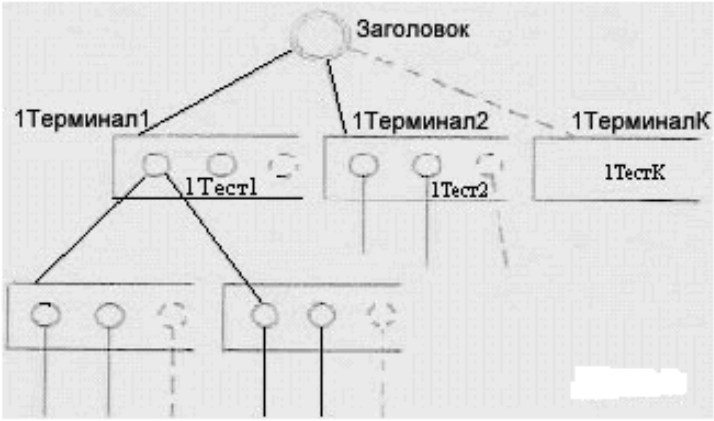


Рис. 8.1

Узел самого верхнего уровня сети (на рисунке ему соответствует кружок, обведенный жирной линией) отождествляется с заголовком сценария. Дочерние вершины этого узла, обозначенные на рисунке прямоугольниками, являются терминалами фрейма. Они отождествляются с набором вопросов типа "а что, если..." или в общем случае с набором любых других вопросов - тестом. Каждое из возможных заданий терминала, являющееся элементарным действием и обозначенное на рисунке кружком внутри прямоугольника-терминала, соответствует значению теста, т. е. совокупности ответов на набор вопросов. В простейшем случае имеется конечный перечень возможных ответов, который и определяет маркер терминала. Каждое из возможных заданий терминала рассматривается в свою очередь как фрейм следующего по рангу уровня (субфрейм) со своими терминалами, которые представляются на рисунке дочерними вершинами заданий - субфреймов и обозначаются прямоугольниками. Каждый из этих терминалов низшего ранга так же, как и терминалы фрейма верхнего ранга, отождествляется со своим тестом. Задания терминалов следующего по рангу уровня, также обозначенные кружками, являются более мелкими и сильнее конкретизированными элементарными действиями. Каждое из них соответствует значению теста. Если и эти задания еще не являются достаточно конкретными, то фрейм может иметь еще один более низкий по иерархии уровень.

Нетрудно убедиться, что описанная структура сводится к обычному графу И/ИЛИ, если отождествить терминалы фреймов и субфреймов, обозначенные на рис.8.1 прямоугольниками, с вершинами типа И, задания этих терминалов - с вершинами ИЛИ, а каждой дуге, идущей от вершины И к ИЛИ, поставить в соответствие значение теста. На рис. 8.2 дано представление ранее представленного фрейма (рис. 8.1) в виде графа И/ИЛИ. Заметим, что при такой формализации дочерними вершинами для верхнего узла будут вершины типа И.

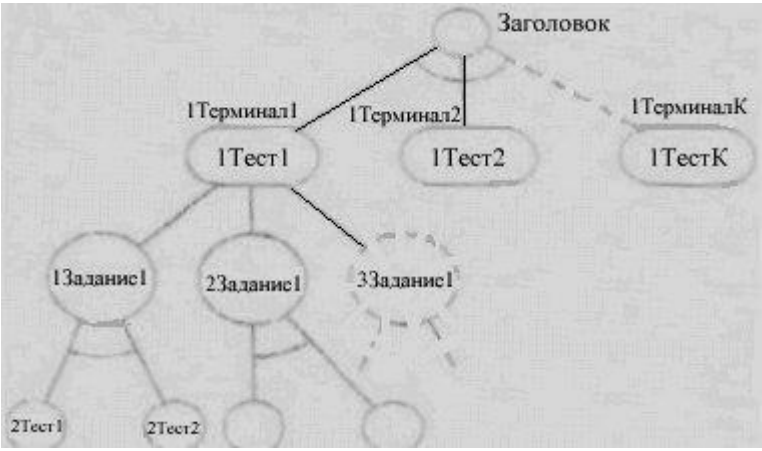


Рис. 8.2

Отметим, что терминалы фрейма и субфреймов в предлагаемой формализации можно в большинстве случаев отождествлять не только с тестом, но и с действием, которое является обобщением всех действий, определяемых возможными заданиями терминале. При таком представлении все вершины И/ИЛИ графа, формализующего фрейм, соответствуют действиям, причем тем более конкретизированным, чем ниже по иерархии вершина. Часто полезным оказывается использование обобщения графа И/ИЛИ, описывающего фрейм-сценарий, которое получается за счет представления вершинами графа не действий той или иной степени общности, а так называемых "схем действий". Эти схемы определяют лишь общую структуру действия, т.е. в них могут быть не определены конкретно или действующие лица, или средства действия, или место действия и т. п., или, наконец, и то и другое вместе. Например, возможна следующая схема действия: *X прибыть в V,* где X может быть любым конкретным

лицом, а V - любым конкретным местом. Чем ниже по иерархии вершина, тем более конкретно определяет она действие.

Важно подчеркнуть, что, если база знаний о мире образована в ЭВМ совокупностью таким образом формализованных фреймов-сценариев, то очень вероятна возможность многократного вхождения одних и тех же схем действий в разные фреймы-сценарии. Эта вероятность тем больше, чем большее количество фреймов включает база знаний. Для использования этого обстоятельства с целью экономии памяти полезно связать каждую схему действия с каждым своим вхождением во фреймы, образующие базу знаний, посредством "множества вхождений", которое представляет указания на все те места в базе знаний, где есть ссылки на данную схему. Кроме того, каждая схема действия связана с объемлющей ее схемой.

8.1. Примеры формализованного представления фреймов-сценариев

Приведенный выше фрейм-сценарий ресторана легко можно изобразить в виде такой графовой структуры И/ИЛИ (рис. 8.3). Номера сцен и действий сценария присвоены вершинам графа, представляющим соответствующие действия. Заголовок фрейма-сценария соответствует вершине графа. Каждая из четырех сцен соответствует **вершине И, т.е. терминалу фрейма**.

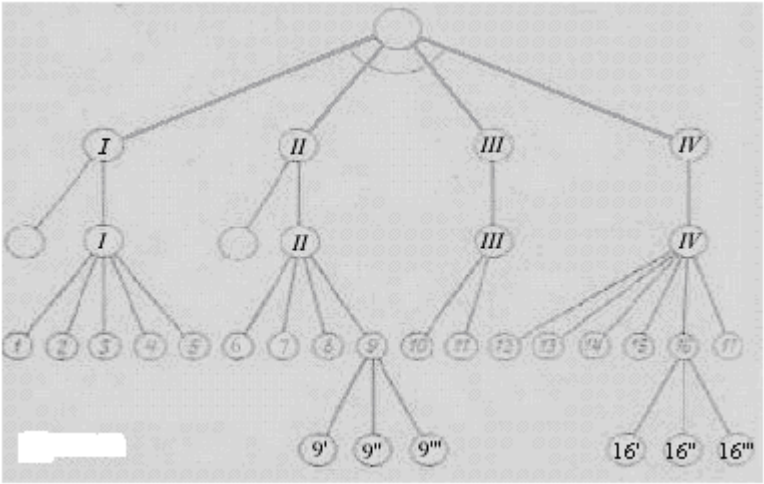


Рис. 8.3

Две первые *И* вершины, соответствующие входу в ресторан и заказу обеда, имеют по две вершины ИЛИ, остальные две - по одной. Первые вершины ИЛИ первых двух вершин И соответствуют действию, совершаемому при утвердительном ответе на вопросы "а что если посетитель уже в ресторане", "а что если заказ сделан другом". Очевидно, в этом случае действия не нужны и вершины пусты. Все остальные ИЛИ вершины так же, как и их материнские вершины, соответствуют входу, заказу, еде, уходу. Каждая из этих вершин имеет вершины *И,* соответствующие действиям, помеченным в сценарии арабскими цифрами. Наконец, вершины И *9* и *16* имеют по три дочерних вершины ИЛИ, соответствующие вариантам заказа и уплаты денег.

Другой иллюстрацией представления фрейма-сценария в виде графа И/ИЛИ является фрейм-сценарий (рис. 8.4).

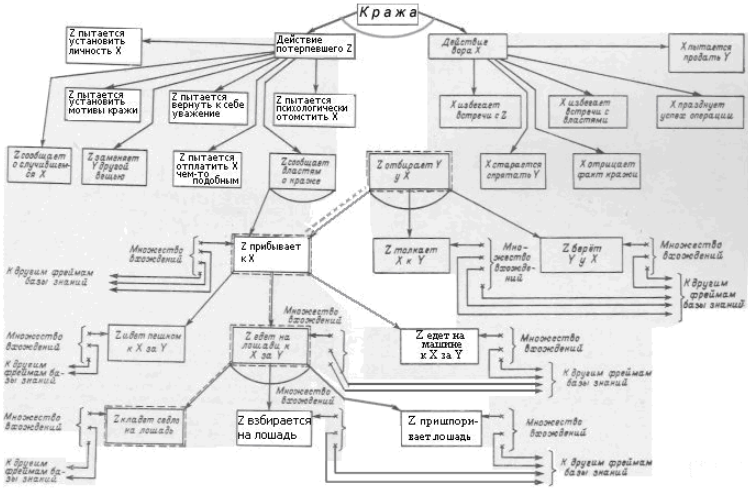


Рис. 8.4

Вершина этого графа соответствует названию сценария "кража". Она имеет две дочерние вершины II, одна из которых представляет собой схему действия вора X, вторая - потерпевшего 2 после кражи.

Дочерние вершины ИЛИ представляют собой более конкретные схемы действий потерпевшего. В данном случае имеет место шесть вариантов действия вора и девять вариантов действий потерпевшего. Всего имеется пятнадцать вершин ИЛИ. Выбор каждой из них зависит от значения теста, который представляется списком вопросов. Функция выбора осуществляется с помощью так называемой **тернарной сети переходов** (рис.8.5). Каждый узел этой сети представляет собой вопрос, входящий в тест, а каждая из трех дуг, исходящих из узла, соответствует трем возможным ответам на вопросы: "да", "нет", "не известно".

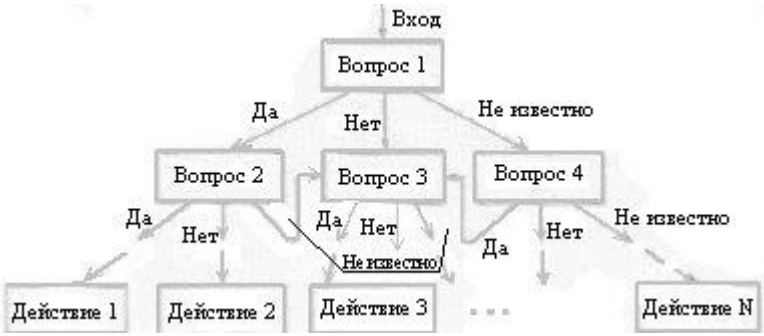


Рис. 8.5

В зависимости, от ответа на вопрос выбирается тот или иной узел сети, определяющий очередной вопрос теста. В результате имеет место продвижение по сети по тому или иному пути в зависимости от характера ответов. Оно заканчивается попаданием в один из заключительных узлов, каждый кз которых соответствует рекомендуемому действию. Дочерние вершины И еще более конкретизируют действия, представляемые вершинами ИЛИ. На рис.8.4 раскрыты лишь две из вершин ИЛИ. Первая соответствует схеме действий: "потерпевший Z отбирает вещь Y у вора X". Эта схема предполагает три обязательных последовательно совершаемых действия, представляемых изображенными на рис.8.4 тремя вершинами типа И. Каждая из этих вершин имеет дочерние вершины типа ИЛИ. На рис.8.4 представлено несколько вершин типа ИЛИ, относящихся к первой из вершин типа И, представляющих собой уточненные варианты схемы действия "потерпевший Z прибывает к X". Уточнение происходит за счет конкретизации средства действия, в данном случае - перемещения (лошадь, машина, пешком). Далее на рис.8.4 раскрыта одна из вершин ИЛИ, соответствующая передвижению на лошади. Вторая вершина ИЛИ соответствует схеме действия "потерпевший Z сообщает властям о краже". Эта схема имеет две обязательные схемы действия, первая из которых совпадает со схемой действия, представляемой дочерней вершиной вышерассмотрен-ной вершины ИЛИ. Однако если в первой схеме под X понимается вор, то во второй - власти.

8.2. Механизмы "приспособления" фрейма к реальной ситуации

Рассмотрим теперь возможные механизмы выбора из памяти фрейма и приспособления его к реальной ситуации. Как отмечено у М. Минского, именно этот процесс лежит в основе понимания человеком реальной ситуации. В случае же машинной базы знаний этот процесс открывает доступ к знаниям, материализированным в памяти ЭВМ в виде **совокупности систем фреймов**. Этот механизм приводится в действие двумя дополняющими друг друга потребностями. Первая - состоит в необходимости нахождения заданий терминалам фрейма, удовлетворяющим маркерам этих терминалов. Вторая - обусловлена требованием, чтобы рассматриваемый фрейм удовлетворял маркерам терминала более общего фрейма, объемлющего первый. Иными словами, **каждый фрейм считается приспособленным к ситуации, если он включен в более крупный фрейм в качестве задания его терминалу и если его терминалы заполнены заданиями, удовлетворяющими маркерам.**

В процессе понимания рассуждения, рассказа, точно так же, как и восприятия образа, ключевые слова, идеи, рассуждения, элементы образа **вызывают из памяти различные совокупности фреймов, большинство терминалов которых еще не согласовано с реальностью, а заполнено заданиями отсутствия.** По мере поступления новой информации выясняется, что некоторые из первоначально выбранных из памяти субфреймов не согласуются с реальностью. Они заменяются другими более подходящими субфреймами, удовлетворяющими двум вышеприведенным условиям. В простейшем случае такая замена осуществляется путем так называемой **операции согласования**. Она имеет место, когда отсутствуют особые знания относительно того, как поступать при смене фрейма, кроме некоторой общей стратегии. Эта стратегия состоят в выборе после поступления очередной порции входной информации такого фрейма, для которого вся ранее поступившая информация, например текст в случае понимания языка, и порция новой удовлетворяют маркерам его терминалов.

Проиллюстрируем эту стратегию на простом примере. Пусть в базу знаний входят два фрейма *A* и *B,* представленные в виде графовой структуры И/ИЛИ (рис. 8.6 а). Первая порция входной информации соответствует заданию 1 (зачерненный круг). Оно удовлетворяет

одному из трех T_1, T_2, T_3 терминалов фрейма А-T_2. Это задание является субфреймом фрейма А; субфрейм вызывается из памяти со своими терминалами T'_1 и T'_2, заполненными заданиями отсутствия 1 и 2.

На рис.8.6 б структура активизированного субфрейма показана жирными линиями. Новая порция информации соответствует заданию 2. Она удовлетворяет одному из терминалов активизированного субфрейма и вытесняет задание отсутствия 2, ранее занимавшее этот терминал, так как оно не совпадает с воспринятым заданием 2. В результате происходит уточнение активизированного субфрейма, структура которого изображена на рисунке 8.6в. Следующая порция информации – задание 3, как видно из графовой структуры И/ИЛИ, не может быть заданием ни одного из терминалов T'_1, T'_2 субфрейма задания 1, а является заданием одного из терминалов T''_1, T''_2 субфрейма а (рис.8.6 г).

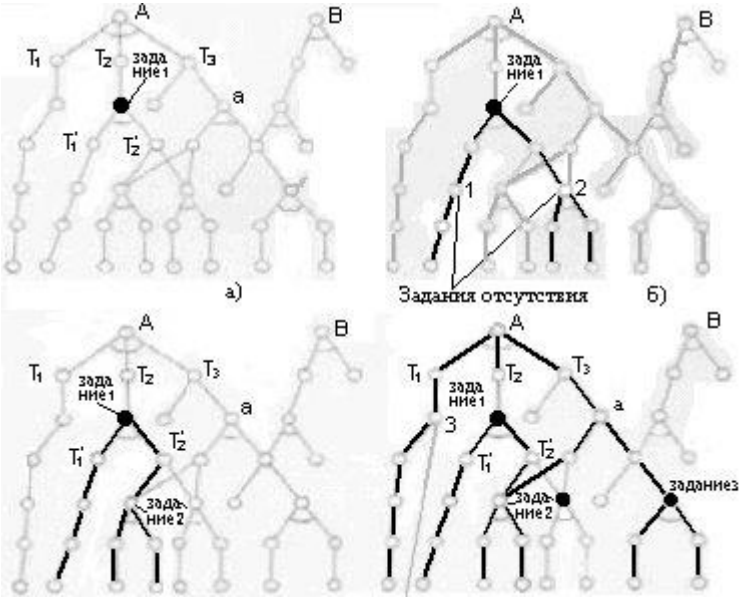


Рис. 8.6

В соответствии с используемой стратегией все три порции воспринятой информации должны удовлетворять терминалам одного

фрейма. Очевидно, этим фреймом может быть только фрейм А, так как субфрейм а является заданием одного из трёх его терминалов, задание же 1 является заданием другого терминала T_2 и в качестве субфрейма объемлет задание 2, так как последнее является заданием этого субфрейма. В результате фрейм А вызывается из памяти в виде структуры, изображенной на рисунке 8.6г жирными линиями. Два из трёх его терминалов заполнены субфреймами: задание 1 и а, третий терминал - заданием отсутствия 3.

Возможным содержательным примером описанной стратегии, является процесс интерпретации текста: "Пит украл скот Джейка. Джейк оседлал свою лошадь. Наутро скот был снова у Джейка". Предположим, что имеется специальная программа, которая осуществляет отбор предложений входного текста путем сопоставления их с субфреймами. Предположим, что в памяти ЭВМ содержатся знания в виде графовой структуры фреймов И/ИЛИ, один из которых описан выше и показан на рис.8.4, и существует программа выбора субфреймов, реализующая рассматриваемую стратегию. Она сравнивает каждое предложение входного текста с субфреймами графоподобной структуры знаний и отбирает те субфреймы, смысл которых соответствует анализируемым предложениям. Затем она выбирает фреймы, объемлющие все отобранные в результате анализа интерпретируемого текста предложения. Очевидно, эта программа отбора - один из вариантов рассматриваемой стратегии приспособления фрейма к реальности.

В данном примере первое предложение с помощью программы отбора активирует фрейм "кража". Далее, после ввода второго предложения программа активирует субфрейм, представляющий собой схему действия "Z кладет седло на лошадь", и отбирает субфреймы более высокого уровня, охватывающие активизированный субфрейм. В данном случае после анализа второго предложения образуются два набора вложенных друг в друга субфреймов, охватываемых субфреймом "действия потерпевшего". На рис.8.4 они помечены пунктиром. В результате на этом этапе возникают две возможные интерпретации текста "Джейк отобрал скот у Пита" и "Джейк сообщил властям о краже". Последняя фраза текста активизирует субфрейм "Z берет Y у X" и с помощью программы, отбора устанавливается ее принадлежность к субфрейму "Джек отбирает скот у Пита", что, очевидно, и является интерпретацией данного отрывка.

9. Анализ общей структуры интеллектуальной системы фреймов

9.1. Что такое интеллектуальная система фреймов?

Этот параграф следует начать с определения интеллектуальной системы фреймов (ИСФ). Сделать это нелегко. Тем не менее мы попытаемся дать определение системы ИСФ хотя бы для того, чтобы определить смысл, в котором мы будем пользоваться этим термином. Предварительно рассмотрим некоторые наиболее типичные подходы к определению ИСФ. Первое определение можно сформулировать в виде теста: если поведение ИСФ, отвечающей на вопросы, невозможно отличить от поведения человека, отвечающего на аналогичные вопросы, то она обладает интеллектом. Однако это определение дает лишь достаточные условия интеллектуальности ИСФ. В поисках необходимых условий мы можем прийти к определению типа «ИСФ разумна, если она решает задачи, требующие от людей интеллекта». Например, цель работ по ИИ состоит в создании ИСФ, выполняющих такие действия, для которых обычно требуется интеллект человека. Недостатком этого класса определений является их неоперациональность и излишняя антропоморфичность. Более операциональными представляются следующие определения .

Первое из них следующее: Существо *разумно,* если оно имеет адекватную модель окружающего мира (включающую интеллектуальный мир математики, понимание своих собственных целей и другие мыслительные процессы), если оно достаточно «умно», чтобы уметь дать ответ на широкий круг вопросов, используя для этого модель мира, и если оно может в случае надобности получать дополнительную информацию из внешнего мира и может выполнять во внешнем мире задачи, диктуемые его целями и не противоречащие его физическим возможностям.

Второе определяет ИСФ, используя следующий своеобразный вопросник:

1. Использует ли система модель мира?
2. Использует ли система модель мира для формирования планов действия, выполняемого в этом мире?
3. Включают ли планы направленный анализ альтернативных возможностей?

4. Может ли система переформулировать план, если его выполнение ведет к непредсказанным состояниям мира?
5. Может ли система использовать прошлый опыт для индуктивного расширения и корректировки модели мира?

Если ответ на все эти вопросы положителен, то система является ИСФ.

Мы хотим определить понятие ИСФ, используя приведенные выше два определения и основываясь на концепциях *цели* и *знания.* Под *целью* мы понимаем выраженные в некотором виде условия задач, возлагаемых на систему, в том числе задач, которые ставит система перед самой собой. Таким образом, система ведет себя *целенаправлено, если ищет план решения поставленной задачи (задач) и выполняет действия, адекватные этому плану.* Под *знанием* системы мы понимаем совокупность информации о внешнем мире, о мире абстрактных наук и о знаниях разумных существ, т. е. об их внутреннем мире, включая знание о внутреннем мире самой системы. Каждая из указанных компонент знания выражается в виде описаний объектов (истин), отношений между ними и законов, действующих в соответствующем мире.

Хотя приведенное деление знания является довольно относительным, оно может оказаться практически полезным при структурировании ИСФ. Так, мир философии мы считаем миром абстрактных наук, однако если ИСФ поручено создать новую философскую систему, то мир философии будет рассматриваться как внешний мир системы. Выделением абстрактного знания мы хотим подчеркнуть существование таких областей глобального знания, которыми система может пользоваться, но на которые она не воздействует в процессе решения задач.

Выделение знаний о внутреннем мире разумных существ может оказаться важным при работе в *динамическом мире,* т. е. в мире с несколькими независимыми источниками его изменения, описанном в явно зависимом от времени виде. Действительно, чтобы делать выводы о возможных и уже совершенных действиях другого разумного существа, нужно обладать моделью его модели внешнего мира, т. е. моделью его внутреннего мира.

Под ИСФ мы будем понимать *систему, обладающую способностью*

— *к накоплению и корректировке знания на основе активного восприятия информации о мире и обобщенного опыта,*

— *к целенаправленному поведению на основе накопленного знания.*

Главной темой *первого* этапа развития ИСФ, является разработка *теории эвристического поиска.* Общая постановка задачи эвристического поиска в неформальном виде выглядит следующим образом. Задана начальная ситуация и целевая ситуация. **Найти**

последовательность преобразований, приводящую из начальной ситуации в целевую. Например, в случае игры в шахматы начальной ситуации соответствует начальное расположение фигур на доске, целевой ситуации — множество всех положений, в которых одна из сторон выигрывает (или на доске стоит теоретическая ничья). Требуется найти последовательность ходов, преобразующих начальную ситуацию в целевую.

Задачи эвристического поиска подробно рассматривались в книге 1. Отметим, что исследовательским полем (пространством) для различных методов эвристического поиска являются всевозможные игры и головоломки. Некоторые из них могут считаться классическими (задачи о миссионерах и людоедах, об обезьяне и банане, «Ханойская башня», «крестики и нолики» и многие другие). Причину выбора такого рода задач прекрасно выразил Минский в предисловии к сборнику «Семантическая обработка информации» [Минский, 1968]: «Игры и математические задачи берутся не потому, что они просты и ясны, а потому, что они при минимальных начальных структурах дают нам наибольшую сложность, так что мы можем заняться некоторыми действительно трудными ситуациями». Другими словами, игры и головоломки позволили подвергнуть подробному анализу сложные *стратегии решения задач* при относительно простых и легко описываемых внешних мирах.

К концу 60-х годов был завершен ряд фундаментальных исследований в области эвристического поиска, использовавших теоретико-графические, формально-логические и некоторые другие представления. Появилась потребность испытать результаты этих исследований на более сложных и более близких к реальным мирах.

Однако попытки создания ИСФ, работающих в подобных мирах, натолкнулись на ряд серьезных трудностей, связанных в первую очередь с *моделированием внешнего мира.* Оказывается, что моделирование внешнего мира отнимает львиную долю труда, требуемого для создания всей ИСФ, и потому наиболее экономичным и эффективным хранилищем информации о реальном мире является сам реальный мир.

Так развитие ИИ потребовало постановки задач проектирования и реализации ИСФ, дающих возможность исследовать системы ИИ в реальном физическом мире.

Симбиоз ИИ и ИСФ оказывается весьма плодотворным. Проблема создания ИСФ выдвинула ряд новых проблем ИИ. В частности, задачи планирования ИСФ достигают той степени сложности, при которой многие из методов эвристического поиска, разработанных на первом этапе развития ИИ, оказались практически непригодными. Таким

образом, усложнение внешнего мира привело к необходимости дальнейшей разработки теории эвристического поиска. Задачи понимания естественного языка и обработки изображений, рассматривавшиеся исследователями и на первом этапе развития ИИ, получили более конкретные постановки в преломлении к ИСФ.

Таким образом, в качестве основных задач ИСФ выдвигаются:

1) *Проблема представления знаний о динамических открытых мирах.*
2) *Проблема построения сложных планов в этих мирах.*

В настоящей главе мы кратко опишем функциональную организацию алгоритмической системы фреймов (АСФ). На самом общем уровне АСФ состоит из трех алгоритмических систем — *системы восприятия, системы решения задач* и *эффекторной системы*. Однако такое представление является слишком поверхностным. С другой стороны, описание подробной блок-схемы АСФ заняло бы слишком много места и, не обладая достаточной наглядностью, не соответствовало бы главной цели этой главы — создать у читателя общее представление о том, как функционируют и как взаимодействуют между собой указанные выше алгоритмические системы фреймов. Поэтому наше описание будет носить промежуточный характер, с введением некоторых подробностей только там, где это касается ключевых моментов функционирования АСФ.

Это описание мы начинаем с системы решения задач.

9.2. Система решения задач

Система решения задач ИСФ состоит из трех главных частей: *подсистемы представления знаний, планирующей подсистемы* и *исполнительной подсистемы* (рис.9.1).

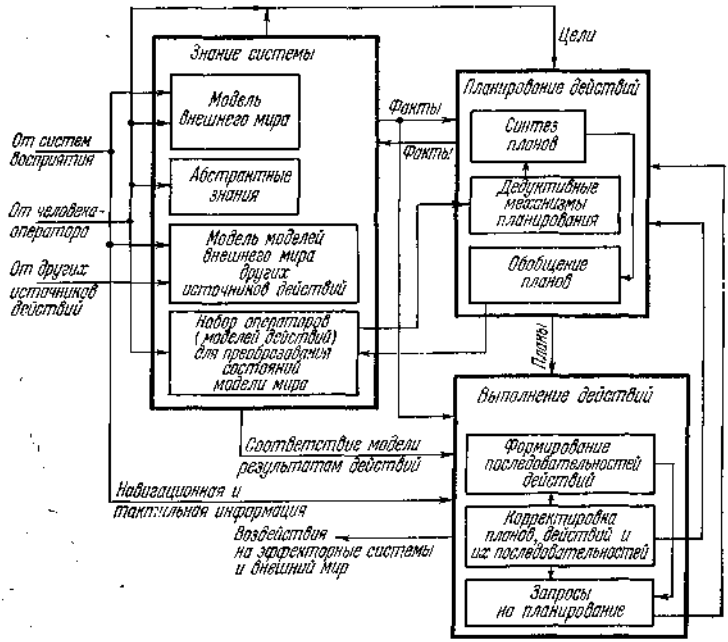


Рис. 9.1. Решающая система интеллектуальной системы фреймов.

Знание системы включает в себя *модель внешнего мира, абстрактные знания, знания о других источниках действий во внешнем мире и набор операторов,* преобразующих внешний мир. Причины выделения первых трех компонент знания мы упоминали ранее. Набор *операторов* представляет собой формальное описание *моделей действий,* которые способена совершать ИСФ, преобразуя с помощью этих действий одни состояния внешнего мира в другие и соответственно с помощью операторов одни состояния моделей внешнего мира в другие. Аналогичным образом могут быть определены операторы, преобразующие модели моделей внешнего мира других источников действия.

Информация о мире обычно хранится в виде совокупностей фактов (фреймов), связанных между собой теми или иными причинно-следственными связями. Каждый оператор обычно описывается условиями, определяющими возможность его применения, и результатами, ожидаемыми от его применения.

Следует выделить четыре важнейших аспекта, связанных со знанием ИСФ: *выражение знания,* его *накопление, корректировка* и *использование в процессе решения задач.*

Форма выражения знания называется *представлением.* Проблема представления знания, обладающего достаточной общностью и позволяющего эффективно решать задачи, является одной из центральных в ИСФ. Постановка этой проблемы и основные методы представления рассматриваются в книге 1.

Накопление знаний выступает в свою очередь в двух аспектах. С одной стороны, это накопление множества фактов (фреймов) и установление между ними требуемыз отношений. Оно осуществляется за счет восприятия информации из внешнего мира и от человека-оператора, а также в результате решения задач системой. В последнем случае **факты (фреймы) называются <u>выведенными</u>**. С другой стороны, ИСФ должна обладать возможностью накапливать методы решения задач. Накопление методов может происходить как в результате общения с человеком-оператором, так и в результате обобщения самим ИСФ опыта решения задач.

Корректировка знаний осуществляется как результат взаимодействия ИСФ с внешним миром, человеком-оператором. Наконец, использование знаний в процессе решения задач является ключевым моментом при построении формализмов, моделей и методов ИИ. Особенно следует отметить важность правильного использования смыслового содержания задач, или *семантического знания,* являющегося основой построения всех эффективных систем ИИ.

Общие модели и методы решения задач и особенности наложения на них семантических знаний рассматриваются в книге 1.

Планирующие подсистемы ИСФ являются его *основной решающей частью.* Синтез планов осуществляется на основе поставленных целей и накопленного знания, причем цели могут как задаваться человеком-оператором, так и вырабатываться самой ИСФ. При синтезе планов широко используются дедуктивные механизмы планирования, **<u>определяющие план</u> как последовательность применимых и пригодных операторов, выбираемых из накопленного набора операторов**. Поскольку в достаточно сложных задачах планирования на каждом шаге имеются альтернативные возможности выбора пригодных операторов среди множества применимых, чисто дедуктивные механизмы должны были бы испробовать все возможности. Использование *эвристик,* т. е. *наложение семантики на дедуктивные механизмы,* позволяет сделать выбор более направленным, а следовательно, процесс планирования — более эффективным.

Важнейшим моментом в планировании ИСФ является возможность *обобщения планов.* Эта проблема выступает в двух взаимосвязанных аспектах. С одной стороны, представляется целесообразным строить приблизительные планы с последующей их детализацией. Это повышает эффективность планирования, особенно в сложных мирах, как за счет лучшего «понимания» планирующей подсистемой общих закономерностей пространства поиска решений, так и за счет структурирования общего описания мира на более и менее существенные факты о мире. С другой стороны, для ИСФ важно обобщать уже построенные планы так, чтобы использовать их в дальнейшем как дополнительные укрупненные операторы, «обучаясь» решать все более и более сложные задачи. Накапливая таким образом модели обобщенных действий, ИСФ может эффективно использовать их и для построения приблизительных планов все увеличивающейся общности и сложности.

Другим важным фактором, определяющим эффективность планирования, является обеспечение гибкого *взаимодействия процессов планирования* и *выполнения действий,* соответствующих построенным планам. Именно такое взаимодействие определяет способность ИСФ успешно функционировать в реальном мире. Напрашивающимся решением проблемы взаимодействия указанных выше процессов является организация обратной связи, позволяющей корректировать модель мира и планы в соответствии с действительными изменениями, происходящими в мире.

Эти функции выполняются исполнительной подсистемой ИСФ. Как видно из рис. 9.1, эта подсистема формирует и выполняет последовательность действий, соответствующих операторам плана, получаемого от планирующей подсистемы, а также выдает команды о корректировке плана в случае несоответствия действительного состояния мира и его модели в ИСФ. Вопросы планирования и выполнения действий ИСФ рассматриваются в гл.10.

9.3. Система восприятия

Рассмотрим теперь особенности функционирования алгоритмических систем восприятия и воздействия на эффекторные системы (рис.9.2). Мы выделяем два основных источника информации ИСФ: внешний мир и человек-оператор.

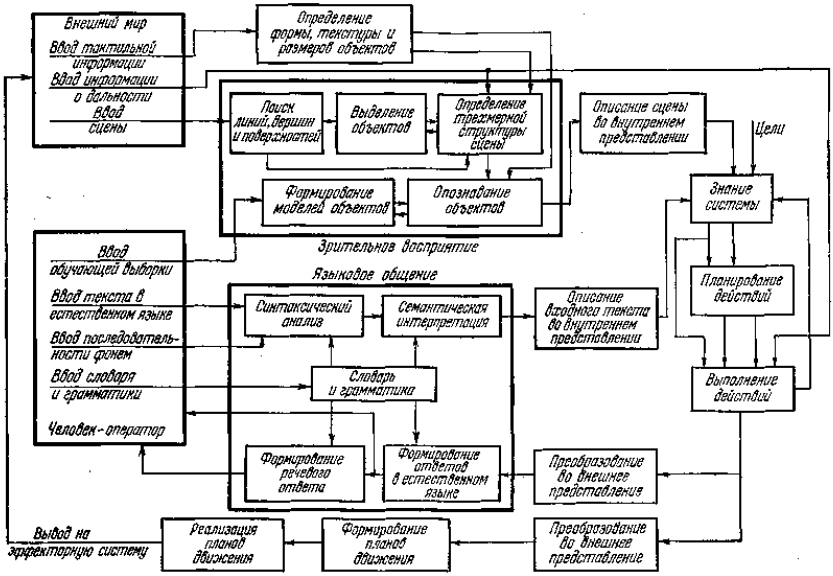


Рис. 9.2. Функциональная организация систем восприятия и воздействия на внешний мир.

С первым из них ИСФ общается с помощью систем датчиков (тактильных, телевизуальных, дальномерных) и систем ввода информации с этих датчиков. Эти системы на рис. 9.2 не показаны, что соответствует введенному нами понятию об ИСФ.

С человеком-оператором ИСФ общается с помощью стандартных терминалов диалоговых систем (телетайпов, дисплеев и т. д.) или с помощью систем понимания речи.

ИСФ осуществляет воздействие на человека-оператора с помощью терминалов диалоговых систем или систем речевого вывода, а на внешний мир — с помощью электромеханических систем Эти системы также не показаны на рис. 9.2.

Периферийные алгоритмические системы ИСФ представляют из себя сложные специализированные системы ИИ. Мы вкратце опишем некоторые из таких систем, с тем чтобы читатель смог составить представление об уровне их сложности.

На рис. 9.2 приведена одна из возможных схем функциональной организации зрительного восприятия ИСФ. Следует сразу заметить, что приведенная на рисунке последовательность этапов не является единственно возможной. Более того, она может определяться внешним миром системы восприятия, образуя сложные итеративные циклы. Во введенном в некоторой форме (например, в виде распределения точек

яркости) изображении производится поиск линий и вершин объектов (мы рассматриваем для простоты сцены с объектами, ограниченными плоскими гранями), на основании результатов которого может быть описана *трехмерная сцена.* Эти две задачи зрительного восприятия, вообще говоря, относятся к первичной обработке сцен, со свойственными ей специализированностью и сильной зависимостью от особенностей применяемой телевизуальной аппаратуры. Однако и в этих задачах уже можно наблюдать некоторые элементы ИИ. Дело в том, что описанная трехмерная сцена отнюдь не является единственно возможной. Появляющаяся в связи с этим недетерминистичность выбора среди множества альтернатив требует решения задач направленного поиска, эффективность которого в значительной степени зависит от накопленного в системе или введенного в нее семантического знания. В еще большей степени семантика определяет эффективность решения важнейшей задачи анализа сцены — *разбиения сцены на тела* (объекты, фреймы). Это касается такой проблемы, как идентификация границ, где семантика, выраженная в виде законов физически возможных конфигураций линий в области вершин, буквально в десятки тысяч раз снижает объем требуемого перебора; такой проблемы, как устранение теней, где оптические законы света и тени не только резко сокращают поиск, но и позволяют использо-. вать информацию о тенях как дополнительную для выделения объектов; такой проблемы, как выделение скелетов (фреймов) объектов, где неоднозначность, связанная с неполной видимостью объекта в сложной сцене, может быть эвристически устранена с помощью построения гипотез и сложных, основанных на семантике процедур проверки и доказательства (или опровержения) этих гипотез.

Очевидно, что система, способная решать упомянутые задачи, должна обладать моделью своего мира (сцены) в представлении, внешнем по отношению к системе решения задач ИСФ; она должна также обладать универсальными механизмами вывода (в частности, дедуктивными), управляемыми семантическим знанием. Другими словами, эта система должна обладать всеми существенными признаками ИИ.

Характерной для ИСФ является и задача *идентификации (опознавания) объектов и формирования понятий высшего уровня,* которые становятся исходной информацией для описания сцены во внутреннем представлении и накоплении знания в модели внешнего мира решающей системы ИСФ. Ключевым фактором на пути к успешной идентификации объектов является семантически ориентированное устранение неоднозначности выбора, характерное для ИСФ.

Весьма важным для решения задач разбиения сцены и идентификации объектов является взаимодействие систем восприятия различного вида,

в частности, *бинокулярное зрение* и *взаимодействие тактильной и телевизуальной систем.* При этом вспомогательные виды восприятия обеспечивают важный семантический материал для окончательной проверки выдвинутых гипотез.

Некоторые вопросы построения математических моделей и методов решения задач зрительного восприятия рассматриваются в гл. 12.

Рассмотрим теперь функциональную структуру подсистемы языкового общения ИСФ с человеком-оператором. Отметим прежде всего, что функциональная структура подсистемы, приведенная на рис.9.2, совпадает с классической структурой *«вопросно-ответных систем»* (ВОС), за исключением того, что этап дедуктивного вывода замкнут через решающую систему ИСФ. Термин «вопросно-ответная система» обычно употребляется в двух смыслах: в широком смысле термины «вопросно-ответная система» и «система ИИ» суть синонимы; в узком смысле ВОС — это информационно-поисковая система, обладающая способностью *воспринимать смысл текста* в естественном языке, ограниченном в той или иной степени, и способностью давать ответы, не только непосредственно извлекаемые из знания системы, но и *ответы, выводимые из знания* системы и входного текста. Следует понимать, что указанные обобщения определяют качественно новый уровень ВОС по сравнению с информационно-поисковыми системами относительно класса решаемых задач (и соответственно относительно их сложности). Здесь и далее в гл. 11, посвященной изложению методов языкового общения ИСФ, мы употребляем понятие ВОС в узком смысле слова.

Основными этапами обработки текста в естественном языке до его преобразования во внутреннее представление решающей системы ИСФ и дедуктивного вывода ответа являются *синтаксический анализ* и *семантическая интерпретация.*

Синтаксический анализ текста сводится к определению структуры входящих в текст предложений в соответствии с грамматикой языка и к идентификации слов текста и соответствии со словарем.

Семантическая интерпретация сводится к пониманию смысла текста, т. е. к нахождению семантических отношений между словами текста в терминах знания системы.

Согласно существующим представлениям для понимания текста необходимо *итеративное взаимодействие* этапов синтаксического анализа и семантической интерпретации, так что изображенная на рис. 9.2 последовательность этих этапов носит до некоторой степени условный характер.

Глобальная задача ВОС заключается в том, чтобы на основе накопленного знания и входного текста (который идентифицируется с

целью) однозначно выразить последний в терминах внутреннего представления, а затем преобразовать, если это необходимо, внутреннее представление в ответ, выраженный в естественном языке. Таким образом, и здесь мы видим, что подсистема языкового общения ИСФ удовлетворяет нашему определению системы ИСФ.

В рамках наших общих рассуждений мы, как указывалось выше, замкнули подсистему языкового общения через решающую систему ИСФ. Это, однако, не означает, что подсистема не может обладать СБОИМ локальным знанием и локальными дедуктивными механизмами. Кроме того, мы показываем на рис. 9.2 этап формирования ответа в естественном языке, связанным с решающей системой через условную подсистему выполнения планов. В случае ВОС эта подсистема представляет собой высший уровень диалоговой системы общения человека с ИСФ, через который осуществляется формирование сложных ответов (например, неполных или условных) и корректировка процесса вывода в соответствии с указаниями человека.

9.4. Эффекторная система

Кажущаяся простота функциональной структуры подсистемы воздействия на внешний мир не должна вводить читателя в заблуждение: вся сложность этой подсистемы заключена в этапах формирования и реализации планов управления ИСФ. Рис. 9.3 показывает, что более подробная функциональная структура рассматриваемой подсистемы почти аналогична структуре решающей системы ИСФ (с точностью до уровня планирования и выполнения).

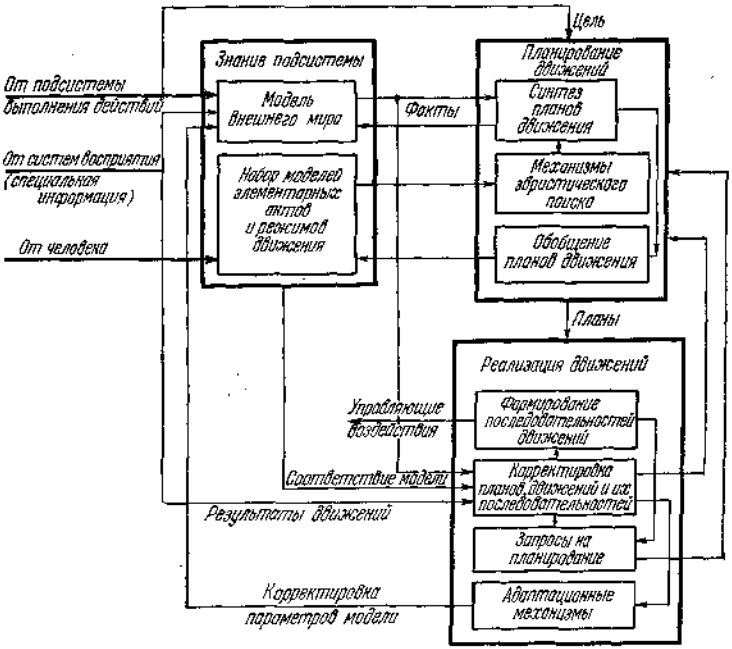


Рис. 9.3. Система воздействия на внешний мир.

На этом рисунке под внешним миром подразумеваются факты об окружающей подсистему среде, т. е. *механические и геометрические свойства объектов,* с которыми работает ИСФ, *свойства управляющих органов* и их текущие положения, а также основные *параметры управляющих органов.* Соответственно модель отражает текущее знание подсистемы о ее внешнем мире. Это знание может корректироваться человеком, знанием решающей системы («центральное» знание) или самой подсистемой в процессе ее функционирования.

Набор *операторов управления* представляет собой описание элементарных операций управления и их характерных режимов. Он может задаваться человеком и корректироваться как человеком, так и подсистемой планирования управления в результате обобщения планов точно так же, как и в случае решающей системы.

Подсистема планирования управления получает в качестве цели планируемое решающей системой ИСФ действие и синтезирует план выполнения этого действия в виде последовательностей операторов управления. Поскольку эти последовательности могут быть составлены

отнюдь не единственным способом, синтез управляется механизмами эвристического поиска, обеспечивающими направленный перебор альтернатив и, возможно, оптимизирущий план управления по некоторому критерию, например, минимуму потребления энергии, максимуму скорости отработки или какой-либо функции этих параметров.

Подсистема реализации управления формирует последовательность управляющих воздействий на соответствующие органы и следит за соответствием запланированных управлений действительным результатам. В случае несоответствия подсистема либо корректирует требуемые управления или их последовательности, либо выдает запрос на повторное планирование части плана или плана в целом.

Наконец, *адаптационные механизмы* следят за качеством отработки управляющих воздействий и в случае обусловленных нарушений корректируют те или иные параметры модели внешнего мира.

Приведенное описание показывает, что и подсистема воздействия на внешний мир обладает всеми признаками ИСФ.

Конечно, такая функциональная организация должна реализовываться, начиная с некоторого уровня сложности задач, возложенных на ИСФ.

Таким образом, мы показали, что построение периферийных подсистем ИСФ требует использования в них систем ИСФ, мало отличающихся от решающей системы ИСФ. При этом не утверждается, должна ли обладать каждая периферийная подсистема своими системами ИСФ или работать, например, с центральным дедуктивным механизмом в режиме разделения этого ресурса между всеми подсистемами. Однако так или иначе любая из алгоритмических подсистем ИСФ должна использовать те или иные общие формализмы, модели и методы ИСФ.

В гл. 11 и 12 рассматриваются приложения общих и некоторых специализированных моделей и методов к построению подсистем языкового общения и зрительного восприятия соответственно.

10. Планирование и выполнение действий в интеллектуалных системах фреймов

10.1. Анализ систем решения задач

10.1.1. Среды функционирования ИСФ и планы

Рассмотренные в предыдущих разделах методы представления и решения в ИСФ создают необходимую концептуальную и алгоритмическую основу для перехода к анализу систем решения задач

в ИСФ. Мы предполагаем, что ИСФ существует и работает в среде обеспечивающей функционирование ИСФ, которая обновляется как в результате действий самой ИСФ, так и в результате некоторых независимых от ИСФ действий в этой среде.

ИСФ обладает способностью воспринимать информацию о среде функционирования, рещать задачи и выполнять заданные действия (предписания, директивы) в этой среде. С алгоритмической точки зрения это означает, что ИСФ обладает описанием среды функционирования в виде ее (среды) модели, списком моделей элементарных действий (директив), или операторов, в общем случае изменяемых разработчиком ИСФ, либо автоматически в результате обобщения опыта функционирования ИСФ, а также общим механизмом создания целенаправленных стратегий в виде последовательности элементарных действий (директив) или операторов и воплощения стратегий в рациональные программы действий ИСФ в окружающей среде.

Будем различать два больших класса окружающих сред — статические и динамические среды.

Статической средой мы назовем среду, в которой источником активных действий является ИСФ или сеть ИСФ, управляемая централизованно, а факты о среде, представляемые в ИСФ не зависят в явной форме от времени. Соответственно мы можем указать следующие факторы, определяющие *динамичность среды:*

а) наличие в среде независимых от ИСФ источников действий, изменяющих среду; к числу таких источников могут относиться другие ИСФ или «слепые силы природы»;

б) наличие в среде фактов и определение в ней действий, явно зависящих от времени (например, факт OPEN (D1, 17, 19), т. е. «дверь D1 открыта с 17 до 19 часов», или оператор TURN (R1, 20), т. е. «вернуться в комнату R1 к 20 часам».

Следствием статичности среды является возможность упорядочения множества фактов, описывающих среду, по степени их неизменности в процессе решения задачи. Такое упорядочение является весьма относительным и может меняться от задачи к задаче. Так, если рабочий перемещает ящики, то факты об их цвете являются более неизменными, чем об их положении. Если же рабочий красит ящики, то ситуация оказывается противоположной. Однако, по-видимому, положение ящика является всегда более неизменным фактом, чем положение рабочего.

Возможность ранжирования фактов по степени их неизменности является для нас важной (см. п. 10.4.1). Отметим, чго в сложных динамических средах можно и не найти такое ранжирование.

Целенаправленная стратегия поиска решения задачи называется *планом.*

Определим *пространство моделей M* как множество всех возможных состояний моделей среды, описываемых ИСФ.

Моделью элементарного действия *оператором,* называется некоторое отношение F_j, определенное в пространстве моделей, т. е. $F_j{:}M{\to}2^M$. Множество всех операторов обозначим через F. В общем случае отношение, определяющее оператор, является частичным отношением, определенным на собственном подмножестве M_k множества M.

Теперь мы можем дать формальное определение плана

Планом называется помеченный направленный граф, удовлетворяющий следующим условиям:

I) каждая дуга графа помечена оператором $F_j \in F$, в общем случае оператором-схемой, т. е. семейством операторов, определяемым некоторым параметром;

2) с каждой вершиной графа n_k связана некоторая формула R_k, определяющая в свою очередь подмножество $M_k \subseteq M$,

M — множество моделей;

3) из одной вершины исходят одинаково помеченные дуги;

4) множество M_k, соответствующее n_k, содержится в области определения оператора F_j, помечающего дуги, исходящие из этой вершины.

Это определение является важным по нескольким причинам. Во-первых, оно определяет *обобщенный план,* т. е. множество планов, которое можно получить из исходного конкретной подстановкой значений параметров. Во-вторых, множество дуг, исходящих из одной вершины графа, определяет множество *возможных результатов применения* оператора, т. е. для каждого $m_i \in M_k$ $F_j(m_i)=\{m_1, m_2, \ldots, m_n\}$. Наконец, в-третьих, из определения вытекает применимость оператора F_j к любому состоянию модели $m_i \in M_k$.

План называется *простым,* если для всех вершин n_k оператор является однозначной функцией модели, т. е. $F_j(m_i)=m$, $m_i \in M_k$, $m \in M$ (из каждой вершины исходит ровно одна дуга).

План называется *сложным,* если каждому результату применения (віполнения) оператора m_p, p=1, 2, ..., n, приписывается некоторая оценка C_p правдоподобия его появления или оценка его полезности с точки зрения достижения цели (т. е. оценки приписываются дугам, исходящим из вершины n_k). В случае, если альтернативным результатам применения оператора не приписаны оценки (или, что то же самое, приписаны одинаковые оценки), *план* называется *составным.* План не обязательно может приводить к достижению поставленной цели. Мы вводим в связи с этим определение *полного* и

неполного плана. План P называется *полным планом из начальной модели* m_0 *к целевой* m, если существует $P' \subset P$ такой, что 1) P' есть план; 2) все вершины P' достижимы из такой вершины n_r, что $m_k \in M_r$; 3) существует по меньшей мере одна такая вершина $n_t \in P'$, что R_t влечет за собой m.

План, не удовлетворяющий хотя бы одному из этих условий, называется *неполным.*

Алгоритмическая система ИСФ, осуществляющая построение планов решения задачи, называется *планирующей системой* (ПС).

10.1.2. Планы и действия

Проблема планирования для ИСФ занимает важное место в проблемах решения задач. Поскольку ИСФ функционирует в реальной среде, т. е. среде, неполностью соответствующей той модели среды, которая заложена в ИСФ, взаимосвязь проблемы планирования и выполнения действий является принципиально важным фактором, определяющим возможность выполнения ИСФ поставленных перед ней задач.

В отличие от оператора, *элементарное действие* ИСФ в среде функционирования есть отношение, определенное на декартовом произведении пространств среды W и моделей M и отображающее его в самого себя,

$$Q_j : W \times M \rightarrow W \times M. \qquad (10.1)$$

Таким образом, мы можем выделить в действии две компоненты: компоненту нового состояния среды

$$w = Q_{j,w}(w_i, m_i) \qquad (10.2)$$

и компоненту нового состояния модели этой среды в ИСФ

$$m = Q_{j,w}(w_i, m_i). \qquad (10.3)$$

Идесь Q_j представляет из себя *действие-схему,* т. е. множество действий, определяемое некоторым параметром.

В силу неточности и неполноты моделей для достаточно сложных сред, а также погрешностей восприятия окружающей среды ИСФ и ее подсистем мы не можем ожидать функционального соответствия между средой и ее моделью в ИСФ, т. е. отношение моделирования $R: W \rightarrow M$ не является функцией n в строгом смысле слова, что не позволяет в общем случае надеяться на полную формализацию процесса построения адекватных моделей среды функционирования ИСФ.

При создании формальной теории планирования предписаний действий в ИСФ естественным решением является организация

обратной связи, позволяющей корректировать модель среды в соответствии с действительными изменениями, происходящими в ней.

Выполнение планов и корректировка модели среды в соответствии с действительными результатами их выполнения осуществляются специальной алгоритмической системой ИСФ, называемой *исполняющей системой* (ИС). Таким образом, ИС выполняет план, обращаясь к определенным действиям, которые, как предполагается, соответствуют операторам в плане, получает планы (в том числе неполные), обращаясь к ПС, и выдает команды ПС о корректировке плана в случае выявления несоответствий действительного состояния среды и ее модели в ИСФ. В каждом состоянии ИС должна осуществить выбор между планированием и действием.

Определения полных и неполных, простых, составных и сложных планов создают основу для классификации ИС. Эта классификация приведена в таблице 10.1.

Таблица 10.1

		Наличие только полных планов	Наличие полных или неполных планов
Отсутствие обратной связи (только простые планы)		A	B
Наличие обратной связи	Простые планы	C	F
	Составные планы	D	G
	Сложные планы	E	H

Примечание. Буквы идентифицируют класс ИС для таблицы 10.2 и дальнейшего изложения.

Здесь под обратной связью понимается проверка модели с помощью ИС после каждого шага выполнения плана. В таблице 10.2 сведены основные функциональные особенности ИС различного класса.

Таблица 10.2

Класс ИС	Характер взаимодействия ПС и ИС	Действия ПС при несоответствии мира и текущей модели	Характер работы ИС
A	Полное соответствие последовательности действий плану (даже если он неудачен)	Несоответствие не учитывается, так что управление к ПС в процессе выполнения плана не передается	Простое прохождение списка действий
B	Полное соответствие на уровне неполного плана	При несоответствии неполного плана результатам его выполнения возможен пересмотр плана при хранении дерева планирования с оценками полезности	Простое прохождение списка действий, соответствующего неполному плану
C	Полное соответствие при успешном выполнении. Обмен информацией для проверки соответствия модели миру после каждого шага выполнения	Полное перепланирование, однако при хранении дерева планирования возможно использование промежуточных результатов	Выполнение действия — проверка соответствия — сигнал о выполнении (соответствие) или о перепланировании (несоответствие)
D E	То же, что и C, но производится проверка и модели, и плана	Поиск дочерней вершины, соответствующей новому состоянию модели. Если такой вершины нет, аналогично C	Выполнение действия — проверка соответствия — поиск возможных продолжений, при несоответствии — перепланирование в случае неудачи
F G H	То же, что в C; То же, что в D и E — На уровне неполного плана. Выбор планирования или выполнения в соответствии с оценками	То же, что в C, D и E соответственно для неполного плана. Перепланирование по сигналу ИС	Решение задачи выбора планирования или выполнения на каждом шаге

Наиболее интересным является выбор направления деятельности ИС в случае работы с неполными планами. На каждом этапе решения задачи ИС стоит перед дилеммой: выполнять действия в соответствии с намеченным неполным планом или продолжать построение неполного плана. Каждая из альтернатив может привести к нежелательным результатам. Действительно, при выполнении неполного плана можно не достигнуть цели, в то время как дальнейшее планирование могло бы это показать. С другой стороны, продолжение планирования может оказаться нецелесообразным, если выполнение уже построенного плана могло бы показать его бесперспективность. Таким образом, планирование и выполнение для ИС класса F, G и H являются конкурирующими действиями. С каждым из этих действий могут быть связаны некоторые оценки полезности в виде, например, функции стоимости уже построенного плана и оценки стоимости остатка плана, приводящего к цели. Другим возможным подходом являлась бы формулировка задачи планирования как игры с природой. Во всяком случае, именно здесь открываются возможности использования изложенных ранее алгоритмов. В свою очередь ИС по результатам своих действий и оценивая текущее состояние планирования, могла бы определить в каждом конкретном случае, планировать или выполнять.

Постановка задачи о совместной оптимизации процессов планирования и выполнения связана, по-видимому, с некоторым обобщением процессов в единый процесс, аналогичный процессу решения задачи. Два таких возможных обобщения мы рассмотрим в п. 10.5.2.

В заключение этого параграфа мы отметим еще одну задачу, встающую в связи с проблемой планирования и выполнения действий,— задачу обобщения моделей и уже найденных планов и использования их как в последующих процессах построения планов, так и при выполнении действий. Один шаг на пути к такому обобщению уже сделан введением операторов-схем. Дальнейшее развитие идей обобщения — это постепенное наращивание множества операторов путем их укрупнения, а также соответствующее обобщение условий их применения, с тем чтобы создать иерархию планов от укрупненных к более детализированным.

Заучивание и обобщение получаемых в процессе решения задач успешных частичных решений является предметом исследований в области теории истин и информации. Мы рассмотрим идеи решения задач в преломлении к проблемам планирования и выполнения действий в ИСФ.

10.2. Планирующая система «Решатель задач»

Рассмотрим вопросы, связанные с планированием и выполнением действий в ИСФ, кратким описанием планирующей системы «Решатель задач» (ПС РЗ) по следующим причинам:

1) система представляет весьма широкий класс универсальных решателей задач для ИСФ, работающих в достаточно сложной среде;

2) ПС РЗ является системой, которая работает в реальной среде, взаимодействуя с ИС «Исполнитель планов» (ИП);

3) ПС РЗ является иллюстрацией проблем планирования и выполнения действий и возможных путей их решения;

4) формализм ПС РЗ отражает все преимущества и недостатки декларативных предписаний, создавая в то же время приемлемую основу для ряда обобщений.

Тривиальная (примитивная) ИСФ ПС РЗ функцинрует в среде, состоящей из комнат с дверьми и предметами (ящики, призмы), и способна в автоматическом режиме осуществлять с этими предметами относительно простыедействия.

Среда ИСФ представляется в ПС в виде модели, состоящей из набора правильно построенных формул (ппф) в исчислении предикатов первого порядка, описывающих состояние среды в данный момент.

Действия в ИСФ моделируются множеством операторов, определяемых *наименованием, списком параметров,* а также *условиями применимости (предусловиями)* и *результатами действия* в виде *схем ппф,* т. е. ппф, зависящих от параметров.

Результаты действия оператора описываются *списком вычеркивания* тех схем ппф, которые перестают быть истинными после применения оператора, и *списком добавлений* схем ппф, которые становятся истинными после применения оператора.

Важно различать параметры, входящие в схему ппф, и обычные связанные переменные, т. е. переменные под знаком кванторов общности или существования. В связи с этим системы доказательства предписаний, описанные в книге 1, требуют некоторой модификации.

Пусть имеется некоторая целевая схема ппф $G(p)$, p — множество параметров схемы, которую нужно доказать на множестве M дизъюнктов, т. е. найти опровержение множества дизъюнктов $M \bigcup \{\sim G(p)\}$. Для вычисления частного случая p' множества p, при котором множество $M \bigcup \{\sim G(p)\}$ невыполнимо, можно использовать стандартный алгоритм унификации (см. книгу 1). С помощью этого алгоритма можно найти наиболее общие частные случаи параметров, при которых обеспечивается унификация. Однако необходимо

определить, какие подстановки допустимы в случае параметров. Определим следующие типы термов, которые могут быть подставлены вместо переменной: переменные, константы, параметры и функциональные термы, не содержащие переменных. Вместо параметра могут быть подставлены следующие типы термов: константы, параметры и функциональные термы, не содержащие функций Сколема, переменных или параметров.

Поскольку один и тот же параметр может иметь несколько вхождений в множестве дизъюнктов, он должен замещаться при резолюции (директиве) термом во всех дизъюнктах, являющихся производными от резольвенты.

В качестве примера приведем оператор (фрейм) «переместить объект *k* из места *m* в место *n*». Наименование:

PUSHTO *(k, m, n) (k, m, n* — параметры).

Предусловия:

At (Рабочий, *m)* $\wedge$ At *(k, m)* (и рабочий, и объект *k* должны быть в месте *m).*

Список вычеркиваний:

At (Рабочий, *m*);

At *(k, m)* (Рабочий и объект *k* больше не находятся в месте *m).*

Списокдобавлений: At (Рабочий, *n);*

At(*k, n)* (Рабочий и объект *k* находятся в месте *n).*

Конечная модель или цель также описывается в виде ппф.

Трудности использования формальных методов доказательства предписаний в качестве стратегий поиска плана, главным образом связанные с проблемой границ, оказались непреодолимыми. В связи с этим в ПС процесс поиска плана полностью отделен от метода доказательства предписаний. Последний используется только внутри модели среды для ответа на вопросы, связанные с анализом применимости операторов и проверкой выполнимости условий достижения цели.

Для поиска решения в пространстве моделей ПС РЗ использует описанный ранее механизм редукции GPS (см. книгу 1). Преимущество такого комбинированного подхода заключается в возможности рассмотрения сложных моделей, используя описательную мощь исчисления предикатов первого порядка, и использования эффективных эвристик разбиения цели на подцели, свойственных механизму редукции GPS.

В процессе поиска механизм редукции порождает иерархию цели, подцелей и моделей, которую можно представить в виде дерева поиска. Каждая вершина дерева имеет вид (модель, (список целей)) и

соответствует задаче достижения по порядку подцелей из списка целей указанной модели среды. Схема работы ПС представлена на рис. 10.1.

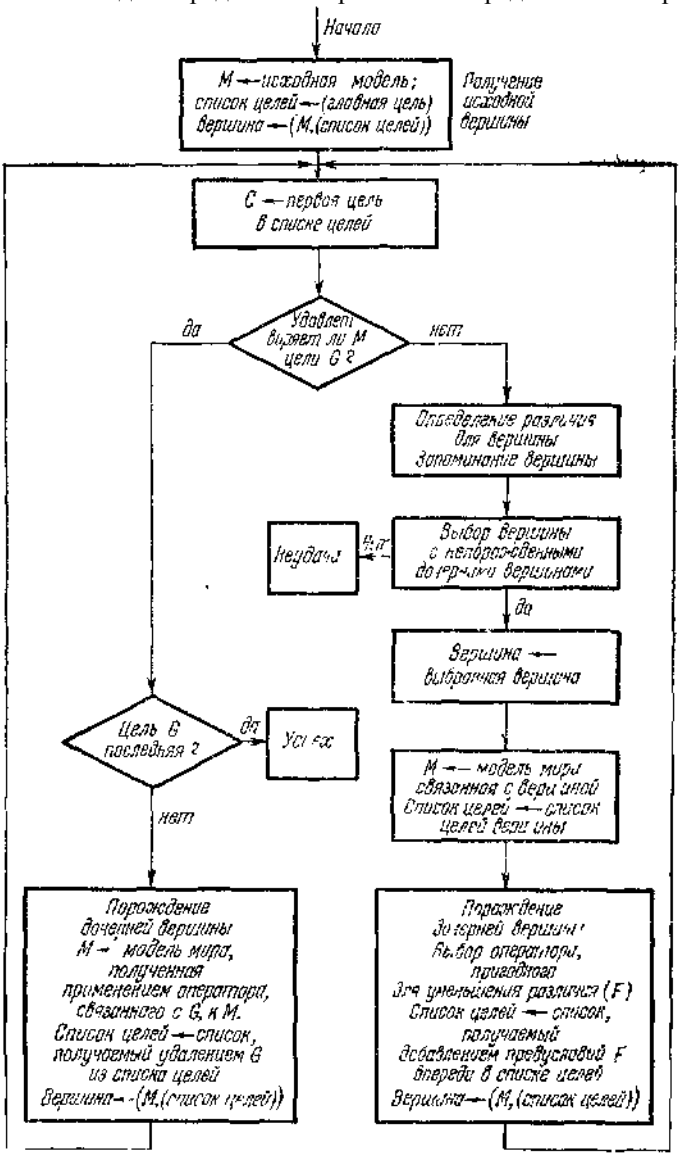


Рис. 10.1. Схема работы планирующей системы P3.

Мы рассмотрим принципы работы системы на примере и дадим для этого примера дерево поиска. Мир отправвителя изображен на рис.10.2.

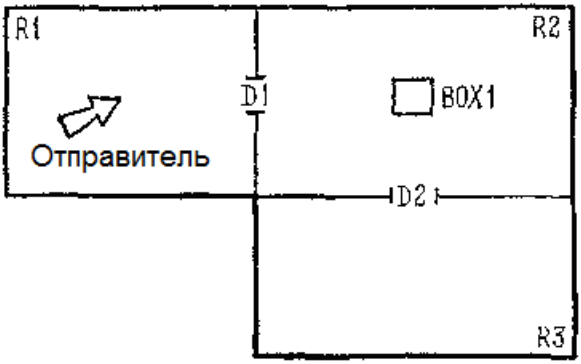


Рис. 10.2. Мир отправителя для примера задачи, решаемой ПС РЗ.

Пусть перед отправителем ставится задача передвинуть ящик из комнаты R2 в комнату R1.
Начальная модель M_o.
M_o: INROOM (Отправитель, R1) — отправитель находится в комнате R1,
CONNECTS (Dl, Rl, R2) — комнаты R1 и R2 соединены дверью D1.
CONNECTS (D2, R2, R3) — комнаты R2 и R3 соединены дверью D2.
BOX (BOX1) — ящик есть BOX1.
INROOM (BOX1, R2)— ящик BOX1 находится в комнате R2.
$(\forall x \forall y \forall z)$ [CONNECTS (x, y, z)→CONNECTS (x, y, z)]— порядок комнат в предикате CONNECTS безразличен.
Целевая ппф G_o.
G_o: $(\exists x)$[BOX(x) $\wedge$ INROOM (x, Rl)].
Нам задан список из двух операторов:
1) GOTHRU(d, $r1$, $r2$) — отправитель идет через дверь d из комнаты $r1$ в комнату $r2$ (d, $r1$, $r2$— параметры).
Предусловия:
INROOM (Отправитель, rl) $\wedge$ CONNECTS (d, rl, r2).
Список вычеркиваний:
INROOM (Отправитель, $), где $ может принимать любые значения.
Список добавлений:
INROOM (Отправитель, $r2$).

2) PUSHTHRU *(b, d, r*1*, r2)* — отправитель толкает объект *b* через дверь *d* из комнаты *r*1 в комнату *r2 (b, d, r*1*, r2* — параметры).
Предусловия:
INROOM *(b,* rl) ∧ INROOM (Отправитель, *r*1) ∧ CONNECTS (*d, r*l, *r*2).
Список вычеркиваний:
INROOM (Отправитель, $);
INROOM *(b,* $).
Список добавлений:
INROOM (Отправитель, r2);
INROOM (*b,* r2).
Поиск решения начинается с попытки доказать, что G_0 следует из M_o Эта попытка терпит неудачу, однако часть целевой ппф BOX *(x)* удовлетворяется при *x*=BOX1. Поэтому определяется различие между G_0 и M_0 в виде INROOM (BOX1, Rl). Система определяет, что частичная подстановка PUSHTHRU (BOX1, *d,* rl, Rl) может обеспечить это предписание, так как в его списке добавлений имеется INROOM *(b, r2).* Тогда в качестве новой цели G1 устапавливаются предусловия этого оператора с соответствующими подстановками параметров, т. е.
Gl: INROOM(BOX1, *r*1) Л INROOM (Отправитель, *r*1) ∧
∧ CONNECTS(*d, r*l, Rl)
Gl не может быть доказано из M_o и определяется различие между ними в виде INROOM (Отправитель, R2), так как при *r*1= R2 и *d=D*l остальные литералы G1 имеются в M_o. Система устанавливает, что для устранения этого различия уместен оператор GOTHRU при r2=R2 Устанавливается очередная подцель
G2 INROOM (Отправитель, rl) ∧ CONNECTS(*d, r*l, R2), которая может быть выведена из M_o при rl=Rl и *d=D*l Поэтому к M_0 применяется GOTHRU (D1, Rl, R2), преобразуя ее в
Ml: INROOM (Отправитель, R2);
CONNECTS (Dl, Rl, R2);
CONNECTS (D2, R2, R3);
BOX (BOX 1);
INROOM (BOX 1, R2);
(∀ *x* ∀ *y* ∀ *z)*[CONNECTS *(x, y,* z)→CONNECTS (*x, z,y*)].
Теперь делается попытка доказать Gl из новой модели Ml. Эта попытка приводит к успеху при rl=R2, *d=D*1. Таким образом, к Ml применяется оператор
PUSHTHRU (BOX1, Dl, R2, Rl),
так как остальные подстановки были сделаны ранее. Модель Ml преобразуется в
M2: INROOM (Отправитель, Rl);

CONNECTS (Dl, Rl, R2);
CONNECTS (D2, R2, R3);
BOX (BOX 1);
INROOM (BOX 1, Rl),
$(\forall x \forall y \forall z)$ [CONNECTS $(x, y, z) \rightarrow$ CONNECTS (x, z, y)].

Теперь делается попытка доказать G_o из M2. Эта попытка оказывается успешной, так что решением является план

GOTHRU (Dl, Rl, R2); PUSHTIIRU(BOX1, Dl, R2, Rl). (10.4)

Дерево поиска для этой задачи представлено на рис. 10 3.

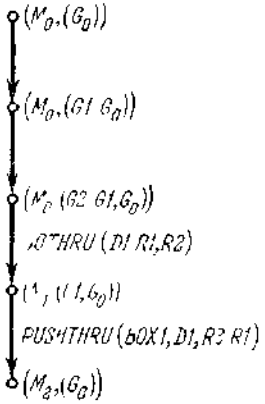


Рис. 10.3. Дерево поиска для примера задачи

В данном случае наше дерево выродилось в путь, поскольку в примере не возникло альтернативных возможностей (выбор оператора GOTHRU, а не PUSHTHRU, на втором шаге, объясняется точным совпадением различия со списком добавлений GOTHRU). Естественно, что в более сложных задачах и при большем списке операторов деревья поиска могут быть существенно «ветвистее».

Следует отметить два обстоятельства, не затронутые нашим простым примером.

Во-первых, порядок образования дуг дерева поиска, соответствующих применению операторов, вовсе не предопределяет вхождение операюров в этом порядке в окончательный план. Это лишний раз подчеркивает гибкость стратегии поиска механизма редукции GPS, сочетающего в себе свойства как прямого, так и обратного поиска.

Во-вторых, ПС РЗ снабжена эвристическим механизмом выбора узлов дерева поиска. В системе используется оценочная функция, учитывающая такие факторы, как число оставшихся целей в списке

целей, число и тип предикатов в оставшихся выражениях цели, а также сложность различий, связанных с данным узлом

Как отмечалось ранее, главной трудностью использования механизма редукции GPS является выбор операторов, пригодных для устранения или уменьшения различий. В рассматриваемой системе этот вопрос решается следующим образом Предположим, что в дереве поиска образован узел *(M,* (G_i, G_{i-1},…, G_0)*)*, причем система доказательства предписаний пытается доказать невыполнимость множества $M \bigcup (\sim G_i\}$. Если доказательство успешно, то к модели *M* применяется оператор с предусловиями G_i. Если же в течение определенного времени опровержение не будет найдено, то незавершенное доказательство или, в случае, если оно велико, его часть, выбираемая из эвристических соображений, и берется в качестве различия между *M* и G_i и связывается с данным узлом

Процесс выбора пригодного оператора происходит в два шага. На первом шаге создается упорядоченный список операторов-кандидатов. Выбор кандидатов основан на простом сравнении предикатов в различии с предикатами списков добавлений операторов. На втором шаге программа доказательства предписаний определяет, могут ли директивы из списка добавлений оператора резольвировать с директивами в различиях. Если новые резольвенты являются производными от директив в списке добавлений, то соответствующий оператор объявляется пригодным. Заметим, что из одного оператора-схемы может быть образовано несколько пригодных частных случаев.

Ранее мы отмечали трудности решения проблемы границ для декларативных предписаний. Представление результатов применения операторов в виде списков добавлений и списков вычеркиваний решает часть этой проблемы в духе метода контекстов и контекстных графов. Однако, как отмечалось, проблема выведенных фактов этим методом не решается, так что в общем случае требуется вновь выводить их в каждом новом состоянии, т. е. в каждой новой модели. Эта проблема разрешается в ПС РЗ путем задания множества простейших предикатов, имеющих наинизший ранг в смысле, указанном в п. 10.1.1. Остальные предикаты связываются с этим простейшим множеством. Другими словами, любой производный дизъюнкт связывается с теми из простейших предикатов, от которых зависит его истинность. Таким образом, предикат On(B2, B1) должен быть связан с предикатом At (B1, *A)*. Тогда истинность всех производных дизъюнктов может быть определена непосредственно из списков вычеркивания соответствующих операторов.

10.3. Обобщение планов и планирование с помощью макрооператоров

10.3.1. Представление планов

Задача обобщения планов состоит в том, чтобы после построения успешного плана преобразовать этот конкретный план в план, который мог бы быть затем использован для множества подобных задач. Другими словами, мы хотим получить план-схему, т. е. **параметризованное семейство планов**. При этом необходимо, чтобы такой обобщенный план мог быть использован как в последующих процессах планирования в качестве дополнительного к имеющемуся списку операторов, так и при выполнении планов, составленных из таких обобщенных планов-операторов, с помощью ИС.

Прежде всего необходимо определить формализм представления планов, в понятиях которого можно было бы описать и решить поставленную задачу.

Таким формализмом является *треугольная таблица* (ТТ), строкам и столбцам которой соответствуют операторы плана. Пример ТТ представлен на рис. 10.4.

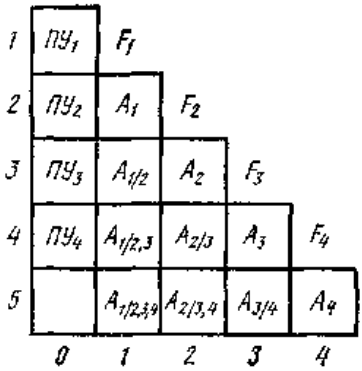


Рис. 10.4. Треугольная таблица.

Столбцы ТТ, кроме нулевого, помечены операторами F_1, F_2, F_3, F_4 составляющими план. Обозначим через $c_{i,j}$ ячейку ТТ, i— номер столбца, j — номер строки.

В общем случае для каждого столбца i, $i \neq 0$, в ячейку $c_{i,\ i+1}$ помещается *список добавлений* A_i *оператора* F_i. В ячейки $c_{i,\ i+k}$, k=2, 3, ... n+1—i (n— число операторов в плане), помещаются те предписания списка добавлений оператора F_i, которые остаются неизменными после применения операторов F_{i+1}, F_{i+2}, ..., F_{i+k-1}. Они обозначаются через $A_{i,\ i+1, i+2, \ldots, i+k-1}$. В нашем примере $A_{1/2}$ означает те предписания A_1, которые остались неизменными после применения F_2, $A_{1/2,3}$ — те предписания $A_{1/2}$, которые остались неизменными после применения F_3 и т. д.

Рассмотрим теперь строку j ТТ. Эта строка (исключая ячейку $c_{0,\ j}$) содержит список добавлений, получаемый применением последовательно F_1, F_2, . . ., F_{j-1}, или (j—1)*-й «шапкой» плана.* В частности, (n+1)-я строка содержит список добавлений, полученный после выполнения всего плана.

Назовем множество предписаний, используемых для доказательства предусловий оператора, *поддержкой предусловий.* Необходимо, чтобы строка j содержала все ппф в поддержке предусловий F_j.

Часть таких ппф будет добавлена (j—1)-й шапкой плана и поэтому будет включена в строку j. Остающиеся необходимые ппф находятся в начальной модели и не были вычеркнуты ни одним из F_k, k=1, 2, . . ., j—1. Эти предписания, обозначенные нами ПУ$_j$, и помещаются в столбце i, i=0. Следовательно, нулевой столбец ТТ содержит те предписания из начальной модели, которые были использованы в доказательствах предусловий для плана. Отметим, что ПУ$_j$, j=1,2, . . ., n, отнюдь не содержат полное описание начальной модели.

Назовем предписания, входящие в поддержку предусловий F_j, j=1, 2, . . ., n, *отмеченными предписаниями.* По построению ТТ все предписания в ячейке $c_{0,\ j}$, j=1, 2,. . . . . ., n, являются отмеченными. Однако не обязательно все предписания в $c_{0,\ j}$, $i \neq 0$, являются отмеченными.

На рис. 10.5 приведена ТТ для плана решения задачи, рассмотренной в § 10.2. Звездочка указывает отмеченные предписания.

*INROOM(Robot,R1) *CONNECTS(D1,R1,R2)	GOTHRU(D1,R1,R2)	
*INROOM(BOX1,R2) *CONNECTS(D1,R1,R2) *CONNECTS(x,y,z)→CONNECTS(x,z,y)	*INROOM(Отправ,R2)	PUSHTHRU(BOX1,D1,R2,R1)
		INROOM(Отправ,R1) INROOM(BOX1,R1)

Рис. 10.5. Треугольная таблица для примера (§ 10.2).

Таким образом, отмеченные предписания в j-й строке составляют поддержку предусловий F_j. Представляет интерес исследование условий применимости для последовательности операторов $F_j, F_{j+1}. . ., F_n$, т. е. *j-го остатка плана.* Очевидно, что j-й остаток плана применим к модели, если она содержит ту часть поддержки предусловий F_k, $k=j, j+1, \ldots, n$, которая не вырабатывается внутри самого остатка.

Назовем *j-м ядром ТТ* такую прямоугольную подтаблицу ТТ, что она содержит ячейку $c_{0,\ n+i}$ и строку j. Мы утверждаем, что j-й остаток плана применим к модели, если все отмеченные предписания в j-м ядре истинны в этой модели.

Действительно, **если все отмеченные предписания в *j*-м ядре истинны, то F_j применим к модели**. В результате применения F_j получится новая модель, в которой будут истинны предписания A_j. Поскольку по построению таблицы и отмеченные предписания внутри j-го ядра истинны, то все отмеченные предписания в строке $j+1$ истинны; следовательно, F_{j+1} также применим. Продолжая аналогичные рассуждения относительно строк $j+2, . . ., n$, получаем, что j-й остаток плана применим к модели.

Таким образом, доказано достаточное условие применимости j-го остатка плана.

Заметим, что первое ядро устанавливает достаточные условия применимости плана в целом, т. е. конъюнкция всех предписаний в нулевом столбце представлюет собой предисловия для всего плана. В свете доказанного достаточного условия выполнение плана может рассматриваться как последовательное преобразование ядер ТТ, т. е. для всех j, $F_j(K_j)=K_{j+1}$, где K_j — j-е ядро ТТ.

10.3.2. Обобщение планов

Опишем процедуру обобщения плана, представленного в виде ТТ. Эта процедура осуществляется в три шага.

На первом шаге мы преобразуем исходный конкретный план в наиболее общую форму. Этот шаг осуществляется следующим образом:

1) Каждое появление константы в первом ядре, в том числе одной и той же, замещается отдельным параметром.

2) Остальные столбцы озаглавливаются описаниями операторов-схем с параметрами.

3) В предписаниях в столбцах *i*, i=l, 2, . . ., *n,* исходного плана все константы заменяются соответствующими параметрами операторов-схем F_i.

В нашем примере ТТ (рис. 10.5) преобразуется в ТТ рис. 10.6).

*INROOM(p1,p2) * CONNECTS(p3,p4,p5)	GOTHRU(p11,p12,p13)	
*INROOM (p6,p7) * CONNECTS (p8,p9,p10) * CONNECTS(x,y,z)→CONNECTS (x,z,y)	*INROOM(Отправ,p13)	PUSHTHRU(p14,p15,p16,p17)
		INROOM(Отправ,p17) INROOM(p14,p17)

Рис.10.6. Треугольная таблица для примера (§ 10.2) после первого шага процедуры обобщения.

На втором шаге мы вводим такие ограничения на полученные параметры, чтобы, с одной стороны, отмеченные предписания в строке *j*, *j*=1, 2, . . ., *n,* были поддержкой оператора F_j и, с другой стороны, чтобы исходный план оставался частным случаем получающейся ТТ. Этот шаг осуществляется следующим образом:

1) Извлекаются графы опровержения, построенные в процессе доказательства предусловий всех операторов исходного плана.

2) Для каждого такого графа, соответствующего оператору F_j, строится изоморфный образ выполнением на каждом шаге резолюций тех же предписаний и унификаций тех же литер, причем в качестве аксиом используются отмеченные предписания из строки *j*, а в качестве доказываемого предписания — предусловия оператора-схемы F_j из ТТ, полученной на первом шаге.

3) Все получаемые в процессе доказательства подстановки вносятся в полученную на первом шаге ТТ.

На рис 10.7 и 10.8 приведены деревья опровержения для предусловий операторов ТТ (рис. 10.6).

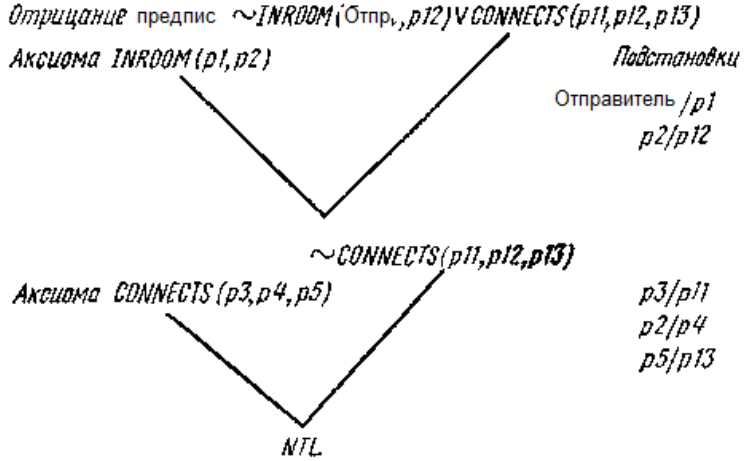


Рис. 10.7. Дерево опровержения для предусловий оператора GOTHRU (p11, p12, p13).

Рис. 10. 8. Дерево опровержения для предусловий оператора PUSHTHRU (p14, p15, p16, p17).

После соответствующих подстановок ТТ (рис. 10. 6) преобразуется в ТТ (рис. 10.9),

*INROOM(Отпр, p2) *CONNECTS(p3,p2,p5)	GOTHRU(p3,p2,p5)	
*INROOM(p6,p5) *CONNECTS(p8,p9,p5) *CONNECTS(x,y,z)→CON-NECTS(x,z,y)	*INROOM(Отпр$_v$,p5)	PUSHTHRU(p6,p8,p5,p9)
		INROOM(Отпр,p9) INROOM(p6,p9)

Рис. 10. 9. Обобщенная треугольная таблица для примера (10.2).

На третьем шаге мы проводим два дополнительных преобразования, имеющие своей целью исключить излишние обобщения, а также один вид возможного противоречия.

Источником излишнего обобщения является случай, когда одно и то же предписание в начальной модели плана входит в поддержку более чем одного оператора. Тогда на первом шаге процедуры обобщения это предписание преобразуется в два различных предписания. Во многих случаях это приводит к полезным обобщениям. Так, например, произошло в нашем примере (рис. 10.9), когда CONNECTS (D1, Rl, R2) преобразовалось в CONNECTS *(p*3*, p2, p5)* и CONNECTS *(p8, p9, p5),* что обобщило исходный план в том отношении, что теперь отправитель может передвинуть ящик в некоторую третью комнату, а не обязательно в ту, в которой первоначально находился сам.

Однако, если бы различные параметры (в нашем примере *p2* и p9), полученные из одной константы, не использовались бы оба как аргументы операторов в плане, их без ущерба можно было бы связать вместе с последующей подстановкой одного вместо другого в ТТ.

Источником возможных противоречий в обобщенной ТТ является сделанное нами на шаге 1, п. 3, молчаливое предположение о том, что вычеркивания в наиболее общей ТТ могут быть теми же самыми, что и в исходной ТТ.

Рассмотрим случай, когда в исходной ТТ имеются два одинаковых предиката с различными аргументами (например, At (BOX1, Rl) и

At(BOX2, R2)). На первом шаге обобщения они превратятся в одинаковые предикаты с различными параметрами (например, *At(p1, p2)* и At(p3, p4)). Предположим, что в результате второго шага обобщения эти параметры продолжают оставаться различными, хотя, возможно, и отличаются от *p1, p2* и *p3, p*4 соответственно. Далее, вполне возможно, что при использовании обобщенного плана и нахождении его частного случая часть соответствующих параметров в предикатах будет связана с одной и той же константой, в то время как другая часть соответствующих параметров предикатов свяжется с различными константами. Это и приводит к противоречию. В приведенном выше примере эта ситуация возникла бы, если бы *p1* и p3 были связаны с одним объектом, скажем, BOX7, а *p2* и *p4* — с различными местоположениями (R4 и R8 соответственно). В результате мог бы быть доказан план (или его часть), в котором BOX7 одновременно находился бы в R4 и R8. Ясно, что в конкретном плане предикат At(pl, *p2)* был бы вычеркнут, прежде чем была установлена истинность предиката At(p3, *p4).*

В общем случае мы должны провести коррекшровку обобщенной ТТ путем анализа списков вычеркивания всех операторов, входящих в план. Алгоритм коррекции обобщенной ТТ работает с полученной на втором шаге таблицей и анализирует последовательно списки вычеркивания операторов F_1, F_2, ..., F_n. Рассмотрим совместно высказывания в строке j и список вычеркиваний F_j. Очевидно, что применение списка вычеркивания к строке будет менять ее тогда, когда некоторые из предписаний строки будут унифицироваться с некоторыми из литер списка вычеркиваний и когда для унификации потребуется, чтобы некоторый параметр *p1* замещался другим параметром p2 или константой C. Если $p1=p2$ или $p1=C$, то предписание R вычеркивается, иначе оно остается неизменным и переходит в строку j+1. Это условное вычеркивание разрешается путем замещения предписания в строке j+1 одной из импликаций:

а) $p1 \neq p2 \rightarrow R$,

б) $p1 \neq C \rightarrow R$. (10.5)

Предположим далее, что замещенное предписание в (j+1)-й строке отмечено, т. е. входит в поддержку предусловий F_{j+1} (если оно не отмечено, алгоритм заканчивает работу). Тогда для того, чтобы поддержка предусловий F_{j+i} сохранилась после введения одной из импликаций (10.5), мы должны добавить в ячейку $c_{0,\ j+1}$ отмеченное предписание вида

а) $p1 \neq p2$

или (10.6)

б) $p1 \neq C$,

так что прежнее в предписание легко выводится из (10.5, *а)* и (10.6, *а)* или (10.5, *б)* и (10.6, *б).*

На этом третий шаг обобщения плана завершается, и полученный обобщенный план может быть занесен в список операторов системы. Мы будем называть такой план *макрооператором.*

10.3.3. Особенности планирования с макрооператорами

Рассмотренный в предыдущем параграфе процесс образования макрооператоров представляет собой типичный пример обучения ПС в ходе решения ею задач.

Этот процесс обеспечивает непрерывное обогащение возможностей системы при условии, что она будет способна

1) использовать макрооператор и любую его часть в процессе планирования;

2) свести к минимуму в каждом конкретном использовании избыточность макрооператора, являющуюся естественным следствием его обобщенного предписания;

3) регулировать процесс накопления макрооператоров в системе.

Предположим, что в процессе планирования для уменьшения различия выбирается некоторый макрооператор, представляющий последовательность операторов $F_1, F_2,, \ldots, F_n$. Такой макрооператор содержит n различных *списков добавления* $A_{1,j}$, j=2, 3, . . ., n, соответствующих j-й шапке плана. Эти списки добавлений могут быть использованы обычным образом для выбора наиболее пригодного из них с позиции уменьшения различия. В случае выбора равнокачественных списков добавлений естественно выбирать список с наименьшим j, поскольку ему будет соответствовать более короткая последовательность.

Пусть выбран список добавлений $A_{1,j}$. Очевидно, что для целей планирования нас не интересует (j+1)-й остаток полного плана, так что из j-й шапки плана можно без ущерба удалить те операторы, которые предназначены для формирования предписаний, входящих в поддержку предусловий всех F_k, $k=j+1, \ldots, n$. Кроме того, для целей планирования не нужны и те операторы, результатом которых являются предписания, не используемые для установления пригодности списка добавлений $A_{1,j}$.

Эти соображения приводят нас к следующей формулировке алгоритма упрощения макрооператора:

1) Производится отметка тех предписаний из $A_{1,j}$, которые необходимы для установления пригодности $A_{1,j}$ (т. е. входят в неполный вывод, являющийся различием).
2) Снимаются метки у всех предписаний, являющихся поддержкой предусловий F_{j+1}.
3) Находится первый столбец справа, начиная с j-го, не содержащий отмеченных предписаний. Пусть таким столбцом будет i_k-й, $i_k \leq j$. Тогда снимаются метки у всех предписаний в i_k-й строке, а оператор F_{i_k} исключается из j-й шапки плана.
4) Процесс, указанный в п. 3), повторяется вплоть до первою столбца. Оставшаяся в результате подпоследовательность операторов выдается как макрооператор, пригодный для уменьшения установленного различия.

Пример. Рассмотрим ТТ (рис. 10.10).

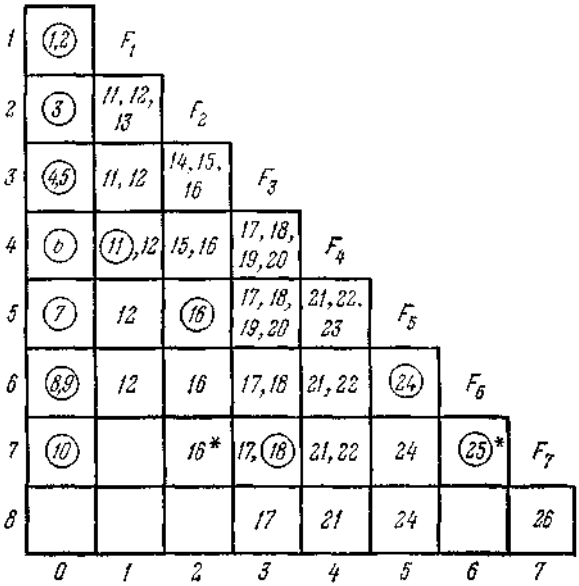


Рис. 10.10. Пример ТТ для иллюстрации алгоритма упрощения макрооператора.

Числа в ячейке представляют предписания, отмеченные предписания обведены кружками, предписания, пригодные для уменьшения различий, отмечены звездочками. Структура ТТ может быть прослежена с помощью таблицы 10.3, где I — начальное, а G —- конечное состояние плана.

Таблица 10.3

Оператор	Источник поддержки предусловий	Адресат поддержки предусловий	Оператор	Источник поддержки предусловий	Адресат поддержки предусловий
F_1 F_2 F_3 F_4	I I I I, F_1	F_4 F_5 F_7, G G	F_5 F_6 F_7	I, F_2 $\neq I$, F_5 I, F_3, F_6	F_6, G F_7 G

В нашем примере j=6. Алгоритм работает следующим образом (предписания 16 и 25 отмечены как уменьшающие различие).

1) Снимается метка у предписания 18.

2) Столбец 4 — первый справа, не содержащий отмеченных предписаний. Снимается метка у предписания 11, оператор F_4 исключается.

3) Столбец 3 — следующий столбец, не содержащий отмеченных предписаний (метка у предписания 18 уже снята). Оператор F_3 исключается.

4) Столбец 1 — следующий, не содержащий отмеченных предписаний (метка у предписания 11 уже снята). Исключается оператор F_1.

В результате получается последовательность операторов F_2, F_5, F_6, представляющая собой упрощенный макрооператор. На рис. 10.11 представлена ТТ для этого макрооператора, а таблица 10.4 показывает структуру ТТ (рис. 10.11).

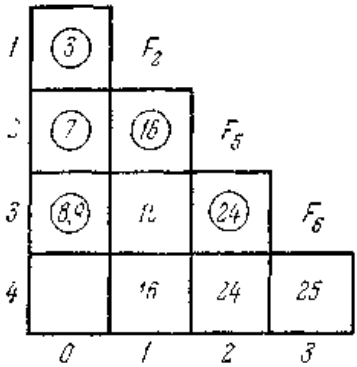


Рис. 10.11. Результат упрощения ТТ (рис. 10.10)

Таблица 10.4

Оператор	Источник поддержки предусловий	Адресат поддержки предусловий
F_2	I	F_5, G
F_5	I, F_2	F_6, G
F_6	I, F_5	G

Отметим, что мы не снимали метки у предписаний в нулевом столбце, так как соответствующие предписания исключаются автоматически при исключении операторов.

Рассмотрим теперь, как может применяться упрощенный макрооператор для преобразования моделей. Главный вопрос состоит в выборе предусловий для макрооператора.

Если бы мы ограничились применением полного плана, выраженного макрооператором, то очевидно, что в качестве предусловий следовало бы выбрать конъюнкцию предписаний в первом ядре ТТ. Однако в случае, если весь макрооператор неприменим, мы хотим использовать частичную возможность применения макрооператора, поскольку она тоже может привести к полезным преобразованиям модели. Таким образом, мы приходим к необходимости установления предусловий для каждого остатка макрооператора. В п. 10.3.1 мы установили достаточное условие применимости j-го остатка плана. Таким образом, мы должны установить истинность всех отмеченных предписаний в j-м ядре. Это может быть осуществлено с помощью процедуры сканирования ячеек ТТ.

Идея процедуры заключается в следующем. Чем меньше i и чем больше j, тем в большее число ядер одновременно входит ячейка $c_{i,j}$ ТТ. Таким образом, просматривая все ячейки $c_{i,j}$ в порядке возрастания i и убывания j и проверяя истинность всех отмеченных предписаний в этих ячейках доказательством их из текущей модели, мы будем последовательно находить те ядра в порядке убывания их номеров, все отмеченные предписания в которых истинны (если такие ядра имеются). Как только мы находим некоторую ячейку $c_{i,j}$, в которой хотя бы одно предписание не может быть доказано из текущей модели, мы исключаем из рассмотрения все ячейки c_k,j, $k \geq i$, так как при этом мы можем найти требуемое ядро с номером не более, чем i—1.

По мере доказательства высказываний в ячейках ТТ из текущей модели мы осуществляем соответствующие подстановки параметров макрооператора. Важно заметить, что сделанные для доказательства высказываний в одной ячейке подстановки должны немедленно распространиться на всю таблицу. Если имеются альтернативные возможности, то следует хранить дерево возможностей и, в случае необходимости, осуществлять возврат к точке ветвления.

С помощью описанной процедуры сканирования ТТ мы находим j-й остаток плана с наивысшим j, применимый к модели. В этом случае последовательность операторов $F_j, F_{j+1}, \ldots, F_n$ может быть выполнена, преобразуя текущую модель в некоторую новую модель. Если же неприменим весь план (т. е. ни один из j-х остатков плана неприменим, j==1,2,..., n), то конъюнкция отмеченных предписаний в первом ядре устанавливается как новая подцель.

Чтобы завершить изложение особенностей планирования с помощью макрооператоров, нам необходимо описать возможные подходы к решению задачи регулирования количества макрооператоров в системе. Хотя, казалось бы, богатый набор макрооператоров повышает возможности системы, наличие множества альтернативных решений задачи всегда будет в определенной степени снижать эффективность поиска плана. Конечно, трудно указать точно границы разумного компромисса между указанными факторами. Во всяком случае, есть два очевидных пути разумного сокращения количества макрооператоров в ПС.

Первый из них — исключение тех макрооператоров, которые являются частью более общих макрооператоров. Это исключение не только разумно, но и необходимо, поскольку ПС, работающая с макрооператорами, все время порождает подобные случаи.

Действительно, пусть решая задачу, ПС включает в план некоторую часть макрооператора или весь макрооператор. Получив полный план решения задачи, ПС обобщает его, порождая новый макрооператор, включающий в себя весь старый макрооиератор пли его часть. Если подстановки параметров в старом и новом макрооператорах соответствуют друг другу, то очевидно, что каждый список добавлений старого макрооператора является подмножеством некоторого списка добавлений нового макрооператора. В этом случае старый макрооператор может быть исключен из системы. Аналогичным образом следует поступать и в случае использования в новом макрооператоре частей старого, исключая, однако, при этом те списки добавлений нового макрооператора, которые являются подмножествами списков добавлений уже существующего макрооператора.

В общем процедура исключения избыточных списков добавлений формулируется следующим образом: если любой частный случай *j*-й шапки одной ТТ является также частным случаем некоторой последовательности операторов в этой же или любой другой ТТ, то все списки добавлений *j*-й шапки (т. е. начиная со 2-й и кончая (*j*+1)-й строкой) ТТ исключаются.

П р и м е р. Рассмотрим две последовательности операторов:

A: $F_1(p1)$, $F_2(p1,p2)$, $F_3(p3)$, $F_4(p3,C1)$, $F_1(p3)$, $F_2(p4, p5)$.

B: $F_3(p6)$, $F_4(p6, \mathrm{Cl})$, $F_1(p7)$, $F_5(p6, p7)$.

1) $F_1(p1)$ и $F_2(p1, p2)$ в последовательности *A* могут быть исключены, так как в конце плана есть $F_1(p3)$ и $F_2(p4, p5)$.

2) $F_3(p6)$ и $F_4(p6, \mathrm{Cl})$ в последовательности *B* могут быть исключены, так как в последовательности *A* имеются $F_3(p3)$ и $F_4(p3, C1)$.

3) $F_1(p7)$ в последовательности *B* не может быть исключено последовательностью *A*, так как $F_3(p6)$, $F_4(p6, C1)$, $F_1(p7)$ и $F_3(p3)$, $F_4(p3, C1)$, $F_1(p3)$ могут иметь разные частные случаи.

Следует отметить, что определенная выше процедура исключения не учитывает случая одинаковых операторов, включенных в разные макрооператоры, но имеющих различные поддержки предусловий. Вообще говоря, ни один из таких операторов не должен исключаться.

Второй путь сокращения количества макрооператоров в ПС — введение механизмов «забывания». Подход к решению этой задачи заключается в наборе статистики использования макрооператоров и исключении тех из них, которые используются реже, чем заранее установленный порог частоты использования.

10.4. Обобщение пространств поиска решений и планирование в абстрактных пространствах

10.4.1. Принцип образования иерархии пространств

Обобщение планов путем автоматического создания макрооператоров является важным шагом на пути к созданию универсальных и эвристически эффективных решателей задач, обладающих свойством самоусовершенствования. Однако накопление набора методов решения задач и укрупнение самих методов составляют лишь часть общей проблемы функционирования ИСФ. Мы отмечали, что решающим шагом на пути к созданию мощных ИСФ является глобальное исследование пространства поиска, имеющее своей целью «понимание» решателем задач общих закономерностей этого

пространства. Обращаясь к рассмотренной ранее задаче о миссионерах и людоедах (см. книгу1), мы видим, что укрупнение операторов позволило нам лишь исключить некоторые промежуточные состояния. Наиболее существенный выигрыш был получен за счет введения в пространстве поиска решения некоторого комплексного образа, позволившего решить задачу на абстрактном уровне.

Главным недостатком метода обобщения планов (п. 10.3.2) является тот факт, что, укрупняя макрооператоры, мы оставляем другой элемент предписания — модели — на том же, весьма детальном уровне, что и в случае обычных оператором. В понятиях формальной модели, последовательное обобщение макрооператоров приводит к настолько большим конъюнкциям предусловий, составляющих первое ядро ТТ, и настолько большим спискам добавлений, что эффективность решения достаточно сложных задач может оказаться весьма малой.

Поэтому для планирования решения сложных задач в сложных областях необходимо описывать области в некотором существенно обобщенном пространстве, которое, значительно упрощая описание области, игнорировало бы незначительные детали. При этом полный процесс решения задачи можно было бы представить как пошаговый процесс в иерархии пространств от наиболее абстрактного к базовому пространству, в котором получалось бы окончательное решение. Существенной характеристикой такого процесса является поиск приблизительного наброска решения задачи в абстрактном пространстве и, при его достаточной перспективности, преобразование этого пространства в пространство более низкого уровня, уточнение наброска решения и т. д. Такой процесс должен значительно увеличивать эвристическую эффективность системы, в то же время допуская достаточно легкое преобразование предписаний от абстрактного к базовому.

Рассмотрим, каким образом приведенные выше соображения могут быть преломлены в контексте описания моделей области с помощью ппф в исчислении предикатов первого порядка и описания операторов с помощью предусловий, списков вычеркивания и добавления, также выраженных в виде ппф. Прежде всего возникает вопрос, каким образом следует отличать пространства друг от друга. Очевидно, что было бы целесообразно опускать часть описаний моделей и операторов. Однако нам необходим критерий «детальности» выражений, входящих в эти описания. Таким критерием может быть **ранжирование фактов об области по степени их неизменности** (см. п. 10.1.1.). Эвристическим основанием использования такого ранжирования являются следующие соображения. С точки зрения построения плана существование предметов (истин) и их свойства

(наличие комнат и ящиков, расположение дверей) являются более важными фактами, чем положение предметов, которые могут передвигаться отправителем и тем более, чем положение самого отправителя (отправитель не умеет строить двери и ящики!). Поэтому именно эти важные факты должны использоваться в абстрактном пространстве для частичных описаний модели, и после построения наброска плана он может наполняться такими подробностями, как положение предметов и т. д.

Второй вопрос заключается в том, какие из элементов описания моделей и операторов ранжировать по их неизменности. По-видимому, нет смысла разбивать на классы предписания, относящиеся к описанию модели: второстепенные детали в модели могут просто игнорироваться. Точно так же нет особого смысла ранжировать списки добавлений и вычеркиваний операторов: те из операторов, действие которых сводится к изменению деталей, не будут выбираться как пригодные в пространствах высокого уровня. Кроме того, изменения в списках вычеркивания и добавления приведут к тому, что в различных пространствах результаты действий операторов окажутся весьма различными, поэтому преобразования между пространствами могут оказаться сложными.

В то же время, ранжируя предусловия операторов, мы добиваемся весьма важного результата: **в пространствах высокого уровня мы будем всегда выбирать те операторы, которые производят наиболее существенные преобразования моделей области**.

Таким образом, мы выбираем предусловия оператора как средство различения уровня пространства поиска решения. Могут быть предложены различные методы присвоения весов литерам, входящим в предусловия операторов. Один из них заключается в следующем.

На все предикаты, описывающие область функционирования ИСФ, накладывается некоторое частичное упорядочение, которое определяет порядок, в котором будут исследоваться литеры предусловий всех операторов, используемых в этой области. Исследование литер происходит следующим образом:

1) Всем литерам, чье значение истинности не может быть изменено никаким оператором, присваивается максимальный вес.

2) Оставшиеся литеры исследуются по порядку. Если литера может быть достигнута из состояния, где все ранее исследованные литеры истинны, с помощью короткого плана, она считается деталью и ей присваивается вес, равный ее рангу в частичном упорядочении. Если такого плана нет, то ей присваивается вес, больший, чем наивысший ранг в частичном упорядочении.

10.4.2. Планирование в иерархии пространств

Рассмотрим возможную схему процесса планирования в образованной взвешиванием литер предусловии операторов иерархии пространств. Схема системы, приведенная на рис. 10.12, носит рекурсивный характер.

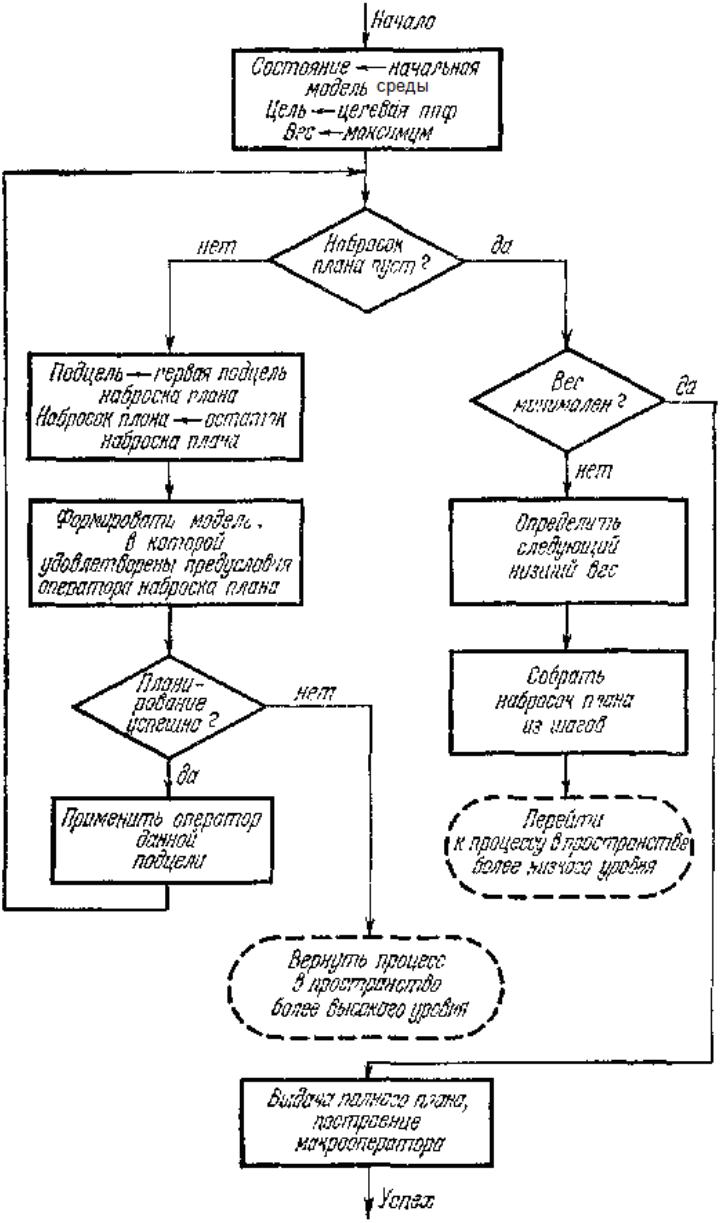


Рис. 10. 12. Схема работы планирующей системы в иерархии пространств.

Сначала схема работает в абстрактном пространстве высшего уровня, строя набросок плана достижения главной цели. Когда этот план найден, система определяет вес следующего более низкого по уровню пространства, в котором требуется планирование, и строит скелет (фрейм) вершин, к которым применялись операторы на высшем уровне пространства. Затем система вызывается рекурсивно, решая подзадачи соединения шагов в наброске плана и обеспечения применимости этих шагов в соответствующих точках нового плана. Когда все подзадачи решены, система вновь вызывает себя рекурсивно для планирования в пространстве еще более низкого уровня. Этот процесс продолжается, пока не будет построен полный план в базовом пространстве, т. е. пространстве самого низкого уровня.

Описанный процесс осуществляет планирование в каждом абстрактном пространстве вплоть до главной цели, прежде чем осуществится переход к планированию на более низком уровне. Если какая-либо из подзадач в некотором пространстве не может быть решена, то управление возвращается в пространство более высокого уровня, вершина, послужившая причиной неудачи, исключается из рассмотрения и продолжается поиск успешного плана.

Заметим, что действия системы при неудаче аналогичны описанному ранее механизму возврата к точке ветвления со всеми его преимуществами и недостатками. В связи с этим требования к качеству поиска на высших уровнях иерархии пространств должны быть весьма высоки. Описываемая система должна управляться оценочными функциями, позволяющими строить допустимые стратегии в смысле приведенного ранее определения, возможно, в ущерб эффективности поиска, поскольку каждый лишний шаг в абстрактном пространстве может привести к большому количеству лишних шагов в пространстве более низкого уровня. В частности, на самом высоком уровне планирования используется оценочная функция при $\omega=1/2$ с постепенным увеличением веса эвристической функции по мере перехода в пространства более низкого уровня.

Проблема равнокачесгвенных решений в пространствах высокого уровня также становится весьма серьезной, поскольку произвольный выбор уменьшения различий может привести к неправильным решениям в пространствах более низкого уровня, где при выяснении деталей эти решения могут оказаться не эквивалентными, что приведет к необходимости возврата на более высокий уровень планирования. Поэтому при наличии равнокачественных альтернатив решение откладывается до планирования на низшем уровне путем частичной подстановки значений параметров оператора.

10.4.3. Пример. На рис. 10.13 изображена область отправителя в начальном и конечном состоянии.

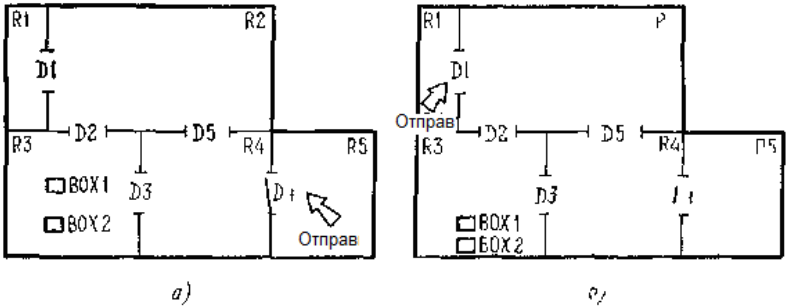


Рис. 10.13. Начальная (а) и конечная (б) область отправителя в примере (10.4.3).

Эта область достаточно сложна, однако работа системы может быть продемонстрирована только на примере повышенной сложности. В интересах краткости изложения мы не будем давать все формальные шаги построения плана, компенсируя эти пробелы словесным описанием. Мы хотим показать этим примером общий характер планирования в иерархии пространств. Нам нужны следующие 5 операторов-

F_1: PUSHB *(bx, by)* — передвинуть *bx* к объекту *by*.

Предусловия:

{6} TYPE *(by*, OBJECT) *(by* типа «объект»),

{6} PUSHABLE *(bx) (bx* можно передвигать),

($\exists$ *rx*)[{5} INROOM (*bx, rx)* $\wedge$ {5} INROOM *(by, rx)* $\wedge$

{5} INROOM (Отправитель, *rx)}*,

{1} NEXTTO (Отправитель, *bx)* (отправитель стоит вслед за *bx).*

Вычеркивания:

AT (Отправитель, $), NEXTTO (Отправитель, $), AT (*bx*,$), NEXTTO *(bx,* $), NEXTTO ($, *bx).*

Добавления:

*NEXTTO(by, *bx),* *NEXTTO(*bx, by),*

NEXTTO (Отправитель, *bx).*

F_2: GOTHRUDR *(dx, rx)* — идти через *dx* в *rx.*

Предусловия:

{6} TYPE(*dx*, DOOR) *(dx* типа «дверь»),

{6} TYPE(*rx*, ROOM) *(rx* типа «комната»),

{2} STATUS (*dx*, OPEN) (состояние двери «открыта»),
($\exists$ *ry*) [{5} INROOM (Отправитель, *ry)* $\wedge$ {6} CONNECTS (*dx*, *ry*, *rx*)].
Вычеркивания:
AT (Отправитель, $), NEXTTO (Отправитель, $),
INROOM (Отправитель, $).
Добавления
* INROOM (Отправитель, *rx).*
F_3*:* GOTOD *(dx)* — идти к двери *dx.*
Предусловия:
{6} TYPE(*dx*, DOOR),
($\exists$ *rx*)($\exists$ *ry*) [{5} INROOM (Отправитель, *rx*) $\wedge$
{6} CONNECTS (*dx*, *rx*, *ry)}.*
Вычеркивания:
AT (Отправитель, $), NEXTTO (Отправитель, $).
Добавления:
*NEXTTO (Отправитель, *dx).*
F_4*:* GOTOB *(bx)* — идти к объекту *bx.*
Предусловия:
{6} TYPE(*bx*, OBJECT),
($\exists$ *rx*)[{5} INROOM (bx, *rx)* $\wedge$
{5} INROOM (Отправитель, *rx*)]
Вычеркивания:
AT (Отправитель, $), NEXTTO (Отправитель, $).
Добавления:
* NEXTTO (Отправитель, *bx).*
F_5*:* OPEN *(dx)* — открыть дверь *dx.*
Предусловия:
{6} TYPE (*dx*, DOOR),
{5} STATUS (*dx*, CLOSED) (состояние двери «закрыта»),
{5} NEXTTO (Отправитель, *dx).*
Вычеркивания:
STATUS (*dx*, CLOSED).
Добавления:
*STATUS(*dx*, OPEN).

Примечания. 1. Числа, стоящие в фигурных скобках, соответствуют весу литер предусловий.

2. Звездочки, стоящие перед добавлениями, указывают первичные добавления, т. е. те, которые рассматриваются системой при поиске пригодного оператора.

Мы не приводим здесь полного описания начальной модели. Часть этого описания очевидна из рис. 10.13, *а.* Заметим, что все предикаты

типа TYPE, STATUS, CONNECTS, PUSHABLE входят в начальную модель. Кроме того, предполагается наличие двух аксиом:

($\forall$ x) (PUSHABLE (x) → TYPE (x, OBJECT)),

т. е. то, что можно передвинуть, является объектом, и

($\forall$ x) (STATUS (x, CLOSED)↔ ~ STATUS (x, OPEN)),

т. е. если дверь закрыта, то она не открыта, и наоборот.

Мы решаем задачу: «пододвинуть ящик BOX1 к ящику BOX2, отправителю перейти в комнату R1». Целевая ппф имеет вид

NEXTTO (BOX1, BOX2) $\wedge$ INROOM (Отправитель, Rl).

Поиск начинается в пространстве высшего уровня (веса 6). Ищется различие между целью и начальной моделью. Различием является сама целевая ппф. Пригодным для уменьшения первого члена различия является оператор F_1 при bx=BOX1, by=BOX2. Поскольку предусловия веса 6 истинны в начальной модели, оператор применяется, и в новой модели BOX1 придвинут к BOX2. Различие между целью и этой новой моделью—INROOM (Отправитель, Rl). Пригодным для его уменьшения является оператор F_2 при rx=Rl. Его предусловия в пространстве веса 6

TYPE (R1, ROOM) $\wedge$ TYPE *(dx,* DOOR) $\wedge$ *(*$\exists$ *ry)*CONNECTS*(dx, ry,* Rl)

удовлетворяются при dx=Dl. Применение F_2: GOTHRUDR (Dl, Rl) порождает модель, в которой удовлетворяется целевая ппф. Таким образом, мы получаем набросок плана в абстрактном пространстве высшего уровня в виде

PUSHB (BOX1, BOX2), GOTHRUDR (Dl, Rl). (10.7)

Производится переход к планированию в пространстве веса 5. Первая подцель — предусловия веса 5 первого оператора наброска плана, PUSHB (BOX1, BOX2). Различие между начальной моделью и предусловиями будет

INROOM (Отправитель, R3).

Пригодным для уменьшения этого различия является F_2 при rx=R3. Однако F_2 неприменим к начальной модели. Различиями являются INROOM (Отправитель, R2) или INROOM (Отправитель, R4). Для уменьшения первого различия уместен F_2 при rx=R2, однако он неприменим к начальной модели. Для уменьшения второго различия уместен F_2 при rx=R4 и он применим при dx =D4 Таким образом, применяется GOTHRUDR (D4, R4), формируя модель, к которой применим F_2 при rx=R3, dx=D3. Оператор GOTHRUDR (D3, R3) формирует модель, в которой удовлетворяют предусловия PUSHB (BOX1, BOX2) веса 5 Этот оператор применяется.

Тепечь устанавливается новая подцель — предусловия второго оператора (10.7). Сравнение текущей модели с его предусловиями вырабатывает различие INROOM (Отправитель, R2). Пригодным для

уменьшения этого различия является оператор F_2 при *rx*=R2. Этот оператор применим при *dx*=D2, так что GOTHRUDR (D2, R2) формирует новую модель, в которой подцель удовлетворена. Тогда применим GOTHRUDR (Dl, Rl), формируя модель, в которой вновь удовлетворяется целевая ппф. Мы получаем набросок плана в абстрактном пространстве веса 5:

GOTHRUDR (D4, R4); GOTHRUDR (D3; R3);
PUSHB (BOX1, BOX2); GOTHRUDR (D2, R2);
GOTHRUDR (Dl, Rl). (10.8)

Переходим к планированию в абстрактном пространстве веса 2.

Первая подцель — предусловия веса 2 первого оператора (10.8). Сравнение их с начальной моделью вырабатывает различие STATUS (D4, OPEN). Пригодным для уменьшения этого различия является оператор F_5 при *dx*=D4, однако он неприменим. Различие между его предусловиями и начальной моделью — NEXTTO (Отправитель, D4). Пригодным для уменьшения этого различия является F_3 при *dx=D4.* Этот оператор применим к начальной модели и преобразует ее в модель, в которой применим и F_5. Поскольку, как легко проверить, все остальные подзадачи непосредственно удовлетворены, мы приходим к модели, в которой удовлетворяется целевая ппф с помощью наброска плана в абстрактном пространстве веса 2:

GOTOD (D4); OPEN (D4); GOTHRUDR (D4, R4);
GOTHRUDR (D3, R3); PUSHB (BOX1, BOX2);
GOTHRUDR (D2, R2); GOTHRUDR (Dl, Rl). (10.9)

Переходим к планированию в базовом пространстве (веса 1). Первые четыре оператора (10.9) применяются по очереди, однако пятый оператор неприменим. Различием является NEXTTO (Отправитель, BOX1). Пригодным для уменьшения этого различия является F_4 при *bx* =BOXl. Этот оператор применим, вырабатывая модель, в которой применим PUSHB (BOX1, BOX2). Окончательный план выглядит следующим образом:

GOTOD (D4); OPEN (D4); GOTHRUDR (D4, R4);
GOTHRUDR (D3, R3); GOTOB (BOXl);
PUSHB (BOX 1, BOX2);
GOTHRUDR (D2, R2); GOTHRUDR (Dl, Rl). (10.10)

Если внимательно проследить за ходом построения плана, легко заметить, что система автоматически строит сначала критические части плана, а потом переходит к построению плана для легко преодолеваемых областей базового пространства, действуя точно таким образом, как мы поступали на заключительном этапе анализа задачи о миссионерах и людоедах (см. книгу 1).

10.5. Стратегии выполнения действий

10.5.1. Исполнительные макрооператоры

Как указывалось в п. 10.1.2, ИС должна осуществлять выполнение планов и корректировку моделей области в соответствии с действительными результатами их выполнения. Очевидно, что стратегией, которая обеспечивала бы на каждом шаге наилучшее соответствие модели области самой области и плана его действительным результатам, была бы стратегия полного перепланирования после каждого шага выполнения. Однако мы отвергаем такую стратегию как крайне неэффективную. Настолько же неэффективной была бы и стратегия полного игнорирования результатов выполнения действия и их несоответствия «задуманному» плану.

Разумный компромисс между этими крайними альтернативами заключается в том, чтобы

1) при получении новой информации, обнаруживающей несоответствие *j*-го остатка плана, распознать ее и исключить из рассмотрения ИС этот остаток;

2) при неудаче *j*-й шапки плана с точки зрения ожидаемых результатов следует распознать неудачу и либо перевыполнить некоторую часть плана, либо осуществить перепланирование.

Такая стратегия обеспечит выполнение простых или составных планов (полных или неполных), т. е. явится основой для создания ИС класса А — Н, в зависимости от эффективности поиска планов с помощью ПС и сложности предъявляемых задач.

Рассмотрим воплощение указанной стратегии в системе, использующей макрооператоры. Мы предполагаем, что, построив план, система обобщает его и передает полученный макрооператор для выполнения.

Таким образом, первоначально мы должны осуществить частичные подстановки в макрооператоре, обеспечивающие доказательство целевой ппф. Важно отметить, что условия, при которых он может быть выполнен, остаются в обобщенном виде.

Алгоритм подготовки макрооператора для выполнения работает следующим образом:

1) В ячейку $c_{o,\ n+1}$ помещаются все предписания из начальной модели, которые были использованы в процессе построения плана при доказательстве целевой ппф.

2) Предписания из *(n*+1*)*-й строки используются для доказательства целевой ппф. Подстановки, произведенные в процессе этого доказательства, распространяются на весь макрооператор.

3) Предписания в (*n*+1)-й строке, представляющие собой поддержку доказательства целевой ппф, отмечаются. Полученный макрооператор готов для управления выполнением действий.

Рассмотрим пример из п.10.2. Обобщенная ТТ для этого примера приведена на рис. 10.9. Осуществляя шаги алгоритма, мы

1) помещаем в $c_{0,3}$ BOX (BOX1);

2) используем BOX (BOXl), INROOM (Отправитель, *p9)* и INROOM (*p6*, *p9)* для доказательства

G_o: ($\exists$ *x*) [BOX *(x)* $\wedge$ INROOM *(x*, R1)];

3) осуществляем подстановки, произведенные во время доказательства, а именно *p6* =BOX1 и *p9*=*R1* во всей ТТ (рис. 10.9);

4) отмечаем предписания BOX (BOX 1) и INROOM (BOX1, Rl).

Подготовленная для выполнения ТТ приведена на рис. 10.14.

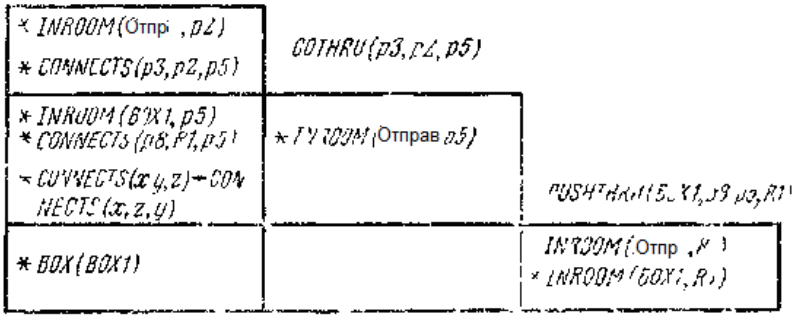


Рис. 10.14. Подготовленный для выполнения макрооператор (пример из п.10.2).

Полученный *макрооператор* мы будем называть *исполнительным.*

Управление выполнением плана с помощью исполнительного макрооператора основано на рассмотренном в п. 10.3.1 достаточном условии применимости *j*-го остатка плана. Таким образом, для продолжения выполнения действий на каждом шаге необходимо иметь хотя бы одно такое ядро, все предписания в котором были бы истинны. Заметим, что в начале выполнения плана ИС исходит из начальной модели, так что первое ядро содержит только истинные предписания. Однако в дальнейшем непредвиденные в плане обстоятельства могут либо неожиданно приблизить нас к цели, либо, наоборот, сделать план невыполнимым.

Для иллюстрации работы с исполнительным макрооператором мы предполагаем, что хотя бы часть плана может быть использована для выполнения (в противном случае должна начаться фаза полного перепланирования). Производится проверка ядер, начиная с конца, т. е. с ядра с наивысшим номером ((*n*+1)-е ядро — это (*n*+1)-я строка ТТ). Если все предписания в *(n*+ 1)-м ядре истинны, план выполнен. В противном случае мы проверяем на истинность предписаний *n*-е, (*n*—1)-е и т. д. ядра. Пусть первое ядро, все предписания которого истинны, имеет номер *j*. Тогда выполнены предусловия *j*-го остатка плана, F_j, F_{j+1} . . ., F_n. Мы применяем F_j и вновь ищем ядро с наивысшим номером, все предписания в котором истинны. В случае полного соответствия ожидаемых результатов планирования действительным, указанная процедура просто выполняет все операторы плана последовательно. Однако она имеет возможность устранить ненужные операторы и ликвидировать неудачи повторным выполнением операторов, например, путем других подстановок параметров в соответствии с изменившейся по каким-либо причинам моделью.

Это последнее обстоятельство представляется особенно важным, поскольку позволяет ИС осуществить фактическое перепланирование без обращений к ПС (при наличии альтернативных возможностей).

Очевидно, что процедура аналогична механизму возврата к точке ветвления, причем точка ветвления находится автоматически в соответствии с текущей моделью области.

Остается заметить, что проверка ядер на истинность всех предписаний, входящих в них, осуществляется с помощью процедуры сканирования, описанной в п. 10.3.3.

10.5.2. Обобщенные процессы планирования и выполнения

Как отмечалось в п. 10.1.2, постановка задачи совместной оптимизации процессов планирования и выполнения действий связана с обобщением этих процессов в единый процесс. Рассмотрим две возможности такого обобщения.

Первая из них связана с рассмотрением ПС как некоторой программы действий, которая оказывает воздействие не на внешнюю область, как это делает ИС, а на абстрактную среду, состояниями которой являются планы. Если оказывается возможным оценить полезность построенного неполного плана, то в каждом состоянии в пространстве планов ИС с помощью некоторого метаоператора может выбрать наиболее выгодное направление планирования. Более того, если оценки для

вновь вырабатываемого плана хуже оценок для уже выработанного плана (полного и неполного), то ИС осуществляет действие, а если лучше, то продолжает планирование. В такой модели описания процессов планирования и выполнения ИС представляет собой по существу метапланирующее устройство, осуществляющее на абстрактном уровне рациональный выбор между планированием и действием и выдающее заказы на планирование ПС. В свою очередь ПС играет роль генератора абстрактных квазидействий, формирующего по запросу из ИС некоторые, возможно, гипотетические планы с оценками их полезности. В этой трактовке ИС изменяет состояния модели и отбрасывает неподходящие для этой модели планы, а ПС, не меняя модели, расширяет существующий для нее план или создает новый план. **Формально действие и планирование могут быть описаны как функциональные отношения в пространстве, состояния которого объединяют состояния модели и плана.**

Другая возможность обобщения процессов планирования и выполнения связана с идеей планирования в иерархии абстрактных пространств. Экстраполируя эту идею от базового пространства, в котором строятся планы, на уровни еще большей детализации, вплоть до непосредственного управления подсистемами ИСФ, мы получаем полностью объединенного систему планирования и выполнения действий. Выполнение процесса в такой системе можно представить себе следующим образом

Базовое пространство ПС рассматривается как абстрактное пространство для ИС. Таким образом, уточняя детали, отсутствующие в этом базовом пространстве, общая система постепенно опускается до базового пространства ИС. Можно предполагать, что изменения в заданной области будут всегда в большей степени воздействовать на решения, принятые на низких уровнях иерархии пространств, чем на решения на более высоких уровнях. Тогда на каждом более высоком уровне план можно считать более близким к эффективному. Конечно, в такой трактовке взаимодействия планирования и выполнения действий требования к качеству решений на высоких уровнях иерархии существенно высоки. Перед началом погружения в пространства низкого уровня план должен быть максимально близок к эффективному. Однако важно отметить, что это может быть достигнуто не только за счет введения эффективных эвристик, но и за счет загрубления плана (путем исключения деталей). Стратегия выполнения основана на существенно неполном планировании. Выбирается небольшое число шагов плана как набросок плана действий, и он постепенно, по мере перехода к более низким уровням иерархии

пространств ИС, наполняется деталями становясь, возможно, при этом менее эффективным. На некотором уровне детали начинают вводиться только в начальную часть выработанного к этому моменту субплана и т. д. Наконец, вырабатывается короткий субплан в базовом пространстве ИС. Этот субплан выполняется. Расхождения между предполагаемыми на уровне субплана и действительными результатами наиболее вероятно являются деталями для построенного плана в базовом пространстве ПС и тем более для набросков плана на высших уровнях планирования. Тогда можно считать, что эти планы пока близки к эффективному.

Далее подобным образом строится новый субплан в базовом пространстве ИС, отражающий точно все появившиеся расхождения. В процессе построения нового субплана возможен возврат на тот уровень пространства, в котором это расхождение еще не является деталью.

В крайнем случае, когда расхождения оказываются достаточно серьезными, сигнал о неудаче распространяется точно до того уровня в иерархии пространств, в котором это расхождение является деталью.

Перепланирование начинается именно с этого уровня, оставляя наброски плана на высших уровнях неизменными.

Представляется, что использование подобных простых и коротких субпланов особенно эффективно в сложных областях, где степень неопределенности достаточно высока.

10.6. Особенности планирования при неполном описании мира (пространства, среды)

Для описанных выше ПС в конечном счете нужна исчерпывающая детализация описания внешнго мира (пространства, среды). Очевидно, что для большинства реальных задач, с которыми может столкнуться разработчик фреймов, детализация описания мира (пространства, среды) не может быть обеспечена. Кроме того, подход, связанный с точным запоминанием всех деталей окружающей обстановки и других, возможно, выведенных фактов, может оказаться неудовлетворительным. Во-первых, детальная информация может очень часто меняться. Во-вторых, в реальных мирах количество информации может оказаться существенно большим, чем в состоянии запомнить система.

Таким образом, наряду с дальнейшим развитием ПС, работа которых основывается на представлении всех фактов об окружающем пространстве, представляется необходимым поиск других подходов, в принципе основанных на неполном знании (исформации). Одним из

таких подходов является использование альтернативных планов, основанных на использовании составных и сложных (полных или неполных) планов. Рассмотрим возможность построения таких планов в рамках ПС класса описанных в п.10.2. Расширим стандартное описание оператора с целью преобразования его в сложный оператор, т. е. оператор с несколькими выходами, каждому из которых приписана некоторая вероятность. В этом расширении каждый выход оператора будет описываться своими списками добавлений и вычеркиваний, а также указанной выше вероятностью

Типичным примером таких операторов могут служшь операторы сбора информации о состоянии мира, в частности, следующий оператор, проверяющий состояние двери (открыта или закрыта дверь):

CHECKDOOR (DX) — проверить дверь *Dx*.

Предусловия:

TYPE *(Dx,* DOOR); NEXTTO (Robot, DA:).

Выход 1	Выход 2
Список вычеркиваний:	Список вычеркиваний:
ПУСТО.	ПУСТО.
Список добавлений:	Список добавлений:
DOORSTATUS (D*x*, OPEN).	DOORSTATUS (D*x*, CLOSED).
Вероятность: 0,5.	Вероятность: 0,5.

Вообще говоря, в операторах сбора информации предусловия должны содержать совет о том, когда следует применять оператор (например, в виде требования, чтобы собираемая информация не была известна перед применением оператора). Такое требование гарантировало бы, что полученная оператором информация будет добавляться к модели, и, таким образом, позволило бы отличать оператор сбора информации от операторов, производящих сходные действия (в нашем примере— оператор CHECKDOOR от операторов типа F, в п. 10. 4. 3). Заметим, что такого рода требования представляют собой новый класс предусловий, так как требуют получения неудачи как в доказательстве ппф (дверь закрыта), так и в доказательстве отрицания ппф (дверь открыта), выражая, таким образом, неопределенность ситуации. Для таких предусловий следует определить свой синтаксис, а также добавить возможности получения подобных доказательств.

При использовании многовыходных операторов пространство поиска может быть представлено в виде пропозиционального графа, где вершины соответствуют состояниям окружающей среды, а каждая ветвь — выходу операторов, применяемых к вершине, причем выходы

из различных операторов образуют дизъюнктивные ветви, а выходы одного оператора — конъюнктивные Тогда план может быть представлен в виде поддерева, состоящего из начальной вершины дерева и, для каждой вершины плана, тех дочерних вершин, которые получены в результате применения выходов одного оператора. Каждая конечная вершина плана соответствует модели среды, удовлетворяющей целевой ппф (если план успешен).

Этот план является условным планом с ветвлением, поскольку в каждой вершине производится проверка, каков действительный выход ИС, и в соответствии с ним выбирается та или иная ветвь плана. При этом вероятность появления каждой вершины равна произведению вероятностей выходов операторов, определяющих путь из начальной вершины дерева к данной вершине.

Возникает вопрос: каковы должны быть условия окончания для алгоритма построения условного ветвящегося плана? Двумя крайними и потому тривиальными вариантами являются достижение абсолютного успеха (все конечные вершины удовлетворяют целевой ппф) и достижение абсолютной неудачи (ни одна из конечных вершин не удовлетворяет целевой ппф). Однако нас интересуют промежуточные случаи. Один из таких случаев будет иметь место, когда дерево полностью построено, и некоторые из его конечных вершин (но не все) удовлетворяют целевой ппф. Следовательно, можно в принципе выбрать такой выход каждого оператора в плане, чтобы план был успешен. Поскольку дерево полностью построено, то произойдет окончание работы алгоритма, и в качестве результата разумно выбрать такой план, для которого сумма вероятностей появления вершин плана, удовлетворяющих целевой ппф, будет максимальной.

Рассмотренные варианты являются полными планами. Для неполного плана необходимо иметь условия окончания, отличные от полного построения графа или достижения абсолютного успеха.

Можно принять за условие окончания нахождение такого плана, вероятность успеха которого превышает некоторый порог. Здесь вероятность успеха неполного плана определяется как сумма вероятностей вершин, построенных к этому моменту. Разумность такого критерия аргументируется исключением лишнего планирования в ситуации, когда план, близкий к успешному, уже найден.

При построении неполного плана, как обычно, возникает задача взаимодействия ПС и ИС. Вновь мы хотели бы иметь стратегию, позволяющую осуществлять продолжение планирования или перепланирование после каждого шага выполнения действий; однако такая стратегия слишком непрактична. Другими возможностями взаимодействия являются:

— вызов ПС, когда конечная вершина плана найдена, а задача не решена;
— вызов ПС, когда найдена вершина, ни одна из дочерних вершин которой не удовлетворяет целевой ппф.

Следует также включить в условия окончания и такие ясные случаи, которые должны останавливать построение плана, вероятность неудачи которого превышает некоторый порог. Это условие может быть выражено двояким образом:

1) когда каждый план в графе имеет вероятность достижения терминальной вершины, не удовлетворяющей целевой ппф, большую, чем некоторый предел;
2) когда каждая нераскрытая вершина в графе имеет вероятность, меньшую, чем некоторый малый предел.

Аргументом в пользу такого условия окончания является предположение, что ни один из планов, вероятно, не приведет к успеху.

При планировании целесообразно использовать стратегию выбора наиболее перспективного плана с последующим выбором в нем вершины для раскрытия. При этом в качестве оценок планов могут выступать вероятности успеха и неудачи для вершин плана и эвристические функции в принятом ранее смысле этого понятия. Оценки вершин могут основываться и на эвристических функциях, а также на вероятностях их появления.

Проиллюстрируем сказанное выше на примере (рис. 10.15).

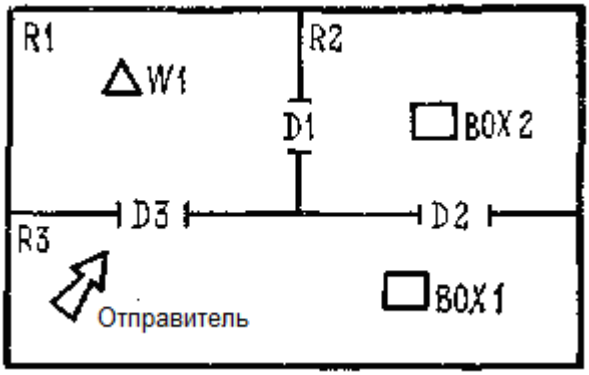


Рис. 10.15. Пример планирования при неполном описании пространства.

Задача отправителя состоит в том, чтобы передвинуть ящик BOX 1 в комнату R1 так, чтобы призма W1 и любой ящик не были в одной

комнате. Предположим, что отправитель не знает состояние дверей Dl, D2 и D3 (открыты они или закрыты) и не знает, есть ли призма W1 в комнате R1. План, использующий операторы GOTHRU и PUSHTHRU из примера в п.10.2 в сочетании с введенным выше оператором CHECKDOOR и новым оператором CHECKOBJECT (Wl, R1) (проверить, есть ли W1 в комнате R1), показан на рис. 10.16.

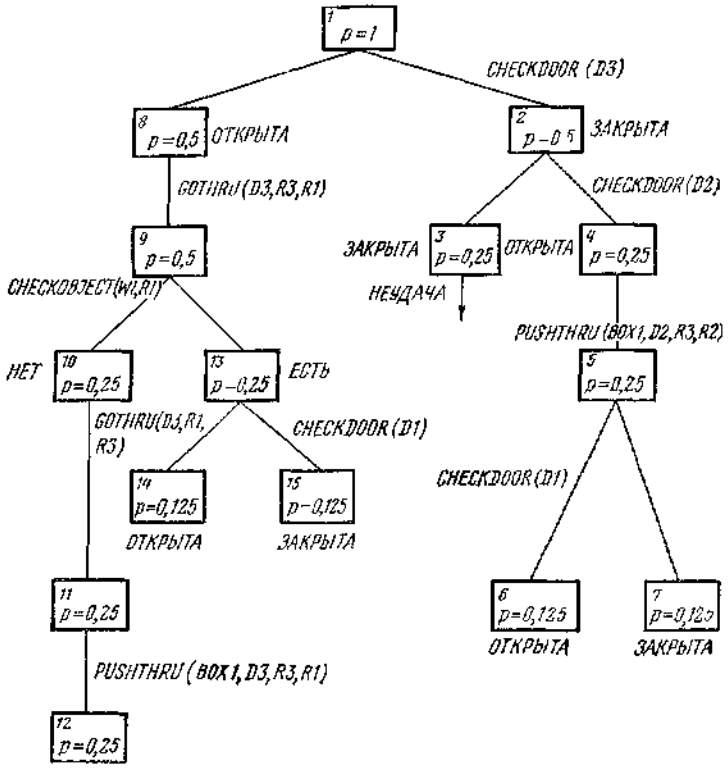


Рис. 10.16. Пример плана со сложными операторами.

Здесь предполагается, что планирование обрывается, когда все нераскрытые вершины имеют вероятность, меньшую 0,2. Номера в квадратах указывают последовательность раскрытия вершин. Мы предоставляем читателю возможность самостоятельно просмотреть этот план. Отметим только, что вероятность успеха для этого плана равна 0,25, вероятность неудачи — также 0,25, оставшиеся

возможные выходы не раскрыты, так как вероятности соответствующих вершин лежат ниже установленного порога.
Дополнительные возможности планирования с многовыходными операторами раскрываются при планировании в иерархии абстрактных пространств. Использование иерархического предписания может упростить построение условных планов, потому что неопределенность выходов операторов сохраняется лишь до определенного уровня детализации. В пространстве высшего уровня предусловия и действия операторов могут выражаться с помощью стандартного описания, т. е. без неопределенности. Однако некоторые из действий на этом уровне могут описываться в понятиях параметров, что усложнит преобразование планов в планы в пространствах низших уровней. Тем не менее простота представления сложных операторов делают эту схему перспективной.

10.7. Планирование в процедуральных предписаниях

Рассмотренный в предыдущих параграфах метод построения планов, основанных на декларативном предписании моделей в виде ппф в исчислении предикатов первого порядка и механизме редукции GPS в качестве стратегии поиска решения, обладает рядом достоинств. К числу их прежде всего относятся:
1) универсальность формализма, позволяющая решать задачи в значительном разнообразии миров;
2) возможность формального обобщения планов и использования макрооператоров как для исполнения построенного плана, так и построения других, более сложных планов;
3) возможность представления задачи в иерархии абстрактных пространств с последовательной детализацией первоначально построенных набросков плана.
В настоящем параграфе мы проиллюстрируем на примере некоторые особенности построения планов в процедуральном предписании, используя ПОЯ QA4.

10.7.1. Построение простых планов

Рассмотрим следующую задачу. Отправитель и ящик BOX 1 находятся в одной комнате. Требуется повернуть выключатель. Эта цель может быть достигнута, если отправитель пододвинет ящик к выключателю, встает на него, а затем повернет выключатель.

Для решения этой задачи нам потребуются следующие операторы (мы описываем операторы в декларативном предписании):
F_6: CLIMBONBOX *(bx)* — отправитель встает на ящик *bx*.
Предусловия:
TYPE (*bx*, BOX), ONFLOOR, NEXTTO (Отправитель, *bx)* (ONFLOOR — на полу).
Список вычеркиваний:
At (Отправитель, $), ONFLOOR.
Список добавлений:
ON (Отправитель, *bx*)
F_7: CLIMBOFFBOX (*bx*)— отправитель слезает с ящика *bx*.
Предусловия:
TYPE (*bx*, BOX), ON (Отправитель, *bx*).
Список вычеркиваний:
ON (Отправитель, *bx).*
Список добавлений:
ONFLOOR.
F_8: TURNONLIGHT(*m*)— отправитель поворачивает выключатель *m*.
Предусловия:
TYPE (*m*, LIGHTSWITCH), ON (Отправитель, *bx*), NEXTTO (*bx*, *m).*
Список вычеркиваний:
STATUS(*m*, OFF) *(m* выключен).
Список добавлений:
STATUS (*m*, *ON) (m* включен),
Операторы GOTOB *(bx)* и PUSHB(*(bx, m)* описываются аналогично операторам F_4 и F_1 соответственно (п. 7.4.3).
ПС STRIPS дает следующее решение указанной выше задачи:
GOTOB (BOXl); CLIMBONBOX (BOX1); CLIMB-
OFFBOX (BOXl);
PUSHB (BOX1, LIGHTSWITCH1); CLIMBONBOX
(BOX1);
TURNONLIGHT(LIGHTSWITCHl).

(10.11)

Заметим, что второй и третий операторы последовательности (10.11) реализуют взаимоисключающие действия. Эти действия были бы исключены при подготовке обобщенного плана в виде макрооперaюра к исполнению (п.10.5).
Рассмотрим теперь описание оператора TURNONLIGHT в ПОЯ QA4:
(LAMBDA (STATUS ← *M* ON)
(PROG (DECLARE *N)*
(EXISTS (TYPE $ *M* LIGHTSWITCH))
(EXISTS (TYPE ← *N* BOX))

(GOAL DO(NEXTTO $*N* $*M))*
(GOAL DO (ON (Отправитель $*N*)))
($ DELETE ('(STATUS $M OFF)))
(ASSERT (STATUS $M ON))
($BUILD('(:$ TURNONLIGHTACTION $M))))).

(10.12)

Общая структура довольно прозрачна, особенно для читателя, знакомого с языком LISP. Оператор выбирает выключатель и ящик и требует, чтобы ящик был поставлен рядом с выключателем, а отправитель встал на этот ящик. Затем осуществляются соответствующие вычеркивания и добавления, после чего сам оператор присоединяется к последовательности действий, выполняемых отправителем.

Оператор может быть применен для целей, сопоставляющихся образцу (STATUS ← *M* ON). Здесь ←*M* обозначает переменную, которая может быть сопоставлена любому предписанию, после чего *M* будет связана с этим предписанием. Если мы в соответствии с (10.11) хотим повернуть выключатель LIGHTSWITCH 1, то цель выражается как (STATUS LIGHTSWITCH I ON). Эта цель сопоставляется с образцом под знаком LAMBDA, и *M* связывается с LIGHTSWITCH 1. Таким образом, образец (STATUS ←*M* ON) играет роль списка добавлений оператора F_8.

Предусловия F_8,, выражены в форме предписаний под знаком EXISTS. Поиск выражений, сопоставляющихся с предписаниями EXISTS, производится в базе данных. Здесь $M означает, что переменная *M* может быть сопоставлена только с уже присвоенными *M* значениями. Таким образом, сопоставление с третьей строкой даст выражение (TYPE LIGHTSWITCH I LIGHTSWITCH). Этот факт, если он истинен, хранится в базе данных под знаком ASSERT. В общем, модель области в процедуральном представлении состоит из всех выражений ASSERT. Заметим, что в случае, если выражение, сопоставляющееся с EXISTS, не будет найдено в базе данных, то это вызовет сигнал о неудаче, включающий механизм возврата.

Второе выражение под знаком EXISTS будет истинным, если есть хоть один ящик. Для случая B0X1 переменная N будет связана с B0X1. Эта строка оператора выбирает ящик для последующих действий.

Следующее предусловие оператора выражено в виде цели (GOAL DO(NEXTTO $*N* $M)). Сначала производится проверка, нет ли выражения формы (NEXTTO $*N* $*M)* в базе данных (действие аналогично EXISTS). В случае положительного ответа цель достигнута и производятся соответствующие связывания *N* и *M* (заметим, что для этого, в силу префикса $, в базе данных должно быть выражение

(NEXTTO B0X1 LIGHTSWITCH1)). Если такого выражения в базе данных нет, то цель рассматривается как подцель. Управление ПС в процедуральном предписании осуществляется специальной управляющей программой, получающей цели и подцели и выбирающей в соответствии с некоторой семантической информацией и рекомендациями пригодные для достижения подцели операторы.

Оператор снабжен рекомендациями относительно того, какие операторы следует использовать для достижения подцели. В данном случае имеется два класса операторов: F_4 принадлежит к классу GO, а F_1, F_6, F_7, F_8 — к классу DO.

Поэтому для установленной подцели (GOAL DO (NEXTTO *$N $M*)) можно использовать один из четырех упомянутых операторов. Эти операторы пригодны, однако применимым из них будет тот, чья форма под знаком λ-выражения будет сопоставляться с выражением, стоящим под знаком GOAL. Приведем теперь запись оператора F_1 в языке QA4 (Мы используем слегка измененную формулировку оператора F_1, исключая из его предусловий TYPE (by, OBJECT) и включая предикат без параметров ONFLOOR.)*:*

```
(LAMBDA (NEXTTO←M ←N)
(PROG (DECLARE X)
(EXISTS (PUSHABLE $M))
(EXISTS (ONFLOOR))
(EXISTS (INROOM $M ←X))
(EXISTS (INROOM $N $X))
(EXISTS (INROOM Отправитель $X))
(GOAL GO (NEXTTO Отправитель $M))
(MAPC (QUOTE (TUPLE (AT (Отправитель ←X))
                    (AT ($M ←X))
                    (NEXTTO (Отправитель ←X))
                    (NEXTTO ($M ←X))
                    (NEXTTO ($X  $M))))
                        $DELETE)
(ASSERT (NEXTTO $M $N))
(ASSERT (NEXTTO $N $M))
ASSERT (NEXTTO Отправитель $M)))
($ BUILD (' (:$ PUSHBACTION (TUPLE $M $N)))))
```

(10.13)

Очевидно, что оператор (10.13) применим для достижения подцели (GOAL DO (NEXTTO *$N $M)),* т. е. переменная *M* в (10.13) связывается с BOX1, а *N* с LIGHTSWITCH1. Оператор (10.13) устанавливает другую подцель класса GO, так что делается попытка применить F_4. Эта попытка будет успешной.

Вернемся к оператору (10.12). Третье его предусловие выражается в виде (GOAL DO (ON (Отправитель $*N*))). Эта подцель может быть достигнута с помощью F_6. Мы не будем в интересах краткости изложения описывать F_4 и F_6 в виде процедур QA4, поскольку общий стиль конструирования этих процедур, вероятно, достаточно ясен.

Мы также не будем описывать синтаксические подробности описания процессов вычеркивания и добавления в операторах (10.12) и (10.13). Они осуществляются выражениями под знаком DELETE и ASSERT соответственно. Последняя строка операторов, как указано в самом начале, присоединяет оператор с соответствующими подстановками параметров в общую последовательность действий, осуществляемых отправителем.

Следует обратить внимание на использование в описаниях операторов дополнительной информации, получаемой от пользователя. Об использовании классификации операторов в рекомендациях мы уже упоминали. Заметим, что выражения типа GOAL в (10.12) установлены в таком порядке, что отправитель не будет пытаться встать на ящик до того, как придвинет его к выключателю. Это позволяет избежать таких взаимоисключающихся последовательностей операторов, как в (10.11). Конечно, и в декларативной формулировке задачи можно было тоже наложить упорядочение на использование предусловий.

В настоящее время работы по созданию законченных ПС в процедуральных предписаниях находятся в начальной стадии, повторяя в значительной степени задачи того же класса, что и STRIPS .

10.7.2. Построение условных и циклических планов

Предположим, что нам задано множество связанных предикатов *P1, P2, . . ., Pn,* значения которых неизвестны во время планирования, однако могут быть установлены во время выполнения плана. Предикаты связаны в том смысле, что по крайней мере один из них должен принимать значение ИСТИНА во время выполнения плана. Тогда общая форма условного плана может быть представлена в виде

IF PI THEN Al ELSE
IF P2 THEN A2 ELSE
. . . . . .
IF *P* (*n*—1) THEN *A* (*n*—1) ELSE *An.*

Заметим, что *Pn* не проверяется на истинность, так как он должен проверяться только в том случае, если *P1, P2, . . ., P(n*—1) принимают значение ЛОЖЬ. Но в этом случае *Pn* должен быть истинным.

Один частный пример условного плана рассмотрен в п. 10.6.2 для случая многовыходных операюров, выходы которых управляются соответствующими вероятностями.

Рассмотрим более общий случай условного плана, где ветвления обуславливаются значениями предикатов *P1, P2, . . ., Pn,* истинность которых устанавливается некоторым источником действий, возможно, независимым от отправителя. Тем самым мы делаем первый шаг на пути к построению планов в динамической среде, одновременно демонстрируя некоторые характерные черты построения относительно сложных планов в процедуральном предписании. Поскольку при построении условных планов значения предикатов не определены, то следует составить такую программу, которая создавала бы последовательность локальных баз данных, или контекстов, в которых все большее количество предикатов получало бы определенные значения, а затем строить планы в этой последовательности контекстов. Эта программа, называемая в дальнейшем ALTPLAN (ALTPLAN (alternative plan) — альтернативный план), воспринимает в качестве аргументов предикат, или условие, и цель и в случае, если это условие истинно, указывает, что план не является условным. В случае, если условие не определено, ALTPLAN создает новый контекст, в котором условие ложно, решает задачу в этом контексте и выдает план решения в качестве результата.

Прежде чем перейти к описанию процесса планирования в ПОЯ QA-4, введем некоторые дополнительные определения, касающиеся этого языка.

Структуры данных. Мы будем использовать два типа структур — тупль (TUPLE) и множество (SET). TUPLE — это выражение формы (TUPLE *e*1. . .*en),* где $e_1, \ldots, e_n$— произвольные выражения, являющееся аналогом списка. Значение тупля есть тупль, элементами которого являются значения элементов исходного тупля. SET — это выражение формы SET (*e*l. . .*en),* причем порядок элементов и число вхождений одного и того же элемента безразлично. Например, SET *(ABC),* SET(*BCA*) и SET*(CAACB)* идентичны, Значением множества является множество, элементами которого являются значения элементов исходного множества.

Встроенные функции. Нам потребуется встроенная функция «равно» (EQUAL). Аргументом EQUAL является множество. EQUAL принимает значение ИСТИНА (TRUE), если значения всех элементов аргумента идентичны. Например, EQUAL (*A* $*X*) принимает значение TRUE, если *X* имеет значение *A*.

Переменные. Переменные — это идентификаторы, снабженные префиксами (нам потребуются префиксы ←, $, ↞ и $$). Значения

первых двух префиксов объяснены в п. 10.7.1. В отличие от переменных с префиксами ← или $, сопоставляющихся одному элементу, переменные с префиксом ←← и $$ являются сегментными, т. е. могут сопоставляться некоторому фрагменту множества или тупля. В остальном префиксы ←← и $$ работают так же, как префикс ← и $ соответственно.

Функция квази-QUOTE обозначается ' и имеет формат ('*e*). Эта функция подставляет значения переменных в *e*, не вычисляя последнего. Например, если *X*⇒2 (*X* имеет значение 2), то

('(PLUS $X 5)) ⇒ (PLUS 2 5), но не 7.

Операция DENY (отрицать) имеет формат DENY (синтаксическая форма) и присваивает значение FALSE (ЛОЖЬ) выражению с данной синтаксической компонентой, т. е. делает это выражение ложным в модели пространства.

Оператор SETQ имеет формат SETQ *(pe),* где *p* — образец, и сопоставляет образец с результатом вычисления выражения *e,* связывая переменные в образце с подвыражениями, которым они были сопоставлены.

Условные выражения. Мы будем использовать условные выражения IF (если) и ATTEMPT(пробовать).

Выражение IF имеет общую форму (IF *e*1. . .*en* THEN *e*'l . . .*e'm* ELSE *e*''l. . .*e''k).* Выражения $e_1, \ldots, e_n$ вычисляются по порядку. Если значение последнего выражения e_n FALSE, то вычисляются по порядку $e''_1, \ldots, e''_k$ и значение e''_k выдается как результат условного выражения IF. Если значение e_n отлично от FALSE, то вычисляются по порядку $e'_1, \ldots, e'_m$ и значением IF является значение e'_m. Например,

значением выражения

IF (EQUAL $X 4) THEN (SETQ ← *Y3)* (PLUS $X *$Y)*

ELSE (SETQ←*Y* 6) (PLUS 4$ *Y*)

будет 7, если X=4, и 10 в противном случае. Выражение ATTEMPT имеет общую форму

(ATTEMPT *e1* . ..*en* THEN *e'1*. . . *e'm* ELSE *e''1* . . . *e''k).*

Выражения $e_1, \ldots, e_n$ вычисляются по очереди, и если одно из них генерирует сигнал о неудаче, то управление передается в ELSE (если ELSE отсутствует, выдается сигнал FALSE). Если ни одно из $e_1, \ldots, e_n$ не выдает сигнала о неудаче, управление передается в THEN. Дальнейшие вычисления подобны вычислениям выражения IF.

Выражение RETURN (возврат) используется для выхода из программы PROG, имеет форму (RETURN *e)* и выдает в качестве результата программы выражение *e* с подставленными значениями переменной. Например, если BOX имеет значение BOX1, то результатом

программы по выходу RETURN (NEXTTO Отправитель $BOX) будет (NEXTTO Отправитель BOX1).

Выражение (CONTEXT PUSH CURRENT) [PUSH CURRENT — протолкнуть текущий] создает новый контекст, дочерний по отношению к текущему в дереве контекстов.

Связывание выражений с контекстом $CONTEXT осуществляется приписыванием к выражению конструкции WRT $CONTEXT (WRT (with respect to) — относительно).

Используя приведенные выше определения, построим программу ALTPLAN:

```
ALTPLAN(LAMBDA(TUPLE←CONDITION←GOAL)
          (PROG (DECLARE NC Y ALT Z)
          (ATTEMPT (EXISTS $CONDITION)
         THEN (RETURN (TUPLE)))
         (EXISTS (UNCERTAIN (SET   $CONDITION←← Y)))
         (SETQ←NC (CONTEXT PUSH CURRENT))
          (DENY (UNCERTAIN (SET $CONDITION
                        $$Y)) WRT $NC)
         (DENY $CONDITION WRT $NC)
         (ATTEMPT (SETQ (SET←Z)$$Y)
         THEN (ASSERT $Z WRT $NC)
         ELSE (ASSERT (UNCERTAIN $Y) WRT $NC)
        (SETQ ←ALT (GOAL $GOALCLASS   $GOAL WRT $NC)
        (RETURN $ALT))).
```

(10.14)

Первое выражение ATTEMPT проверяет, существует ли входное условие CONDITION в модели, и выдает в качестве результата пустой тупль (TUPLE), если это условие существует. Это означает, что альтернативный план не нужен. В противном случае EXISTS извлекает из базы данных множество неопределенных (UNCERTAIN) условий, связывая Y со всеми такими условиями, кроме входного. Затем программа создает новый контекст NC, отрицая в нем множество неопределенных условий как утверждения в текущем контексте. Кроме того, мы предполагаем, что в новом контексте NC условие CONDITION ложно. Второе выражение ATTEMPT проверяет, осталось ли в множестве неопределенных условий только одно такое условие. В случае, если условие одно, оно утверждается относительно нового контекста. В противном случае утверждается неопределенность меньшего множества условий. Затем программа решает задачу в новом контексте и выдает план.

Рассмотрим работу приведенной программы на следующем примере. Пусть отправитель хочет развлечься (HAVEFUN) и имеет для этого две

возможности: если идет дождь (RAINY), то отправитель идет в кино (MOVIE), если же светит солнце (SUNNY), он идет на пляж (BEACH). В результате мы хотим получить план вида

IF SUNNY THEN BEACH ELSE MOVIE. (10.15)

Запишем программы действий отправителя — BEACH и MOVIE.

```
BEACH (LAMBDA HAVEFUN
(PROG (DECLARE ALT) (SETQ ←ALT ($ALTPLAN SUNNY   HAVE-
FUN))
 (IF (EQUAL (TUPLE) $ALT)
     THEN(RETURN BEACH)
     ELSE
     (RETURN ('(IF SUNNY THEN BEACH
     ELSE  $ALT))))));
MOVIE (LAMBDA HAVEFUN
          (PROG (DECLARE ALT)
          (SETQI←ALT ($ALTPLAN RAINY    HAVEFUN))
         (IF (EQUAL (TUPLE) $ALT)
          THEN (RETURN MOVIE)
           ELSE
          (RETURN ('(IF RAINY  THEN MOVIE
           ELSE $ALT))))).
```

Эти программы аналогичны с точностью до условий, поэтому мы опишем работу одной из них, например BEACH. Программа вызывает ALTPLAN, запоминая его в ALT. Если ALTPLAN выдаст пустой тупль (условие SUNNY истинно), то программа BEACH выдаст план BEACH. В противном случае ALTPLAN решает задачу в предположении, что SUNNY ложно, и программа BEACH выдает план IF SUNNY THEN BEACH ELSE $ALT.

Приведем протокол выработки условного плана (10.15) с помощью программ ALTPLAN, BEACH и MOVIE. Итак, отправитель хочет развлечься, так что общая цель может быть представлена в виде

(GOAL $HAVEFUN HAVEFUN), (10.16)

где переменная HAVEFUN указывает, что цель относится к классу целей (GOALCLASS) HAVEFUN. В нашем случае к этому классу относятся две программы действий — BEACH и MOVIE.

Общий план построения условного плана сводится к следующему. Общая цель вызывает, например, BEACH, которая вызывает ALTPLAN. ALTPLAN упрощает неопределенное условие и вновь вызывает BEACH. BEACH опять вызывает ALTPLAN. На этот раз ALTPLAN терпит неудачу, так что вызывается MOVIE. Поскольку условие RAINY истинно, план MOVIE передается через ALTPLAN в

BEACH, который вкладывает ее в общий условный план. Ниже приводится подробный протокол.

1. Ввод в базу данных (ASSERT (UNCERTAIN (SET RAINY SUNNY))).
2. Ввод в базу данных (GOAL $HAVEFUN HAVEFUN).

Примечание. Этот ввод осуществляется извне пользователем.

3. $HAVEFUN ⇒ (TUPLE BEACH MOVIE) — процедуры класса HAVEFUN.
4. HAVEFUN отсутствует в базе данных, неудача.
5. Вызов LAMBDA BEACH.
6. Вызов LAMBDA ALTPLAN.
7. Связывание переменной CONDITION с SUNNY, GOAL с HAVEFUN.
8. Проверка (EXISTS SUNNY), неудача.
9. Проверка (EXISTS(UNCERTAIN (SET SUNNY ↞ Y))), связывание Y с RAINY.
10. Создание нового контекста NC.
11. Отрицание (UNCERTAIN (SET SUNNY RAINY)) относительно нового контекста NC.
12. Отрицание SUNNY относительно нового контекста NC.
13. Определение того, что осталось только одно условие RAINY, связывание Z с RAINY.
14. Утверждение RAINY в новом контексте.
15. (GOAL (TUPLE BEACH MOVIE) HAVEFUN) устанавливается относительно нового контекста.
16. HAVEFUN отсутствует в базе данных, неудача части EXISTS механизма GOAL.
17. Вызов LAMBDA BEACH.
18. Вызов LAMBDA ALTPLAN.
19. Связывание переменной CONDITION с SUNNY, GOAL с HAVEFUN.
20. Проверка (EXISTS SUNNY). В этот момент SUNNY ложно (см. п. 12), неудача.
21. Проверка (EXISTS (UNCERTAIN (SET SUNNY ↞ Y))), это множество отсутствует в NC (см. п. 11), неудача, возврат к точке ветвления (п. 17)
22. Вызов LAMBDA MOVIE.
23. Вызов LAMBDA ALTPLAN.
24. Связывание переменной CONDITION с RAINY, GOAL с HAVEFUN.
25. Проверка (EXISTS RAINY), RAINY истинно в NC (п. 14).

26. ALTPLAN выдает (TUPLE), переменная ALT в MOVIE связывается с (TUPLE).
27. MOVIE выдает константу MOVIE, цель (п. 15) удовлетворена, переменная ALT в программе ALTPLAN принимает значение MOVIE, возврат к программе BEACH (п. 5).
28. Переменная ALT в BEACH принимает значение MOVIE.
29. BEACH выдает план (' (IF SUNNY THEN BEACH ELSE MOVIE)).

Перейдем теперь к рассмотрению процесса построения циклических планов. К сожалению, задача автоматического синтеза циклических планов, включающая в себя определение того, когда и какие части планов следует представлять в виде циклов, еще далека от своего решения. Мы покажем лишь, каким образом построить циклический план, если его цикличность задана явным образом.

Другими словами, если нам задан предикат *P* и действие *A,* содержащие одну и ту же свободную переменную *X,* то задача состоит в том, чтобы построить план, содержащий действие *A* для всех объектов в множестве, определяемом предикатом *P*.

Интуитивно ясно, что задача построения циклического плана может быть решена в три этапа.

Во-первых, необходимо иметь метод перечисления объектов, удовлетворяющих *P*. Во-вторых, необходимо выработать план, который в предположении об истинности *P* будет выполнять действие *A*. Наконец, следует ввести свободную переменную в *P* и *A,* которая к этому моменту является константой в QA-4, соответствующей связанной переменной как в *P,* так и в *A*, после чего объединить новые формы для *P* и *A* в общий циклический план.

Указанные этапы построения циклического плана реализуются в приводимой ниже программе LOOPPLAN (циклический план):

```
LOOPPLAN (LAMBDA (FA ←X (IF ←P THEN ←A))
                (PROG (DECLARE NC BODY NV1 NV2)
               (EXISTS (GENERABLE $P))
               (SETQ←NC (CONTEXT   PUSH   CURRENT))
               (ASSERT $P WRT $NC)
              (SETQ ←BODY (GOAL    $GOALCLASS
               $A WRT $NQ)
               (SETQ (TUPLE ←NV1 ←NV2) ($GENVAR (TUPLE)))
                (SETQ_←P(SUBST     $P   (TUPLE   $X $NV1)))
               (SETQ←BODY(SUBST $BODY (TUPLE $X $NV2)))
             (RETURN (REPEAT (GOAL:     $FIND $P) (DO $BODY))))).
                                                          (10.17)
```

Эта программа требует ряда пояснений. Начнем с пояснений синтаксического характера Форма, стоящая под знаком LAMBDA, содержит идентификатор FA(FA (for all) — для всех), читающийся как «для всех». SUBST — обычная функция языка LISP.

Выражение REPEAT (REPEAT —повторить.) имеет форму (REPEAT nde DO *e*l. . .*en),* где *nde* — недетерминистическое выражение, $e_1, \ldots, e_n$ — выражения, и в нашем случае означает: «повторять выполнение программы $BODY для всех целей класса $FIND, сопоставляющихся образцу $P».

Программа действует следующим образом:

1) Устанавливает перечислимость объектов, удовлетворяющих *P* (GENERABLE $*P).*

2) Создает новый контекст, утверждает в нем *P* и устанавливает цель *A* в этом контексте, вырабатывая план, выполняющий действия *A* для объектов, определяемых *P*.

3) После выполнения плана подставляет произвольную переменную вместо свободной константы в *P* и *A,* т. е. создает план-схему.

4) Выдает в качестве результата итеративный план выполнения действия *A* для всех целей класса $FIND, т. е. осуществляющих поиск объектов, сопоставляющихся со значением *P*.

Рассмотрим пример. Предположим, что отправитель должен перенести все ящики, находящиеся в комнате ROOM1, в комнату ROOM2, т. е. установим цель класса $DO:

(GOAL $DO(FA BOX (IF (INROOM BOX RCOM1)
THEN(INROOM BOX RQOM2)))), (10.18)

и мы хотим построить план достижения этой цели. Предположим, что к классу $DO относятся программа LOOPPLAN и программа, передвигающая ящики,— MOVEBOX. Произвольно предположим также, что поиск ящиков (класс $FIND) осуществляется программами LOCATE (обнаружить) и CHECKMAP (проверить по карте). Построение циклического плана может быть представлено следующим протоколом:

1 Ввод в базу данных (GOAL $DO (FA BOX (IF (INROOM (TUPLE BOX ROOM1)) THEN (INROOM (TUPLE BOX ROOM2))))).

2. $DO $\Rightarrow$ (TUPLE LOOPPLAN MOVEBOX).

3. Цель не найдена в базе данных, неудача.

4. Вызов LAMBDA LOOPPLAN.

5. Связывание переменных: *X* с BOX (свободная константа), *P* с (INROOM(TUPLE BOX ROOM1)), *A* с (INROOM (TUPLE BOX ROOM2)).

6. Проверка (EXISTS (GENERABLE (INROOM (TUPLE BOX ROOM1)))), устанавливается генерируемость.

7. Создание нового контекста NC.
8. Утверждение (INROOM (TUPLE BOX ROOM1)) в контексте NC.
9. Устанавливается (GOAL (TUPLE LOOPPLAN MOVEBOX) (INROOM (TUPLE BOX ROOM2))) относительно контекста NC.
10. Вызов LAMBDA MOVEBOX (программа MOVEBOX не приводится, предполагается, что ее образец (INROOM (TUPLE ←BOX ←ROOM)) сопоставляется с целью, переменная BOX связывается с BOX, переменная ROOM связывается с ROOM2, и программа выдает план (PUSHTO BOX ROOM2)).
11. Переменная BODY принимает значение (PUSHTO BOX ROOM2).
12. Переменная *P* принимает значение (INROOM (TUPLE ←*X* ROOM1)).
13. Переменная BODY принимает значение (PUSHTO *$X* ROOM2).
14. LOOPPLAN выдает результат (REPEAT (GOAL (TUPLE LOCATE CHECKMAP) (INROOM (TUPLE ←*X* ROOM1))) DO (PUSHTO $X ROOM2)).

10.8. Многоцелевое планирование

10.8.1. Общая постановка

В большинстве реальных применений ИСФ необходимо решать одновременно множество задач, а не одну изолированную задачу. Очевидно, что такое планирование становится многоцелевым. В этой постановке следует формулировать цель для ПС в виде множества целевых ппф (выражений GOAL в процедуральном предписании), каждой из которых должен быть присвоен свой вес, или мера полезности. При этом мы должны допустить существование нежелательных целей, описывающих недопустимые состояния пространства, и присваивать им отрицательный вес. Кроме того, следует задать функционал, определяющий общую целенаправленность действий, и обеспечить возможность построения планов, которые минимизируют этот функционал. Таким образом, на каждом шаге планирования необходимо оценивать текущее состояние работы по достижению каждой из целей и принимать решения, над какой из целей работать, на основании вычисления функционала.

Поставленная таким образом задача может потребовать весьма сложных вычислений в каждом состоянии пространства. Кроме того, в общем случае, полезности различных целей должны быть функциями времени и общего состояния процесса планирования. Мы рассмотрим в настоящем параграфе две частные задачи многоцелевого

планирования, не требующие сложных манипуляций, — планирование с ограничениями и кооперация отправителей.

10.8.2. Планирование с ограничениями

Рассмотрим задачу планирования с двумя целями — основной и нежелательной. Мы должны построить план достижения основной цели, избегая при этом любого состояния пространства, удовлетворяющего нежелательной цели. К решению этой задачи можно подойти двояким образом: включить ограничения, представленные ппф нежелательной цели, в предусловия оператора или допустить выполнение операторов без ограничений, но распознавать недопустимые состояния в процессе планирования и направлять поиск по другому пути. Включение нежелательных целей в предусловия нарушает общность и гибкость использования операторов, не говоря уже о дополнительных сложностях, вводимых в механизм работы с предусловиями.

Поэтому мы пойдем по второму пути, принимая следующий план рассуждений. При выборе пригодного для уменьшения различия оператора производится проверка, не явится ли он сам причиной появления недопустимого состояния. Если предусловия оператора имплицируют ппф нежелательной цели, то нет смысла использовать этот оператор, поскольку, устанавливая его предусловия в качестве подцели, мы заранее обрекаем эту часть плана на неудачу. Если список добавлений сам содержит ппф нежелательной цели, оператор также должен быть отвергнут, так как он приведет пространство в недопустимое состояние.

Теперь остается случай, когда ппф нежелательной цели будет удовлетворяться конъюнкцией выражений, уже имеющихся в модели пространства и не вычеркиваемых оператором, и выражений из списка добавлений оператора. В этом случае недопустимое состояние распознается путем доказательства его из текущей модели пространства (после применения оператора). Далее должны быть выделены директивы, использованные в этом доказательстве. Отрицание одного из тех выражений, которые явились поддержкой ппф нежелательной цели, но не были добавлены оператором, вводится временно как дополнительное предусловие оператора. После этого подцель доказательства предусловий этого оператора пересматривается в свете добавлений. Может потребоваться пересмотр целой цепи предшествующих вершин плана (см. рис. 10.1), содержащих предусловия оператора в списках целей. Кроме того, если поддержка ппф нежелательной цели содержит несколько кандидатов,

удовлетворяющих указанным выше условиям, создаются отдельные вершины для каждого из отрицаемых директив, которые добавляются в предусловия оператора.

Проиллюстрируем сказанное выше на примере (рис. 10.15). Пусть основная цель

$$\text{INROOM(R1, BOX1)} \vee \text{INROOM(R1, BOX2)}, \quad (10.19)$$

а нежелательная цель

$$(\forall x \forall y)\ [\sim (\text{INROOM}\ (rx, by) \vee \text{INROOM}\ (rx, \text{W1})], \quad (10.20)$$

т. е. ни в одной комнате не должны стоять одновременно ящик и призма W1.

Для построения плана мы используем операторы GOTHRU и PUSHTHRU, определенные в п.10.2. ПС типа STRIPS решает применить оператор PUSHTHRU (BOXI, D3, R3, Rl), однако это приводит к недопустимому состоянию. Поддержкой доказательства (10.20) являются директивы

$$\text{INROOM(Rl, W1)}, \quad (10.21)$$

$$\text{INROOM (Rl, BOX1)}. \quad (10.22)$$

Поскольку (10.22) получено в результате применения PUSHTHRU, то мы добавляем в предусловия конкретного случая применения оператора PUSHTHRU отрицание (10.21). Тем самым создается подцель извлечения призмы W1 из R1, например, в R2. Эта подцель может быть достигнута применением PUSHTHRU (Wl, Dl, Rl, R2), что приводит опять к недопустимому состоянию. Поэтому создается подцель извлечения BOX2 из R2 в R3. В результате будет построен план

GOTHRU (D2, R3, R2); PUSHTHRU (BOX2, D2, R2, R3);
GOTHRU(D3, R3, Rl); PUSHTHRU(W1, Dl, Rl, R2);
GOTHRU(D2,[1] R2, R3); PUSHTHRU (BOX 1, D3, R3, Rl).

(10.23)

Заметим, что в случае использования макрооператоров необходима проверка всех промежуточных состояний пространства, вырабатываемых компонентами макрооператора. В случае обнаружения недопустимого состояния следует либо запланировать его исключение перед использованием макрооператора либо пытаться обойти его имея по-прежнему в качестве цели результат макрооператора.

Некоторые проблемы возникают и при использовании обобщенного макрооператора для выполнения плана с ограничениями. Во время выполнения обобщенного плана мы должны быть уверены, что процесс подстановки не приведет к недопустимому состоянию. Более того, если даже частный случай плана выполняется в соответствии с планом, то всегда имеется опасность, что в процессе выполнения может появиться новая информация, которая не мешает достижению

основной цели, но вызывает недопустимые состояния пространства. Так, очевидно, что если отправитель (рис. 10.15) не знает, что INROOM (Rl, Wl), и обнаруживает наличие призмы, перемещая BOX1 в R1, необходимо перепланирование. Следовательно, ИС должна распознавать недопустимые состояния и использовать эту информацию для обращения к ПС.

Особый случай возникает, когда ограничения являются постоянным свойством пространства, и, следовательно, недопустимые состояния могут возникнуть при решении любой задачи в этом пространстве. Тогда может оказаться более эффективным построить список частных случаев операторов, применение которых при определенных условиях запрещено. ПС и ИС просто проверяют каждое применение оператора на допустимость, причем ИС исключала бы недопустимые операторы, обращаясь к ПС для перепланирования.

10.8.3. Кооперация отправителей

Назовем *кооперацией отправителей* совокупность нескольких операторов, объединенных общей планирующей системой, избегая при этом термина «многоцелевой отправитель» (этот термин может относиться и к отправителям, решающим одновременно несколько задач одним отправителем). Следует отличать понятия кооперации отправителей от нескольких независимых отправителей, каждый из которых снабжен своим решателем задач и может решить свою задачу вне кооперации с другими отправителями.

В противоположность этому, наши отправители, как мы предполагаем, решают кооперативно одну задачу. Представляется целесообразным выяснить возможность независимого выполнения действий по решению этой задачи каждым отправителем, т. е. каким образом можно распараллелить общий план на субпланы, выполняемые отдельными отправителями.

Если планы представляются ТТ (п. 10.3.1), то задача сводится к разбиению исходной ТТ на подтаблицы, относящиеся к различным отправителям, так, чтобы ИС могла независимо и параллельно управлять их действиями.

Рассмотрим ТТ, представленную на рис. 10.17.

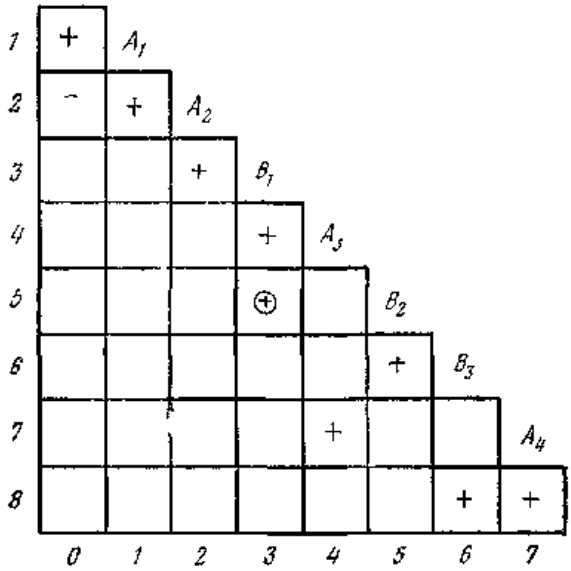


Рис. 10.17. Исходная треугольная таблица.

Здесь *A* и *B* — различные отправители, так что A_1, A_2, A_3, A_4— некоторые операторы, относящиеся к отправителю *A*, а B_1, B_2, B_3 — операторы, относящиеся к отправителю *B*. Знаком «+» обозначены отмеченные директивы, ⊕ в $C_{3,5}$ просто отличает две отмеченных директивы в 4-м столбце. Для каждого из отправителей все отмеченные директивы могут быть разбиты на три класса:

1) Результаты действия B_i, которые являются предусловиями A_j или частью списка добавлений *TT* (класс 1).

2) Результаты действия B_j, которые являются предусловиями B_j (класс 2).

3) Предусловия B_i, которые получаются как результат действия A_j или из начальной модели ТТ (класс 3).

Директивы класса 1 являются внешними выходами для подтаблицы отправителя *B* и поэтому они должны быть помещены в ячейки $c_{i,\ n+1}$ подтаблицы, *n* — количество операторов субплана отправителя *B*. Директивы класса 2 являются внутренними для подтаблицы отправителя *B*, и они должны быть помещены в ячейки $c_{i,j}$ подтаблицы. Наконец, директивы класса 3 являются внешними входами для подтаблицы отправителя *B*, и поэтому они помещаются в ячейки $c_{0,i}$,

подтаблицы. Аналогичным образом может быть построена и подтаблица для отправителя *А*.

Обращаясь к нашему примеру (рис. 10.17) и действуя, как описано выше, мы получим две подтаблицы для отправителя *А* (рис. 10.18, *а)* и отправителя *В* (рис. 10.18, *б)*.

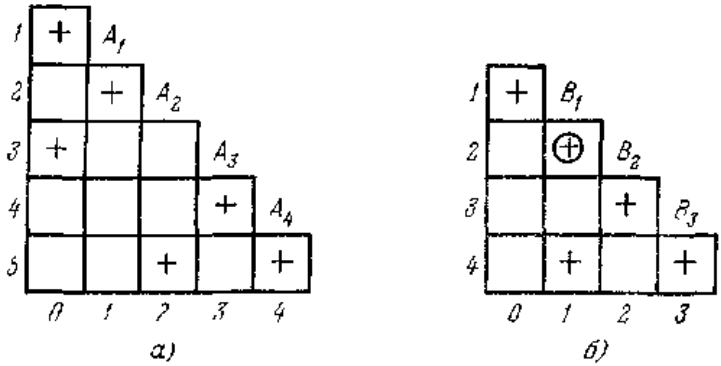


Рис. 10.18. Подтаблицы для отправителей *А* и *В*.

Рассмотренный алгоритм может быть легко обобщен на любое число отправителей последовательным разбиением ТТ на подтаблицы A_1 и $A_2, A_3,..., A_n$, затем подтаблицы $A_2, A_3, \ldots, A_n$ на A_2 и $A_3, A_4, \ldots, A_n$ и т. д.

10.9. Особенности представления планов в динамическом пространстве

В п. 10.1.1 мы указали основные факторы, определяющие динамичность пространства: наличие независимых от отправителя источников действия и явная зависимость элементов модели пространства от времени. В связи с этим возникает вопрос: каковы адекватные средства представления такого пространства в ПС для отправителей? Наиболее естественным решением этого вопроса является описание пространства в виде совокупности параллельно идущих, независимых, равноправных и взаимодействующих процессов, каждый из которых может некоторым образом влиять на пространство, изменяя его. Тогда ПС ИСФ, работающей в динамическом пространстве, должна быть снабжена специальным процессором, в функции которого входит управление процессами, в частности их инициацией,

прерыванием и завершением, и обновление модели пространства в соответствующие моменты времени.

В общих чертах работа такого процессора может быть описана следующим образом. Взаимодействуя с моделью пространства, процессор определяет условия, необходимые и достаточные для инициации тех или иных процессов. Если процесс должен быть инициирован, то процессор запускает его, определяет необходимые связки для переменных процесса, после чего следит за условиями его прерывания или естественного завершения, которые определяются другими процессами, находящимися под управлением процессора. В случае прерывания или естественного завершения процесса процессор производит изменения в модели пространства, соответствующие достигнутым процессом результатам его действия. Взаимодействие процессора с процессами и моделью пространства осуществляется в специальном модельном времени аналогично тому, как это делается в языках моделирования.

Результатом работы процессора является план взаимодействия процессов, необходимый для достижения заданной цели.

Описанная система может быть реализована как в декларативном, так и в процедуральном предписаниях. В качестве процессора может выступать специальная мониторная программа. В настоящем параграфе мы рассмотрим особенности построения планирующей системы, работающей в динамическом пространстве и основанной на базовых представлениях системы STRIPS. Затем мы опишем процедуральный подход к построению децентрализованной системы взаимодействующих процессоров — акторов, основанный на двух основных принципах — неразделимости потока данных и управляющей информации и полной локализации всей обработки информации, включая управление взаимодействием акторов, в самих акторах.

Рассмотрим систему, состоящую из трех основных частей: **модели пространства, множества описаний, или <u>сценариев процессов</u>, и упомянутой выше мониторной программы.** В соответствии с нашими представлениями о ПС, работающих в динамическом пространстве, мониторная программа включает в себя динамически изменяемое множество блоков управления процессами (БУП). Каждый БУП описывается обращением к определенному **сценарию процесса и множеством связок параметров процесса**. Заметим, что если одновременно протекает несколько идентичных процессов, то для каждого из них создается свой БУП, однако с обращением к одному и тому же сценарию. Таким образом, мониторная программа децентрализует управление идущими процессами и, в то время как

БУП производят соответствующие сценариям изменения в модели пространства, другая часть мониторной программы определяет возможность запустить процессы, а также ожидает от БУП сигналов о прерывании или естественном завершении процесса и в этом случае ликвидирует соответствующий БУП. Эту часть монигорной программы назовем *супервизором.* Опишем структуру и дадим примеры сценариев процессов, а затем более подробно в понятиях сценариев процессов опишем ряд функций мониторной программы. При этом будем рассматривать два основных класса процессов.

Первый класс процессов — это процессы, происходящие мгновенно во времени (с точки зрения ПС). Такие процессы можно описывать с помощью стандартной формы операторов системы STRIPS, т. е. **наименования, предусловий и результатов в виде списков добавлений и списков вычеркиваний**. Имеется лишь два отличия. Во-первых, поскольку мы представляем динамическое пространство как совокупность равноправных процессов, то в сценариях, управляющих отправителем, необходимо включать в предусловия указание о процессе выбора отправителем данного сценария из множества альтернатив. Будем задавать это указание в виде (SELECT *rf)*, где *r* — отправитель (в общем случае — один из представителем, так чго *r* является параметром), *f* — сценарий процесса.

Указание предусловия SELECT делает всю совокупность предусловий необходимыми и достаточными условиями для начала процесса, в отличие от предусловий операторов STRIPS, где предусловие определяют только необходимые условия (отправитель может выбрать данный оператор, но не обязан этого делать). Таким образом, как только предусловия сценария процесса удовлетворены, процесс запускается.

Во-вторых, **допустимы два вида предусловий — предусловия, выраженные в исчислении предикатов, или символические предусловия,** которые мы использовали при рассмотрении системы STRIPS, **и предусловия, выраженные в виде отношений, заданных в области действительных переменных, или аналитические предусловия**. Аналитические предусловия необходимы для обеспечения взаимодействия в общем плане мгновенных процессов и процессов второго класса.

Второй класс процессов — это процессы, явно зависящие от времени (в том числе, непрерывные), т. е. процессы, которые воздействуют на модель пространства постепенно. В сценариях таких процессов могут присутствовать результаты действий, выраженные как отношения и функции, заданные в области действительных переменных и принимающие значение также в области

действительных переменных (в отличие от списков вычеркивания и добавлений в обычных результатах действий). Для таких процессов мы будем различать **<u>символические результаты</u>, т. е. выражаемые в исчислении предикатов, и <u>аналитические результаты</u>, выраженные в виде указанных функций и отношений**. Кроме того, для процессов, моделирующих постепенные изменения пространства, мы должны задавать условия их продолжения, нарушение которых приводит к их естественному завершению или прерыванию процесса мониторной программой. Эти условия также могут быть символическими и аналитическими.

Рассмотрим следующий пример: «отправитель должен повернуть кран и наполнить ведро водой». Введем сценарии процессов поворота крана (TURNVALVE) и наполнения ведра (FILLBUCKET).

Наименование сценария: TURNVALVE (r, v, V^c) — отправитель r поворачивает кран v так, чтобы установилась скорость потока V^c.

Предусловия:

символические: (SELECT r TURNVALVE (r, v, V^c));

(V_{max} v V^c_{max}); (AT r n); (AT v n);

аналитические: $0 \le V^c \le V^c_{max}$.

Результаты-М:

вычеркивания: (V v \$);

добавления: (V v V^c).

Процесс, определяемый сценарием TURNVALVE, является мгновенным. В связи с этим его результаты, обозначенные буквой «М», представляют собой обычный список вычеркиваний и добавлений. Значение \$, как и во всех определениях операторов, безразлично. Все параметры, связанные со скоростью потока (V^c, V^c_{max}), принимают численные значения; индекс наверху означает, что переменные связываются при запуске процесса и остаются в течение процесса неизменными.

Наименование сценария: FILLBUCKET (b, v) — ведро b наполняется из крана v.

Предусловия:

символические: (V v V^c); (AT v n); (AT b n);

(ORIENTATION b UP); (C_b b C%)\

(C b C_0);

аналитические: $0 < V^c$;

$C_0 < C_b$

Результаты-П:

символические: (C b C^Y);

аналитические: $C^Y = C_0 + V^c \cdot \tau$.

Условия продолжения:

символические: *(V v V^c); (AT b n);*

(ORIENTATION *b* UP);

аналитические: $C^Y < C_b$.

Этот сценарий определяет постепенно изменяющий свои результаты процесс. В связи с этим его результаты снабжены буквой «П» и имеют символическую и аналитическую часть. Через С обозначено содержимое ведра, C_o— начальное содержимое, C_b— емкость ведра. Индекс «Y» у переменной *C* означает, что эта переменная изменяется самим процессом (в отличие от переменных с индексом С). Таким образом, символическая компонента результатов означает, что в результате процесса содержимое ведра станет C^Y, а аналитическая компонента — что C^Y определяется через начальное содержимое, скорость потока и длительность процесса τ. Предикат (ORIENTATION *b* UP) означает, что ведро должно быть поставлено отверстием вверх.

Условия продолжения указывают, что процесс может быть прерван, если изменится скорость потока, ведро не будет находиться под краном или будет перевернуто. Процесс придет к естественному завершению, если ведро наполнится $(C^Y \geq C_b)$.

Заметим, что, поскольку процесс наполнения независим от отправителя, в его предусловиях отсутствует SELECT.

Рассмотрим теперь работу супервизора мониторной программы на примере наполнения ведра.

Каждый раз, когда супервизор обнаруживает, что предусловия процесса удовлетворены, он создает БУП и записывает в него все значения параметров, при которых удовлетворяются предусловия процесса, и время начала процесса τ_0. Далее, рассматривая символические компоненты предусловий и результатов-П, супервизор определяет отношения, которые будут изменяться процессом. В нашем случае таким отношением является (С ВКТ C_0), где ВКТ — конкретное ведро (значение параметра *b).* Все такие отношения удаляются из множества символических отношений в модели пространства и добавляются к множеству аналитических отношений модели пространства с заменой значения на переменную с индексом *Y.* В нашем примере будет удалено (С ВКТ C_o) и добавлено (С ВКТ C^Y). Каждое из таких новых аналитических отношений связывается указателем с БУП, который моделирует процесс, определяющий *Y*-переменные и отношения, в данном случае FILLBUCKET. Таким образом, процесс запущен. Теперь супервизор должен обеспечить наблюдение за условиями продолжения. Оно осуществляется различным образом для символической и аналитической компонент условий.

С каждым отношением символической компоненты условий продолжения связан список указателей к тем БУП, которые

моделируют процессы, продолжение которых зависит от этого отношения. Как только это отношение вычеркивается из модели пространства, производится прерывание всех процессов, чьи БУП были связаны с этим отношением.

При создании БУП супервизор конструирует систему уравнений из аналитических компонент результатов — П и условий продолжения. Из этой системы могут быть определены условия прерывания процесса относительно одной из Y-переменных или времени. В нашем примере создается система

$$\begin{cases} C^Y = C_0 + V^c \cdot \tau, \\ C^Y < C_b, \end{cases}$$

из которой может быть определен критический момент естественного завершения процесса $t = \tau_0 + \tau$, где

$$\tau = \frac{C_b - C_0}{V^c}.$$

Время t записывается в БУП и может быть использовано супервизором.

Рассмотрим, как происходит прерывание при нарушении символического отношения $(V \vee V^c)$. Это условие может быть нарушено процессом TURNVALVE. Если скорость потока изменилась до нуля, то отношение $(V \vee V^c)$ вычеркивается из символической части модели пространства. Супервизор проверяет список указателей этого отношения, находит указатель к БУП FILLBUCKET и прерывает этот процесс. При этом аналитическая компонента результатов должна быть преобразована в символические отношения в модели пространства. Исходя из времени прерывания τ, вычисляется C^Y и вводится символическое отношение $(C$ ВКТ $C^Y_\tau)$, где C^Y_τ — значение C^Y в момент τ. Затем БУП FILLBUCKET ликвидируется.

Если скорость потока изменяется до некоторой новой положительной величины, то это изменение также прервет FILLBUCKET, но в данном случае он будет вновь запущен с другим параметром V^c и новым начальным содержимым C'_0, так как все его предусловия будут удовлетворены.

Нам осталось рассмотреть работу супервизора со временем. Как указывалось выше, с каждым БУП связано время прерывания, т. е. число, определяющее самый ранний момент времени, в который процесс должен быть прерван. Супервизор определяет самый ранний из таких моментов для всех действующих процессов, например T_0. Далее супервизор ищет среди всех сценариев процессов такой, который при некоторых значениях его параметров мог бы быть запущен раньше, чем в момент T_0. Если такого сценария нет, то супервизор

устанавливает модельное время в T_0 и выполняет прерывание. Если же таких сценариев несколько, то супервизор находит тот из них, который должен быть запущен раньше других, соотносит модельное время моменту запуска этого процесса, создает для него БУП и переходит к запуску следующего по времени процесса.

Описанный механизм осуществляет моделирование параллельных во времени, независимых и взаимодействующих процессов и действий. Очевидно, что **над этим механизмом следует надстроить планирующую систему, которая могла бы на основе поставленной цели и начальной модели пространства определять последовательность совокупностей процессов, приводящую к цели оптимальным в некотором смысле образом.** Основной для построения такой ПС в значительной степени могут служить обычные ПС типа STRIPS. Однако наличие аналитических компонент предусловий, результатов и условий продолжения приводит к серьезным осложнениям. В частности, главной проблемой является необходимость решения систем уравнений, которые могут быть весьма сложными. Поскольку мы хотели бы, чтобы ПС была достаточно универсальной, она должна быть в этом случае снабжена решателем таких систем, обладающим весьма общими (а следовательно, и слабыми) методами, что существенно снизит эффективность ПС

Выходом из положения может быть моделирование человекоподобного поведения. Действительно, человек, осуществляя одновременно несколько действий в реальном мире, как правило не решает систем уравнений для точного определения моментов времени или переменных процессов, соответствующих завершению или прерыванию этих процессов. Хорошая хозяйка, готовя обед из нескольких блюд, может заниматься уборкой квартиры и делает это, изредка окидывая взглядом все поле своей деятельности, а отнюдь не производя аналитических выкладок.

Мы приходим к выводу о важнейшей роли взаимодействия процессов восприятия, приблизительной оценки и планирования действий в динамическом пространстве. Однако эта задача и связанная с ней задача планирования в динамическом пространстве еще далеки от своего решения.

В настоящее время развивается несколько другой подход к планированию и выполнению действий в динамическом пространстве. Главной особенностью его является децентрализованное взаимодействие специальным образом построенных процедур, или скелетов (фреймов). Этот подход представляет собой вполне естественное развитие процедуральных предписаний в направлении локализации знаний и решения задач в процедурах.

Мы опишем основные идеи, связанные с построением одной из наиболее перспективных, на наш взгляд, реализаций скелетов (фреймов)— **системы взаимодействующих акторов**. Взаимодействие между акторами выражается в передаче предписаний (директив), и это единственный вид внешнего поведения акторов (внешней коммуникацией).

Внутреннее поведение акторов (внутренняя коммуникация) определяется рядом механизмов:

— целью, которая проверяет удовлетворимость всех начальных условий актора, которому передано предписание (директива);

— рядом процедур, обеспечивающих связывание переменных и присвоение последним соответствующих значений;

— синхронизатором, определяющим, когда в действительности должен начать работать актор после передачи ему предписания (директивы), а также предписывающим такой порядок действия акторов, при котором удовлетворяются их цели;

— монитором, воспринимающим все предписания (директивы), переданные актору, и определяющим порядок их обработки;

— «банкиром», управляющим распределением ресурсов памяти и времени для актора.

Следует отметить, что каждый из указанных механизмов определен локально и работает только при вызове актора, что **обеспечивает максимальную степень параллелизма**. Кроме того, локализация, как уже отмечалось ранее, обеспечивает решение проблемы границ.

Рассмотрим схему передачи предписания (директивы) актору. Мы должны различать при этом актор, **посылающий** предписание (директиву) (**отправитель, источник**), и актор, **воспринимающий** его (**получатель, адресат**). В структуру предписания (директивы) входит собственно предписание (директива) *M* и продолжение процесса *C,* которое должно быть активировано после выполнения действий получателем (отсутствие продолжения означает, что отправитель закончил свою работу). Итак, если отправитель *R* посылает получателю *T* предписание (директиву) *M* с продолжением *C,* то схема передачи предписания (директивы) может выглядеть следующим образом:

1) Вызов «банкира» *R,* санкционирующего расходы ресурсов отправителем.

2) Посылка предписания (директивы) «банкиром» *R* синхронизатору *T.*

3) Передача предписания (директивы) от синхронизатора Т мониторам *T.*

4) Посылка предписания (директивы) мониторами *T* к целям *T.*

5) Цели *T* посылают предписание (директиву) *M* продолжениям *T*.
6) Продолжение *T* осуществляет некоторую работу (в том числе передачу предписания (директивы) следующему получателю или, в случае успешного завершения работы, возврат к продолжению С).
Следует отметить ряд особенностей такой схемы.
Во-первых, передача предписания (директивы) актору-получателю сопровождается созданием нового актора, являющегося частным случаем получателя, управляемым предписанием (директивой). В частности, это предполагает повторное использование старого актора, если таковой окажется.
Во-вторых, передача предписания (директивы) включает в себя все необходимые управляющие функции, свойственные процедуральному предписанию, например, вызовы функций, повторения, вызовы параллельных процессов и т. д.
В-третьих, общение между акторами осуществляется децентрализованно, т. е. без каких-либо посредников.
В-четвертых, полная локализация позволяет, за счет отсутствия боковых эффектов, осуществлять одновременное общение актора с несколькими другими.
В-пятых, передача предписаний (директив) носит полностью ненаправленный характер, поскольку не предполагает заранее, что продолжение отправителя когда-либо получит предписание (директиву) (требуемый процесс может быть осуществлен другим способом или отвергнут как неудачный).
В заключение отметим ряд особенностей, связанных с накоплением и реорганизацией знаний в формализме акторов. Эти особенности в большой степени относятся к процедуральному предписанию вообще.
Общей проблемой для ИСФ является добавление информации так, чтобы уже накопленная информация могла остаться без изменений (конечно, если нет необходимости в изменениях). В предписаниях в логических исчислениях мы в большинстве случаев легко делаем это путем добавления аксиом. Процедуральные механизмы не позволяют делать этого так легко. Поэтому очень важно развивать гибкие механизмы накопления процедуральной информации.
Формализм акторов пытается обеспечить следующие возможности:
1) Процедуральное вложение информации, т. е. средства, которыми информация может быть вложена в процедуры.
2) Консервативное развитие, т. е. структурирование информации в виде отдельных боксов ориентированных знаний. При этом имеется возможность создания новых боксов знания и связывания их с уже существующими боксами.

3) Модульная связность, т. е. возможность реорганизовать связи между боксами знаний. При этом модульная эквивалентность означает, что один бокс может быть заменен другим, если все связи между этим и остальными боксами знаний сохраняются теми же.

В настоящее время теория процедурального накопления и тем более обобщения знаний сформулирована недостаточно четко.

Можно выделить ряд исследуемых общих подходов в теории процедурального накопления и обобщения знаний. К ним относятся:

1) Процедуральная абстракция, сводящаяся к обобщению частных случаев (в стиле обобщения треугольных таблиц, п. 10.3.2). Грубо говоря, задача заключается в связывании с полученными частными случаями обобщенных образцов в определенных контекстах. Сюда же относится параметризация некоторых часто встречающихся процедур

2) Исключение абстрактно невозможных случаев Эта техника связана с процецуральной абстракцией и заключается в связывании альтернативе частными случаями и выявлением противоречий

3) Протокольная абстракция, заключающаяся в анализе и связывании протоколов выполнения процедур и преобразовании деревьев протоколов в графы путем исключения неразличимых с точки зрения образцов вершин.

11. Языковые формы представления фреймов

11.1. Структура и задачи подсистемы языковых форм представления фреймов

В данном разделе описана структура и общие принципы построения подсистемы языковых форм представления фреймов.

Кажется естественным потребовать, чтобы ИСФ в части языковых форм представления фреймов умела решать по крайней мере следующие задачи:

1. Понимать предложения естественного языка (возможно, ограниченного).
2. Выводить ответ на основании имеющихся у ИСФ фактов.
3. Выражать ответ в ограниченном естественном языке.
4. Накапливать и корректировать свои знания на основании информации, воспринимаемой в процессе диалога.

На рис. 11.1 приведена структура подсистемы языковых: форм представления фреймов, удовлетворяющая перечисленным выше требованиям.

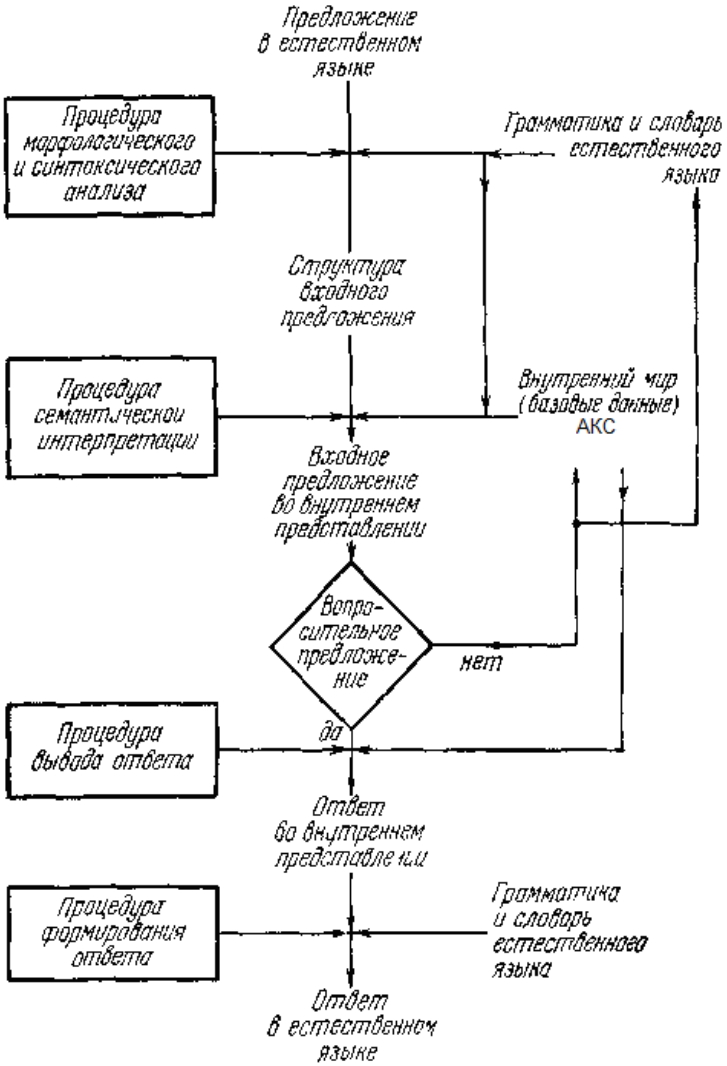


Рис. 11.1. Структура подсистемы языковых форм представления фреймов.

Подсистема, приведенная на рис. 11.1, совпадает по структуре и решаемым задачам с вопросно-ответными системами (ВОС) общего типа.

В дальнейшем в основном будет использоваться термин ВОС вместо термина «подсистема языковых форм представления фреймов» Этим мы хотим подчеркнуть, что изложенный в разделе подход может быть использован не только при построении подсистем ИСФ, но и при разработке изолированных информационно-поисковых или вопросно-ответных систем, воспринимающих ограниченный естественный язык.

Рассмотрим структуру подсистемы языковых форм представления фреймов (рис. 11.1), подразделяемую в соответствии с традициями ВОС на следующие основные этапы.

1. Синтаксический анализ. Синтаксис понимается в широком смысле с включением морфологии. Задача этапа — определить структуру входного предложения в соответствии с грамматикой языка и идентифицировать слова предложения со словарем системы.

2. Семантическая интерпретация. Задача этапа— понять входное предложение, т. е. однозначно выразить его в понятия внутреннего пространства ИСФ. Если входное предложение является *директивой,* то требуется объединить его с базовыми данными (внутренним пространством). Если входное предложение является *вопросом,* то требуется выделить из базовых данных информацию, необходимую для ответа на поставленный вопрос.

3. Вывод ответа. Задача этапа — вывести из выделенного подмножества базовых данных ответ на заданный вопрос. Ответ выражается в понятиях внутреннего представления.

4. Формирование ответа. Задача этапа — перевести ответ из внутреннего представления в естественный язык.

В данном разделе нашей основной задачей является выделение семантической информации входного предложения и установление взаимосвязи ее с информацией ИСФ. Решение указанной задачи в значительной степени не зависит от морфологии и частично от синтаксиса языка. Поэтому мы не будем рассматривать вопросы морфологии и будем касаться синтаксиса только в его связи с семантикой.

Системы, воспринимающие естественный язык, основываются на том положении, что для понимания языка необходимо не последовательное применение этапов обработки входного предложения (**синтаксис, семантика, вывод**), а **интегральное использование всех аспектов языка**. Так, например, для понимания синтаксиса может потребоваться обращение к семантике, к дедукции или к общим знаниям пользователя. Тем не менее мы, в целях методичности изложения, будем описывать этапы работы ВОС последовательно.

На наш взгляд, еще не существует общей теории вопросно-ответных систем, воспринимающих естественный язык. Выполнение ряда этапов

ВОС (семантическая интерпретация, дедуктивный вывод) в значительной степени зависит от многих факторов и особенно от представления базовых данных в виде исчисления предикатов (системы с общим выводом, см. п. 11.3.4) или в виде семантических сетей и процедурального представления (системы с ограниченной логикой, см. п. 11.3.3). Указанное обстоятельство нашло отражение и в изложении материала в даннои разделе. Мы будем описывать этапы семантической интерпретации и дедуктивного вывода отдельно для систем с **общим выводом (исчисление предикатов) и систем с ограниченной логикой (семантические сети).**

Прежде чем перейти к описанию вопросно-ответных систем, отметим, что обработка естественного языка является основной задачей не только ВОС, но и области научных исследований, называемой «машинным переводом» (МП). Несмотря на общность ряда этапов ВОС и МП, мы не будем проводить параллели между используемыми в них принципами. По нашему мнению, различие целей ВОС и МП вызывает (по крайней мере в настоящее время) различие в используемых методах.

Перечислим основные различия ВОС и МП.

1. Основной задачей ВОС по обработке языка является преобразование текста в его смысл (при обработке входного предложения) и преобразование смысла в текст (при формировании ответа).

При МП полного проникновения в смысл фразы может не требоваться. Действительно, если осуществляется перевод с языка Я1 на язык Я2 (рис. 11.2), то априори не видно причин, почему необходимо полное понимание смысла текста (путь Я1→С→Я2 на рис. 11.2), а не частичное (путь Я1 →С1→С2→Я2).

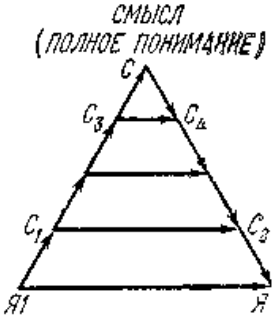


Рис. 11.2. Интерпретация задач машинного перевода и вопросно-ответных систем.

Использование подхода СМЫСЛА $\rightleftarrows$ ТЕКСТ наметилось и в МП.

2. При создании ВОС мы можем ограничить естественный язык как по тематике, так и по грамматике, и при этом не потеряем практической ценности работы. При МП требуется иметь дело с естественным языком. Попытки ограничиться определенной тематикой не делают задачу настолько простой, чтобы удалось получить практически значимые результаты. Как следствие указанного фактора, в ВОС могут быть использованы более формализованные лингвистические модели, существенно упрощающие естественный язык и позволяющие получить эффективные алгоритмы обработки текста.

Методы, используемые при разработке большинства этапов ВОС, в существенной степени зависят от лингвистической модели, принятой для описания языка. Поэтому в очередном параграфе приведены сведения из лингвистики, необходимые для понимания последующего материала.

11.2. Формальные грамматики

11.2.1. Основные определения

Будем называть ***языком*** **конечное или бесконечное множество предложений, каждое из которых имеет конечную длину и построено с помощью операции соединения из конечного множества элементов**. Это определение включает как естественные, так и искусственные языки логики и программирования.

Чтобы точно определить язык, необходимо установить общие принципы, которые отделяют последовательности атомарных элементов, являющихся предложениями, от последовательностей, таковыми не являющихся. Это различие нельзя выразить просто списком, так как для всех систем, представляющих интерес, не накладывается ограничений на количество предложений и их длину. Традиционные грамматики естественных языков не содержат всей информации, необходимой для выполнения указанной задачи. Стремление к созданию более строгой лингвистической теории привело к разработке формальных грамматик, основоположником которых считается Н. Хомский. Он указал на возможность использовать для описания естественных языков некоторые исчисления, рассматривавшиеся ранее в теории алгоритмов, придав им удобную для этого форму (за ними закрепилось данное Н. Хомским название «порождающих грамматик»).

Следуя Хомскому, предъявим к лингвистической теории два требования:

1) порождать все те и только те предложения, которые доставляют описываемый данной грамматикой язык;
2) определять метод, с помощью которого можно было бы дать единственное и единообразное структурное описание каждого предложения, порожденного грамматикой. Было бы удобно, чтобы структурное описание получалось автоматически при порождении предложения.

11.2.2. Формальные грамматики

Введем понятия **цепочек и языков**, а затем определим понятие **порождающей грамматики**.
Пусть *V*— непустое конечное множество, которое мы будем называть *словарем (алфавитом),* а его элементы — *символами (буквами).* Произвольную конечную последовательность элементов ω будем называть *цепочкой* в словаре *V*. Пустую цепочку будем обозначать символом Λ. Число символов в цепочке будем называть *длиной цепочки* и обозначать |ω|. Над цепочками определяется *операция конкатенации.* Конкатенацией непустых цепочек $b_1 \ldots b_n$ и $c_1 \ldots c_p$ называется цепочка $d_1 \ldots d_{n+p}$, где $d_1= b_1, \ldots, d_n=b_n, d_{n+1}=c_1, \ldots, d_{n+p}=c_p$. Конкатенацию цепочек ω и φ мы будем обозначать ωφ. Кроме того, конкатенация цепочки ω и Λ, равно как Λ и ω, считается по определению равной ω.
Если для каких-либо цепочек ω, φ, η_1, η_2 в словаре *V* имеет место равенство $\omega=\eta_1\varphi\eta_2$, то будем называть цепочку $\eta_1*\varphi*\eta_2$, где *с*имвол * не принадлежит *V, вхождением цепочки* φ в ω. Вхождение символов в цепочку будем называть ее *точками.* Если $\alpha= \eta_1*b*\eta_2$ и $\beta = \xi_1*c*\xi_2$— точки одной и той же цепочки $\omega = \eta_1*b*\eta_2 = \xi_1*c*\xi_2$ и если при этом $|\eta_1|<|\xi_1|$, то мы будем писать $\alpha < \beta$ или $\beta > \alpha$ и говорить, что α расположена левее β, а β — правее α. Для любых двух точек α и β цепочки ω таких, что $\alpha \le \beta$, мы будем называть множество точек ξ, удовлетворяющих неравенствам $\alpha \le \xi \le \beta$, *отрезком цепочки* ω.
Произвольное множество цепочек в словаре *V* мы будем называть *языком* в этом словаре. Естественно, что задание языков практической сложности перечислением всех цепочек, составляющих язык, нецелесообразно. Языки задаются с помощью формальных грамматик, порождающих все цепочки данного языка и только их.
***Порождающая грамматика* — это упорядоченная четверка**
$\Gamma= (V_т, V_н, S, P)$, где $V_т$ и V_H — непересекающиеся непустые конечные множества; *S* — некоторый элемент из V_H;

P —конечное множество правил вида φ→ψ, где φ и ψ — произвольные цепочки словаря $V_T \cup V_н$ и символ → не входит в $V_T \cup V_H$.

Множества V_T и V_H называются соответственно *терминальным (основным)* и *нетерминальным (вспомогательным) словарями,* а их элементы соответственно терминальными и нетерминальными символами грамматики Г. Объединение $V_T \cup V_H$ будем называть *полным словарем* грамматики Г.

S называется *начальным символом* Г. Это выделенный нетерминальный символ, обозначающий класс всех тех языковых объектов, для описания которых предназначена данная грамматика. Иногда символ S называют *целью грамматики.*

Отметим, что при изучении естественного языка в аспекте теории формальных грамматик терминальные символы интерпретируются как *словоформы* (**словоформами называют единицы языка, рассматриваемые одновременно в плане выражения (последовательность букв от пробела до пробела**) и в плане содержания (совокупность значений)) или *морфемы* (**морфемами называют наименьшие осмысленные единицы языка (корни, суффиксы и т. п.**)), нетерминальные символы — как названия *классов слов* и *словосочетаний,* а начальный символ— как *предложение.*

P называют *схемой грамматики* Г, а цепочки вида φ→ψ называют *правилами* Г.

Пусть *r*=φ→ψ — некоторое правило грамматики Г и ξ_1*φ*ξ_2 — вхождение φ в цепочку $\omega=\xi_1\varphi\xi_2$ в словаре $V_T \cup V_H$. Говорят, что цепочка $\eta=\xi_1\varphi\xi_2$ получается из ω применением правила *r* к вхождению φ в цепочку ω. Если цепочка η получается из цепочки ω применением какого-либо правила Г, будем говорить, что η *непосредственно выводима* из ω в Г и будем писать ω |= η.

Последовательность цепочек D= (ω_o, ω_1, . . ., ω_n), $n\geq1$, называется *выводом* ω_n *из* ω_o *в грамматике* Г, если для каждого *i*, $1\leq i\leq n$, имеет место $\omega_{i-1}|=\omega_i$. Число *n* называется длиной вывода D. Если существует вывод цепочки η из ω в Г, то будем говорить, что η *выводима из* ω в Г, и писать ω|— η.

Множество цепочек в основном словаре грамматики Г, выводимых из ее начального символа, называется *языком, порождаемым грамматикой* Г, и обозначается *L(T).*

Из приведенных выше определений видно, что **грамматика — это исчисление, т. е. разрешение производить некоторые операции** — в данном случае подстановки, а не алгоритмы, т. е. **предписание производить операции (директивы)**.

Грамматики, на правила которых не наложены никакие ограничения, способны порождать любые множества цепочек, порождаемые каким-либо автоматом. В теории алгоритмов такие множества называются *рекурсивно-перечислимыми.* Языки, порождаемые этими грамматиками, образуют слишком широкий класс и не представляют интерес для лингвистики.

Естественно полагать, что для множества цепочек, составляющих некоторый естественный язык, у носителей языка есть распознающий алгоритм, т. е. **способ узнавания принадлежности каждой предъявленной цепочки к цепочкам данного языка**. Более того, этот алгоритм должен давать ответ довольно быстро. **Множества, для которых существуют <u>распознающие алгоритмы</u>, называются *рекурсивными.***

Мы будем рассматривать грамматики, порождающие рекурсивные множества.

При изучении формальных грамматик выделяют **три типа грамматик**, представляющих наибольший интерес как в теоретическом, так и в практическом смыслах. Эти грамматики задаются путем наложения последовательно усиливающихся ограничений на правила *P*.

Грамматика Г называется *грамматикой составляющих* или *непосредственно составляющих* ***(НС-грамматикой),*** если каждое ее правило имеет вид $\xi_1 A \xi_2 \rightarrow \xi_1 \psi \xi_2$, где ξ_1 и ξ_2— произвольные цепочки в словаре $V_T \cup V_H$, $A \in V_H$ и ψ — произвольная непустая цепочка в $V_T \cup V_H$. При применении НС-правила одно вхождение символа *A* заменяется на ψ в зависимости от наличия нужного контекста. Поэтому данные грамматики называют также *контекстными,* или *контекстно-чувствительными. Грамматика* Г называется *бесконтекстной* или ***контекстно-свободной*** **(КС**), если все ее правила имеют вид $A \rightarrow \psi$, где *A* — нетерминальный символ, а ψ — произвольная непустая цепочка в $V_T \cup V_H$.

Грамматика Г называется *автоматной грамматикой* или ***A*-грамматикой** с конечным числом состояний, если каждое ее правило имеет вид $A \rightarrow aB$ или $A \rightarrow a$, где $a \in V_T$, A и $B \in V_H$.

В указанной последовательности классов порождающих грамматик, каждый последующий класс (в смысле общности правил грамматики) является частью предыдущего.

Языки, порождаемые грамматиками перечисленных классов, называются соответственно *языками непосредственно составляющих,* или *контекстно-чувствительными, бесконтекстными и автоматными.*

Как было указано ранее, задача грамматики состоит не только в том, чтобы породить предложения языка, но и приписать каждому из них **структурное описание**. В современной лингвистике наиболее употребительны два способа описания синтаксической структуры предложения:

1) описание с помощью систем составляющих;
2) описание с помощью деревьев синтаксического подчинения.

Пусть *x* — произвольная непустая цепочка в словаре *V*. Множество С отрезков цепочки *x* называется *системой составляющих* этой цепочки, если оно удовлетворяет двум условиям:

1) множество *C* содержит отрезок, состоящий из всех точек цепочки *x,* и все одноточечные отрезки *x;*
2) любые два отрезка из С либо не пересекаются, либо один из них содержится в другом.

Элементы *C* называют *составляющими.* Одноточечные отрезки называют *точечными составляющими;* отрезок, состоящий из всех точек цепочки,— *полной составляющей;* полную и точечные составляющие называют *тривиальными.*

Для наглядности изображения системы составляющих иногда заключают в скобки каждую нетривиальную составляющую. Скобки можно не нумеровать, так как в силу пункта 2 определения системы составляющих каждой левой скобке можно однозначно указать соответствующую ей правую.

Если цепочка *x* интерпретируется как предложение естественного языка, то система составляющих может быть использована в качестве **способа выражения информации** о его синтаксической структуре. Такая **информация может представлять собой <u>список словосочетаний</u>, то есть тех «кусков» предложения, которые в каком-то интуитивном смысле являются «синтаксически связанными».** Эмпирические соображения позволяют сделать допущения, что словосочетания, во-первых, образуют отрезки и, во-вторых, не имеют «частичных пересечений», т. е. удовлетворяют п. 2 определения системы составляющих.

Таким образом, **<u>нетривиальными составляющими являются словосочетания</u>** (при подходящем выборе системы составляющих). Тривиальные составляющие добавлены для придания системе формальной законченности.

Среди многочисленных систем составляющих, которые имеет предложение естественного языка, лишь весьма немногие «правильны», то есть адекватно отражают синтаксическое строение предложения. При этом понятие «правильной» системы составляющих не абсолютно, оно зависит от соглашений лингвистического характера,

отражающих определенные содержательные представления о синтаксической структуре предложения данного языка. Нетрудно видеть, что системе составляющих можно поставить в соответствие дерево корнем которого служит **полная составляющая**, а висячими узлами — **точечные составляющие**. Это дерево называется **деревом составляющих**. Приведем пример (рис. 11.3) дерева составляющих для фразы

(Онегин, (добрый (мой приятель))),
(родился (на (брегах Невы))).

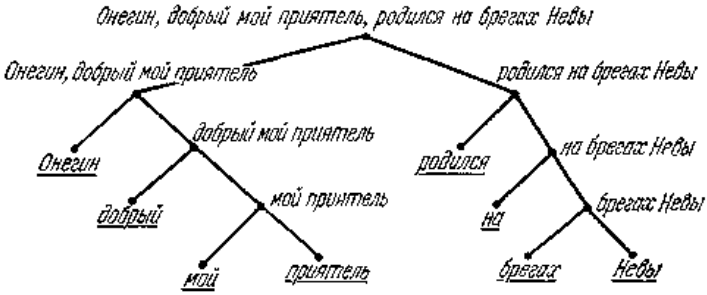


Рис. 11.3. Пример дерева составляющих.

Построенная система составляющих указывает в предложении словосочетания разных «уровней» (составляющие разной высоты), но не вводит при этом никакой иерархии среди словосочетаний одного «уровня». Между тем, в предложениях естественного языка часто интуитивно ощущается «главенствование» некоторого словосочетания над другими, в нем не содержащимися.

Чтобы в некоторой степени отразить этот факт, вводят иерархизованную систему составляющих следующим образом. Пусть С — система составляющих цепочки *x*. Для каждой неточечной составляющей $A \in C$ выделим во множестве всех составляющих, непосредственно вложенных в *A,* какую-либо одну составляющую *A',* которую будем называть *главной.* Множество всех главных составляющих обозначим С' и назовем *иерархизацией системы С.* Упорядоченную пару *(С, С')* назовем *иерархизованной системой составляющих.*

При описании предложений естественного языка с помощью системы составляющих *С* часто используют **способ введения в эту систему <u>дополнительной информации</u>** путем отображения *С* во множество всех подмножеств некоторого конечного множества, элементы которого называются *метками* и содержательно интерпретируются как

символы синтаксических классов слов и словосочетаний. Упорядоченную тройку (C, *W*, φ), где *C*—система составляющих, *W*— множество меток и φ — отображение *C* в 2^W, называют *размеченной системой составляющих.* Пусть множество меток *W*, состоит из элементов, указанных в таблице 11.1.

Таблица 11.1

Члены множества *W*	Содержательная интерпретация слов и словосочетаний
S	Предложение
VP_{xyvw}	Группа глагола } в роде x, числе y, времени
V_{xyvw}	Глагол } v и лице w.
NP_{xyz}	Группа существительного } рода x, в числе y
N_{xyz}	Существительное } и падеже z
A_{xyz}	Прилагательное в роде x, числе y и падеже z
PP	Предложная группа
$ПР$	Предлог «на»

В ней одновременно с перечислением меток приведена их содержательная интерпретация.

Тогда для приведенного ранее примера получим следующую «естественную» разметку (рис. 11.4).

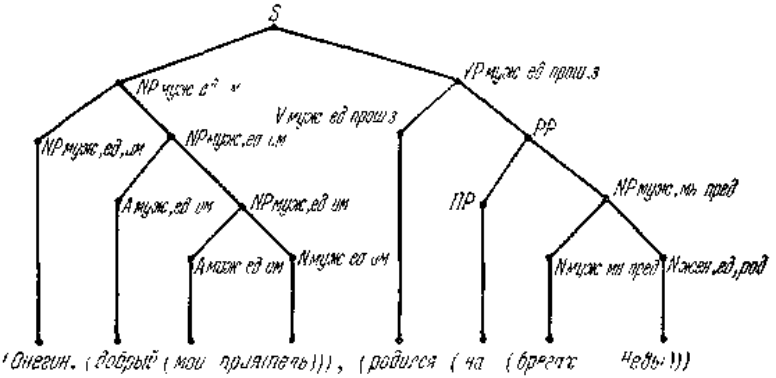


Рис. 11 4. Пример размеченного дерева составляющих.

Заметим, что вывод цепочки в НС-грамматике можно также представить в виде дерева. При этом, если каждому нетерминальному

символу *A,* заменяемому в правиле грамматики $\xi_1 A \xi_2 \rightarrow \xi_1 \psi \xi_2$ на цепочку ψ, поставить в соответствие вершину, из которой исходят ребра к символам, образующим цепочку ψ, также интерпретируемым как вершины, то выводу цепочки будет соответствовать дерево. Нетрудно видеть, что это дерево является деревом составляющих. В теории порождающих грамматик его часто называют *С-маркером*
Тот факт, что НС-грамматика при порождении терминальных цепочек одновременно дает их дерево составляющих, делает НС-грамматики особенно интересными с лингвистической точки зрения.
Определим теперь дерево подчинения.
Пусть *x* — произвольная непустая цепочка в словаре *V* и X — множество всех точек в *x*. Произвочьное бинарное отношение → на *X,* при котором граф <*X;* →> является деревом, будем называть *отношением синтаксического подчинения* или просто *отношением подчинения* для *x.* Само дерево <*X;* →> будем называть *деревом (синтаксического) подчинения* для *x.*
При графическом изображении дерева подчинения точки цепочки *x* помещают на горизонтальной прямой (рис. 11. 5), и для всякой пары точек α, β, для которой α→β будем проводить из α в β стрелку, таким образом, чтобы все стрелки были по одну сторону от прямой.
На рис. 11.5 изображено несколько различных деревьев подчинения для одной и той же цепочки *abcdefg.*

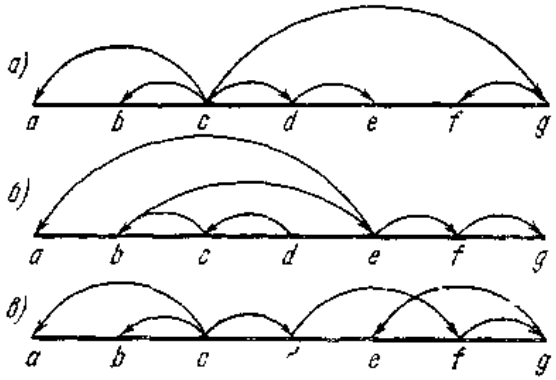
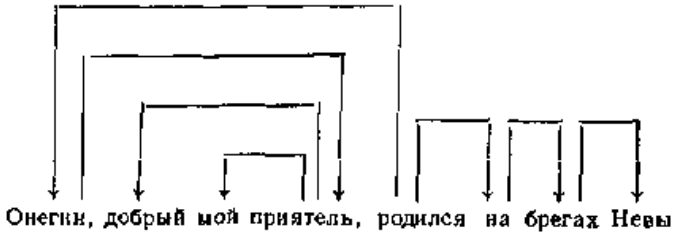


Рис. 11.5. Возможные деревья подчинения для цепочки *abcdefg.*

Деревья подчинения могут быть использованы как один из способов изображения синтаксической структуры предложения. Именно **<u>информация о синтаксическом строении предложения</u> может представлять собой набор сведений о «главенствовании» одних слов (точнее, вхождений слов) в предложении над другими**;

задать такой набор — значит задать некоторый граф на множестве точек цепочки. Из интуитивных соображений вытекает, что этот граф можно считать деревом.

Приведем пример дерева подчинения для предложения из рис. 11.4, соответствующего «естественным» представлениям о главенствовании одних слов над другими.



Понятие «правильного» дерева подчинения, так же как и системы составляющих, для естественного языка зависит, вообще говоря, от некоторых лингвистических соглашений.

Для анализа предложений естественного языка часто используются размеченные деревья подчинения, в которых, кроме подчинения, указывается его вид.

В классе всевозможных деревьев подчинения выделяют подкласс, который содержит подавляющее большинство «естественных» деревьев для предложений реальных языков. Это класс так называемых *проективных деревьев.*

Дерево подчинения $<X; \rightarrow>$ для цепочки x, а также соответствующее отношение подчинения $\rightarrow$, называется *проективным,* если для любых трех точек α, β, γ цепочки x из того, что $\alpha \rightarrow \beta$ и γ лежит между α и β, следует, что γ зависит от α (т. е. в дереве существует путь из α в γ).

Имеется еще один важный класс деревьев подчинения, являющийся расширением предыдущего: класс *слабо проективных деревьев.* Дерево подчинения $<X; \rightarrow>$ для цепочки x (и отношение $\rightarrow$) называется *слабо проективным,* если для любых четырех точек α, β, γ, δ цепочки x из $\alpha \rightarrow \beta$ и $\gamma \rightarrow \delta$ следует, что пары α, β и γ, δ не разделяют друг друга. *(Пары точек α,* β и γ, δ *разделяют друг друга,* если одна из точек γ, δ лежит, а другая не лежит между α и β.)

При графическом способе изображения слабая проективность равносильна возможности провести все стрелки так, чтобы никакие две из них не пересекались, а проективность, кроме того, обозначает, что корень дерева не лежит ни под какой стрелкой.

На рис. 11.5 дерево *а*) проективно, деревья *а), б)* слабо проективны, дерево *в*) непроективно.

Содержательный смысл условий проективности и слабой проективности обозначает, что слова, близкие синтаксически, близки и по положению в тексте.

В так называемой научной и деловой прозе, по крайней мере русской, естественные деревья подчинения подавляющего большинства предложений слабо проективны и даже проективны.

В рассматриваемом нами применении (фреймы) указанное ограничение не является существенным, но упрощает представление синтаксической структуры предложения.

Можно показать, что существуют правила перехода от систем составляющих к деревьям подчинения и обратно.

Рассмотрим вопрос о возможности использования введенных формальных грамматик для описания естественного языка. В литературе подробно освещен вопрос о недостаточности **А-грамматик и КС-грамматик** для описания естественных языков в полном объеме. Заметим, однако, что это не исключает их использования на определенных уровнях описания грамматики.

Что касается **НС-грамматик** и равных им по порождающей силе *неукорачивающих грамматик,* то есть грамматик с правилами $\varphi \rightarrow \psi$, удовлетворяющими тому требованию, что длина цепочки ψ не короче цепочки φ, то они достаточны (хотя и не обязательно удобны) для описания любых естественных языков в полном объеме. Это утверждение вытекает из следующих допущений:

1) любой естественный язык, точнее, множество его правильных фраз, есть легко распознаваемое множество, т. е. существует достаточно простой алгоритм распознавания правильности фраз. Данное допущение является практически очевидным, так как люди обладают таким алгоритмом;

2) алгоритм распознавания правильности фраз естественного языка должен обеспечивать такой процесс распознавания, при котором требуемый объем «оперативной памяти» сопоставим с длиной фразы, например, не превышает числа Mn, где n— длина фразы, а M — достаточно большая константа.

Последнее допущение подтверждается психологическими экспериментами. Автоматы, на работу которых накладывается такое ограничение, называются *линейноограниченными.* Как известно, ***множество цепочек, представимых линейноограниченным автоматом, есть <u>контекстно-чувствительный язык</u>.*** Таким образом, из наших допущений следует, что НС-грамматики в принципе способны описывать множество правильных фраз любого естественного языка.

Однако, приведенные выше рассуждения не гарантируют удобства описания любого естественного языка.

Отметим основные недостатки НС-грамматик:
1) с помощью НС-грамматик не удается естественно описывать фразы, содержащие непроективные конструкции;
2) НС-грамматика, как и любая порождающая грамматика в смысле данного нами определения, содержит только правила образования языковых выражений, но не содержит правил преобразования правильно построенных выражений.
В литературе по математической лингвистике кроме термина «порождающие грамматики» используется и термин ***«распознающие грамматики»***. Деление грамматик на порождающие и распознающие имеет историческое объяснение, хотя по существу порождающие грамматики наиболее интересных типов, в частности, всех типов рассмотренных нами, могут быть использованы также и для распознавания. При этом, если вместо распознавания говорить о «допускании», то все порождающие грамматики могут трактоваться как *допускающие*. ***Грамматика* Г *допускает язык L,* если для Г известна процедура, определяющая для любой цепочки *x* ($x \in L$) ее принадлежность к языку L, если же *x* не принадлежит L, то от процедуры не требуется никакого ответа.** При этом допускающая процедура для любой порождающей грамматики состоит просто в применении правил к данной цепочке справа налево, а не как обычно слева направо, до тех пор пока это возможно. Верно и обратное, что распознающие грамматики, например, категориальные грамматики, могут использоваться для порождения фактически опять же применением правила в обратном порядке.
Таким образом, формальные грамматики по существу нейтральны по отношению к порождению и допусканию, а также распознаванию для рассмотренных нами грамматик. Можно говорить просто о формальных грамматиках, рассматривая отдельно от самих грамматик аспект направления применения правил.

11.2.3. Трансформационные порождающие грамматики (ТПГ)

Владение языком предполагает умение не только построить правильную фразу, но и перейти от одной фразы к другим, либо полностью синонимичным ей, либо отличающимся от нее по смыслу на определенную «величину». Примером таких переходов является переход от утвердительного предложения к вопросительному или отрицательному, переход от активной формы к пассивной, выражение одной и той же мысли разными способами и т. д.

В НС-грамматике все такие предложения будут порождаться более или менее независимо и, следовательно, не будут находиться в каких-либо явных отношениях друг к другу.

Соответствующая задача была впервые четко сформулирована Н. Хомским. Выдвинутая им концепция приобрела широкую популярность под названием **«трансформационной грамматики»**. Фактически, дело заключается во введении еще одного семантического уровня описания языка. В самом деле, **инвариантом всех трансформаций (преобразований) является смысл.** Таким образом, **теория трансформаций оказывается по существу теорией синонимии в языке**.

Отметим, что НС-грамматики описывают синтаксический уровень в широком смысле, т. е. с включением морфологии, а **трансформационные грамматики включают и семантические преобразования**. Поэтому, когда говорят о недостаточности НС-грамматик для описания языка, следует понимать, что это верно только в смысле неохвата НС-грамматиками семантического уровня. На своем, чисто синтаксическом уровне, НС-грамматики оказываются принципиально вполне достаточными.

Основные положения ТПГ изложены в ряде работ Хомского. Согласно этим представлениям, ТПГ состоит из **трех основных компонентов**: ***синтаксического, фонологического, семантического***. По Хомскому, собственно порождающей частью теории является только синтаксическая компонента, где по правилам грамматики происходит порождение *глубинной и поверхностной структур.* **Семантическая и фонологическая компоненты интерпретируют** соответственно **глубинные структуры и поверхностные**. Таким образом, центральная роль отводится синтаксической компоненте, состоящей из двух субкомпонент: *базовой* (или просто *базы)* и *трансформационной.* База синтаксической компоненты служит для порождения *глубинных структур.* В содержательном отношении глубинная структура является представлением логического содержания порождаемого предложения в терминах **элементарных суждений**. Задача трансформационной субкомпоненты состоит в том, чтобы перейти от глубинной структуры к предложению в естественном языке *(поверхностной структуре)* либо полностью синонимичному с ней, либо отличающемуся от нее по смыслу на некоторую величину.

Формально глубинные структуры представляют собой деревья *(С-маркеры),* порождаемые *категориальными правилами* (категориальная компонента) и лексиконом. Категориальные правила представляются в виде правил НС-грамматики, не содержащих терминальные символы, т. е. порождаемые ими цепочки состоят только

из категориальных символов *(N*— существительное, *V*— глагол, *D* — детерминатив и т. п.)

Бесконечная порождающая способность грамматики категориальной компоненты вытекает из того факта, что допускается введение начального символа *S* (предложения) в строки вывода. Таким образом, правила подстановки могут, по существу, включать одни базовые С-маркеры в другие.

Подстановка терминальных символов в порождаемые предложения осуществляется с помощью **трех видов правил**: правила *субкатегоризации,* правил *селекции* и правил *лексического включения.* Субкатегориальные правила и правила селекции комбинируют категориальные символы вместе с субкатегориальными в *комплексные символы,* осуществляя замены типа *N*→[+*N,* + Нарицательность, + Исчисляемость, + Одушевляемость, — Абстрактность]. Правила лексического включения подставляют в порождаемое дерево вместо правой часги приведенного выше правила лексическую единицу с подходящими набором признаков, например «мальчик» (см. табл. 11.2)

Таблица 11.2

Фрагмент лексикона трансформационной грамматики

C	*D*
[+ *N*, + Нарицательность, + Исчисляемость, + Одушевленность, + Человечность]	мальчик
[*V*+, + Транзитивность, [+ Абстрактность] *AUX_DET* [+ Одушевленность]]	испугать
[+ *M*,]	мочь
[+ *N*, + Нарицательность, — Исчисляемость, + Абстрактность]	искренность

Итак, лексикон состоит из записей, связанных с трансформациями подстановки, которые вводят лексические единицы (лексемы) в цепочки, порожденные категориальным компонентом, образуя так называемый *обобщенный С-маркер.* Все контекстуальные ограничения в базе обеспечиваются этими трансформационными правилами лексикона *(лексическими трансформациями).*

Отметим, что даже простому предложению в глубинной структуре может соответствовать система из нескольких элементарных суждений. На рис. 11.6 приведена глубинная структура простого предложения «Невидимый бог создал видимый мир».

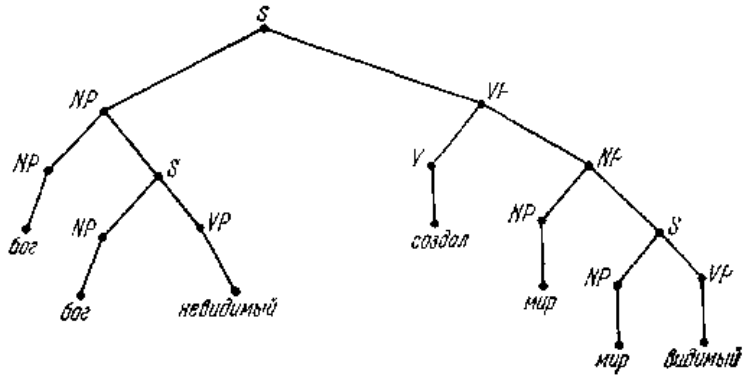


Рис. 11.6. Глубинная структура предложения «Невидимый бог создал видимый мир».

Как уже было указано ранее, задача трансформационной субкомпоненты состоит в том, чтобы из глубинной структуры, порожденной базой, получить поверхностную структуру, используя *грамматические (нелексические) трансформации.* Каждая трансформация представляется в виде правила, условие применимости которого задается в виде С-маркера.

Имея обобщенный С-маркер, мы строим трансформационный вывод, применяя нелексические трансформационные правила последовательно «снизу вверх». Другими словами, мы применяем последовательность правил к данной конфигурации только в том случае, если мы уже применили ее ко всем базовым С-маркерам, вставленным в эту конфигурацию.

Итак, мы определили ТПГ, которая включает систему правил структуры составляющих, порождающих деревьями систему трансформационных правил, отображающих деревья в деревья. Такая грамматика задает бесконечный класс конечных последовательностей деревьев $(P_1, \ldots, P_n)$. Каждая последовательность, называемая выводом, удовлетворяет следующим условиям:

P_1—дерево, порождаемое правилами категориального компонента;

P_n — дерево, представляющее поверхностную структуру порожденного предложения;

P_i (для любого i, кроме i=l) образовано применением к дереву P_{i-1} одного из трансформационных правил.

Отметим, что с каждой **лексической единицей** связана определенная лексическая трансформация, включающая ее в дерево.

В соответствии с требованиями теории Хомского подобные лексические трансформации предшествуют нелексическим трансформациям, и границей между двумя видами трансформаций является глубинная структура.

Полученная в результате трансформации поверхностная структура отображается с помощью фонологических правил на фонетические представления. Система таких правил образует *фонологическую компоненту* (рис. 11.7).

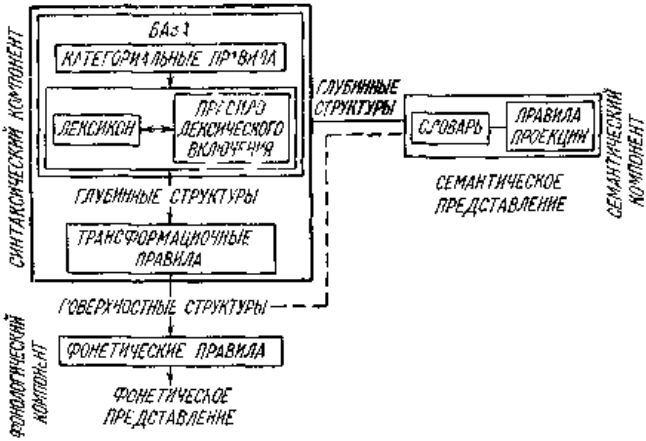


Рис. 11.7. Схема теории трансформационных порождающих грамматик («Стандартная теория»).

С другой стороны, глубинные структуры, после того как в них введены конкретные лексемы, отображаются с помощью **семантической компоненты на семантические представления.** Семантическая компонента состоит из *системы проекционных правил* и *словаря,* где каждая единица получает словарную статью, состоящую из стольких прочтений (readings), сколько у единицы значений.

Правила проекции идут снизу по дереву и сочетают прочтения единиц, образующих составляющие, учитывая при этом имеющиеся в прочтениях селекционные ограничения, запрещающие те или иные сочетания значений. **Получившийся и результате суммарный смысл и представляет собой семантическую интерпретацию, или прочтение данной глубинной структуры**

Катц и Постал показали, что возможна такая перестройка теории, при которой трансформации никогда не меняют смысла, и вся **смысловая**

информация предложения содержится уже в глубинной структуре. Поэтому семантической компоненте достаточно ее только интерпретировать.

Изложенная выше теория ТПГ называется «стандартной теорией» Хомского. В дальнейшем некоторые новые лингвисчические факты заставили Хомского признать в рамках «пересмотренной стандартной теории» зависимость семантического представления не только от глубинной, но и от поверхностной структуры (см. пунктир на рис. 11.7) Концепцию семантической компоненты, развитую в рамках «пересмотренной стандартной теории», называют *интерпретирующей семантикой* (ИС).

Существенный вклад в разработку теории глубинных структур в рамках интерпретирующей семантики внес Ч. Филмор. В отличие от Хомского Ч. Филмор считает, что основу предложения образует не **субъектно-предикатная**, а **предикатно-аргументная структура**. **Аргументами такой структуры Филмор считает имена, для которых может быть указан глубинный падеж.** При этом глубинные падежи выявляются в результате анализа так называемых скрытых категорий, проявляющихся в характере проведения трансформаций, перифразов и т. д. Падеж при таком понимании рассматривается как универсальное явление, присущее всем языкам независимо от того, имеется ли в них падеж в традиционном смысле или нет. Итак, ***глубинный падеж* — это обобщенное отношение между содержанием глагола и содержанием той или иной из его именных групп.** Ч. Филмор предлагает использовать **семь глубинных падежей**:

— агентивный *(A)* — одушевленный возбудитель лействия *(Джон* открыл дверь);

— инструментальный (*I*) — падеж неодушевленного предмета или силы, составляющих причину глагольного действия (состояния) *(Камень* разбил стекло.);

— дательный (D) — падеж одушевленного существа, затронутого глагольным действием (состоянием) *(Мальчик* получил удар в лицо.);

— фактитивный *(F)* — падеж предмета или существа, возникающего в результате действия (состояния) или входящего как часть в само глагольное действие (Мать варит *картошку.);*

— локативный *(L)* — место или пространственная ориентировка глагольного действия (состояния) (Джон идет *по улице.);*

— бенефактивный *(B)* — падеж пользователя (Мать варит картошку *для Джона.);*

— объективный (*O*) — немаркированный падеж, падеж любой вещи, представленной в виде имени, роль которой по отношению к

глагольному действию (состоянию) определяется из семантического толкования самого глагола (Джон открыл *дверь.).*

Филмор подчеркивает, что состав набора, а также характеристики и названия отдельных падежей не являются окончательными.

Глубинная структура по Ч. Филмору имеет следующий вид:

S→Aux+P, где S— предложение, *Aux* — модальный показатель (modality), *P*— пропозиция (proposition). Пропозиция может быть развернута в формулу следующего вида:

$$P \rightarrow V + C_1 + \ldots + C_n,$$

где *V*— глагол, $C_1 + \ldots + C_n$ — глубинные падежи.

С помощью подобных правил порождаются *претерминальные цепочки,* графическое представление которых дано на рис. 11.8.

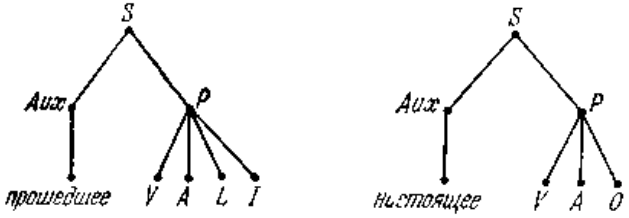


Рис. 11.8. Претерминальные цепочки падежной грамматики.

Отметим, что в данной грамматике, как и у Хомского, предполагается наличие *лексикона, правила лексического включения* и *трансформационных правил.*

Итак, по правилу лексического вывода, замена *комплексного символа C* претерминальной цепочки на некоторую словоформу *D* происходит при отождествлении этого символа с входной информацией *C* этой словоформы в словаре (табл. 11.3).

Таблица 11.3

Фрагмент лексикона в падежной грамматике

C	D
[+A]	*by NP*
[+I]	*with NP* (если во всей фразе уже есть *by*)
+[—ALI]	*smear* (*мазать*)
+[—AOD]	*show* (*показывать*)

Комплексный символ словоформы может содержать признаки двух родов: внутренние, характеризующие внутреннюю структуру

словоформы [+X, +Y], и внешние, указывающие на признаки окружающих слов (с прочерком на месте данного слова) [__X]. В табл. 11.3 приведен фрагмент лексикона в падежной грамматике.

Пользуясь базовым компонентом грамматики, можно вывести лексически заполненные предложения, типа приведенного на рис. 11.9.

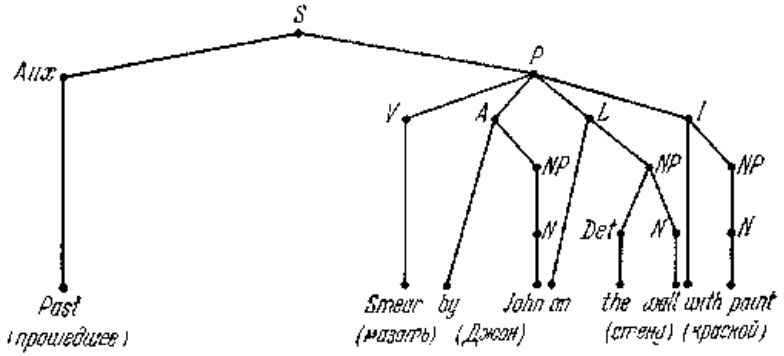


Рис. 11.9. Претерминальная цепочка падежной грамматики с лексическим заполнением.

К окончательному виду подобные предложения приводятся применением *общих и частных трансформационных правил.* Общие правила определяют выбор глубинных падежей на роль поверхностного субъекта и способы субъектного оформления слова. Частные правила определяют особенности состава и поверхностного оформления глубинных падежей у индивидуальных глаголов, если эти особенности не соответствуют типовым словарным предписаниям или общим трансформационным правилам.

Общие основания интерпретирующей семантики (ИС) были подвергнуты критике со стороны представителей *порождающей семантики* (ПС). Следует однако иметь ввиду, что и ИС, и ПС опираются на общую теорию трансформационных порождающих грамматик. Однако в ПС роль активного творческого элемента в процессе порождения высказывания принадлежит не синтаксису, а семантике. Соответственно этому, базовая компонента в модели порождающей семантики представляет собой правила образования семантических представлений предложений. Затем эти представления преобразуются с помощью трансформационных правил в поверхностные структуры. Модель ПС кроме того содержит также лексикон, заменяющий комбинации элементарных семантических смыслов лексемами. Схема лингвистической теории ПС представлена на рис. 11.10.

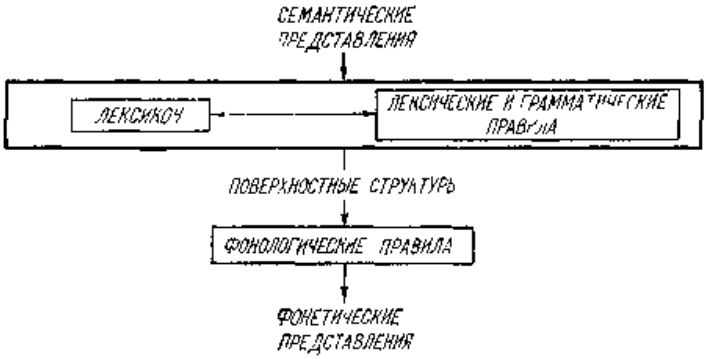


Рис. 11.10. Схема лингвистической теории в порождающей семантике

Следует отметить, что основные расхождения ПС и ИС касаются непризнания ПС уровня глубинных структур.

ПС не представляет собой такой целостной концепции, как ИС, и поэтому, несмотря на свои несомненные преимущества перед ИС, не нашла пока широкого использования в практических разработках.

Следует отметить, что при практической реализации указанные различия между ПС и ИС сглаживаются, и с точки зрения практики нам кажется целесообразно говорить о **двух уровнях языка: уровне глубинных структур в широком смысле и уровне поверхностных структур.** На первом уровне структуры характеризуются чисто семантическими свойствами, а на втором семантика осложнена синтаксическими категориями и преобразованиями. Первый уровень не зависит от конкретного языка, а второй воплощается в конкретном естественном языке. Что же касается необходимости введения промежуточного уровня между глубинным и поверхностным, то он может использоваться просто как удобный технический прием, облегчающий переход от поверхностного уровня к глубинному и обратно.

На этом мы закончим рассмотрение лингвистических аспектов, используемых при построении вопросно-ответных систем (ВОС) и перейдем к вопросно-ответным системам фреймов (ВОСФ).

11.3. Классификация вопросно-ответных систем фреймов, понимающих естественный язык

Вопросно-ответные системы фреймов, понимающие естественный язык, по способам представления и использования знаний можно условно разбить на четыре типа:

1) системы, использующие форматы частного вида;
2) системы, основанные на запоминании текста;
3) системы с ограниченной логикой;
4) системы с общим выводом.

11.3.1. Системы, использующие форматы частного вида

К этому классу относятся наиболее ранние программы формирования фреймов. Они обычно используют два частных формата — один для представления знаний о фреймах, хранимых в системе, а другой для представления предложений (фреймов) в языке. Такие программы исходят из предположения, что **необходимой (требуемой) информацией** в предложении является только та, которая соответствует их **частным форматам**. Хотя программы данного типа могут иметь усложненные механизмы для использования этой информации, они создаются для частных целей и не обладают в управлении информацией гибкостью, которая бы могла позволить использовать программы для других целей.

11.3.2. Системы, основанные на запоминании текста

Стремление преодолеть ограничения рассмотренных выше программ привело к созданию систем, использующих текст в естественном языке во всем его разнообразии и общности как основу для представления знаний о фреймах в системе. Запомненный текст (суперфрейм) снабжается различного рода схемами индексирования, предназначенными для упрощения поиска запрашиваемых предложений (фреймов). Задача системы состоит в выдаче одного или нескольких предложений из знаний системы, имеющих отношение к запросу. Существуют разнообразные методы для поиска уместных предложений и выбора тех, которые наиболее удовлетворяют запросу.

Данный подход имеет ряд трудно разрешимых проблем. К ним относятся:

1) невозможность получения ответа на любой вопрос, который требует некоторых выводов из более чем одной части запомненной информации;

2) качество ответов в большей степени зависит от формы, в которой текст и вопрос определены в естественном языке, чем от смысла текста и запроса.

11.3.3. Системы с ограниченной логикой

Системы данного вида разрабатываются с целью устранения недостатков систем, основанных на запоминании текста в естественном языке. В первую очередь в этих системах в качестве базовых знаний вместо предложений в естественном языке используется более формальная нотация, преследующая цель представить семантические отношения между данными. Раз знания записаны в этом виде, система должна уметь переводить входные предложения в естественном языке в формат внутреннего представления, т. е. выполнять семантическую интерпретацию.

Основным недостатком ранних систем с ограниченной логикой был тот, что сложная информация выражалась в них в форме программ и для введения новых объектов (фреймов) требовалась разработка новых программ и их связь со старыми программами. В свою очередь каждое изменение в программе могло привести (и на практике обычно приводило) к изменению других программ. В результате система росла, теряла стройность, обозримость, и экспериментировать с такой системой становилось практически невозможно. Выход из указанных затруднений был найден за счет разработки новой техники программирования, способной использовать **процедуральную информацию**, но в то же самое время выражающую эту информацию способом, не зависящим от специфики программы или темы диалога.

Типичным примером систем подобного рода, называемых *системами с процедуральной дедукцией,* является система Винограда. Система выполнена в языках PLANNER, PROGRAMMER (модификация LISP) и LISP. Система отвечает на вопросы, выполняет команды и принимает информацию в процессе ведения диалога на английском языке. Система состоит из разборщика грамматики английского языка, выполненного в PROGRAMMERS, программ семантического анализа, выполненных в LISP, и общей системы принятия решений, выполненной в PLANNER'е. Система включает в себя также в виде теорем, записанных в языке PLANNER, детальную модель простого

мира игрушек и упрощенную модель ее собственной разумности. Факты о текущей сцене представлены в виде утверждений PLANNER'а.

Система организует знания как набор программ, называемых «специалистами». Каждая из этих программ несет в себе частные знания системы о структуре языка и окружающем мире. Так, например, существуют синтаксические специалисты, которые анализируют различные типы фраз. Кроме того, существуют семантические специалисты, выполняющие различные функции. Они выбирают значения индивидуальных слов в зависимости от контекста, определяют, какие комбинации слов имеют смысл, определяют виды ссылок. Семантические специалисты могут вызывать знания о мире, о ситуации, о беседе.

Каждый из специалистов может потенциально использовать любую информацию, собранную о предложении, контексте или ситуации реального мира другими специалистами. Эта гибкость необходима для управления английским языком, но она создает самостоятельную проблему, так как часный специалист интересуется частными аспектами процесса понимания языка и имеет специальную организацию информации. Для решения этой проблемы в системе используются развитые описывающие структуры и явное описание всех свойств текущей структуры.

11.3.4. Системы с общим выводом

Попытки увеличить дедуктивную мощность систем с ограниченным выводом привели к идее **выражать знания в некоторой математической нотации (например, в исчислении предикатов) и затем использовать результаты в области математической логики по формальному доказательству теорем. В таких системах вопрос, заданный в естественном языке, представляется в виде формулы исчисления предикатов и трактуется как теорема, которая должна быть доказана**.

Стимулом к развитию таких систем послужил разработанный Робинсоном **принцип резолюции**, являющийся полной универсальной процедурой доказательства для исчисления предикатов первого порядка. **На основании работы Робинсона стало возможным создание эффективных автоматических программ, доказывающих теоремы и обладающих двумя важными свойствами. Первое свойство состоит в том, что процедура доказательства в ней универсальна, т. е. не учитывает специфику данной области. Область задается множеством утверждений (аксиом). Второе**

свойство состоит в том, что если доказательство возможно с использованием правил исчисления предикатов, то процедура гарантирует, что оно будет найдено, хотя, возможно, за очень длительное время.

Эти свойства удобны, но плата за их использование велика: снижение эффективности системы.

Подход к вопросно-ответным системам с использованием универсальной процедуры доказательств был развит в ряде разработок. В этих системах **информация представляется единым образом**, не зависящим от особенностей частной программы. Это дает возможность пользователю описывать знания в нейтральном виде, не приспосабливая их к особенностям и частностям вопросно-ответной системы, и гарантирует то, что система будет применима к любой области, представимой в исчислении предикатов.

Использование исчисления предикатов имеет и серьезные недостатки. Представление сложной информации в нейтральной форме игнорирует важный вопрос, как представлять знания. Человек не хранит в голове точно определенное множество логических аксиом, из которых он выводит знания с помощью процедуры доказательства. Скорее у него есть масса эвристик и процедур различной степени общности для решения и представления различных задач. Игнорирование этого факта приводит к неэффективности при представлении в исчислении предикатов объектов реальной сложности. К недостаткам использования исчисления предикатов относится и то, что определенные свойства естественного языка плохо представимы в двузначной логике. Устранить указанные недостатки можно, используя **размытые множества и модальные логики**. Однако для этих логик в настоящее время не известны эффективные разрешающие процедуры.

11.4. Синтаксический анализ

Задача этапа синтаксического анализа (СА) предложения, записанного в некотором языке, заключается в построении структуры предложения, определяемой грамматикой этого языка. Так, например, при записи предложения в языке, определяемом КС-грамматикой, задача СА заключается в получении дерева составляющих (дерева зависимостей) исходного предложения При обработке языка, описываемого трансформационной грамматикой, задачей СА уже будет не получение **поверхностной структуры предложения, а построение его глубинной структуры**. В связи с тем, что для **удобства и компактности описания естественных языков используются**

трансформационные грамматики или их модификации, мы будем считать задачей СА получение глубинной структуры предложения, выражаемой, например, и **виде дерева составляющих или в виде формул исчисления предикатов**. Однако, для введения в историю вопроса и определения используемой в дальнейшем терминологии, мы приведем краткую характеристику ранних анализаторов.

11.4.1. Синтаксические анализаторы КС-языков

Когда создавались первые *синтаксические анализаторы (разборщики)* предложений естественного языка, не существовало теории синтаксиса, приемлемой для использования на машинах. Анализаторы представляли набор подпрорамм, который постепенно развивался по мере того, как грамматика разширялась и управляла все более сложными предложениями. Анализаторы имели те же недостатки, что и любые программы, конструируемые таким образом. В результате анализаторы становились все сложнее и все труднее становилось понимать их внутренние взаимосвязи. Неудачные попытки в области раннего машинного перевода ясно показали, что задача обработки естественного языка без лучшего понимания основ лингвистики и математических свойств грамматики является преждевременной.

В последующих исследованиях по обработке естественного языка на ЭВМ можно выделить **два подхода.**

В первом подходе игнорируется синтаксис в его традиционном понимании, и для получения **информации о предложении используется процесс сопоставления по образцу**. Системы этого типа не предпринимали попытки полного синтаксического анализа входного предложения. Они или ограничивали входной язык небольшим множеством фиксированных форматов, или ограничивали понимание предложений в результате игнорирования синтаксиса.

Во втором подходе берется упрощенное подмножество естественного языка, которое может быть описано в хорошо изученной формальной грамматике, например, такой, как некоторая вариация КС-грамматики.

Не останавливаясь детально на существе ранних алгоритмов разбора, воспринимающих КС-грамматики, отметим, что они разделяются по **двум основным параметрам**:

1) направлению разбора «сверху-вниз» или «снизу-вверх»;

2) последовательному или параллельному методу генерации деревьев разбора.

Деление алгоритмов по направлению разбора основывается на делении **грамматик** на **порождающие и распознающие. Алгоритмы «сверху-вниз» теоретически основываются на идее использования**

порождающей грамматики при генерации всех возможных предложений языка, пока не будет порождено предложение, соответствующее входному. Такой подход при прямолинейном использовании требует слишком много времени, но существуют способы, повышающие эффективность этого метода.

Алгоритмы «снизу-вверх» основываются на распознающих грамматиках. Алгоритмы пытаются объединить различным образом элементы входной строки до тех пор, пока не будет найдено дерево, покрывающее всю входную строку.

Алгоритмы разбора делятся на последовательные или параллельные в зависимости от того, строят они одно дерево разбора (если оно не соответствует входному предложению, то строится новое дерево, до тех пор пока не будет получен правильный разбор) или одновременно все возможные деревья разбора.

Последовательные алгоритмы являются медленными, но требуют мало памяти. К числу наиболее интересных параллельных алгоритмов относится алгоритм Эрли, который рассматривает одновременно (параллельно) все возможные анализы и получает все варианты разбора входной строки за время, пропорциональное кубу длины входной строки. Коэффициент пропорциональности определяется рассматриваемой КС-грамматикой и не зависит от вида строки. Для грамматик специального вида данный алгоритм автоматически достигает времени разбора, пропорционального квадрату длины (или длине) входной строки.

Как было указано ранее, контекстно-свободные грамматики не могут описать естественный язык в полном объеме.

11.4.2. Анализаторы языков, описываемых трансформационными грамматиками

Все более поздние попытки по расширению мощности грамматики, описывающей естественный язык, так или иначе связывались с **трансформационными грамматиками**. Из определения трансформационной грамматики следует, что **она ориентирована на генерацию предложений (синтез), а не на распознавание (анализ).** Хотя существует алгоритм, использующий трансформационную грамматику для анализа предложений, он слишком неэффективен, и вопрос о его практическом использовании даже не встает. Алгоритм работает по принципу «анализ через синтез», т. е. генерирует все возможные предложения и ищет среди них анализируемое.

Две попытки разработать более практичные алгоритмы трансформационного распознавания привели к созданию алгоритмов,

или требующих больших затрат времени, или теряющих свойство полноты. Оба алгоритма пытались анализировать предложения, применяя трансформации в обратном порядке.

11.4.3. Анализ естественных языков, описываемых расширенными сетями переходов

Далее были разработаны системы, воспринимающие входной язык, близкий к естественному (английскому). Эти системы **используют понятие *расширенной сети переходов* (augmented transition network) и основываются на трансформационной грамматике.** Их задача выработать глубинную структуру предложения, анализируя его поверхностную структуру.

Введем понятия расширенной и *рекурсивной сети переходов,* являющиеся основными в перечисленных выше анализаторах и используемые ими для задания грамматики.

Рекурсивная сеть переходов (РСП) есть направленный граф с помеченными вершинами (состояниями) и дугами. Выделяется **начальное состояние и множество конечных состояний** (рис. 11.11).

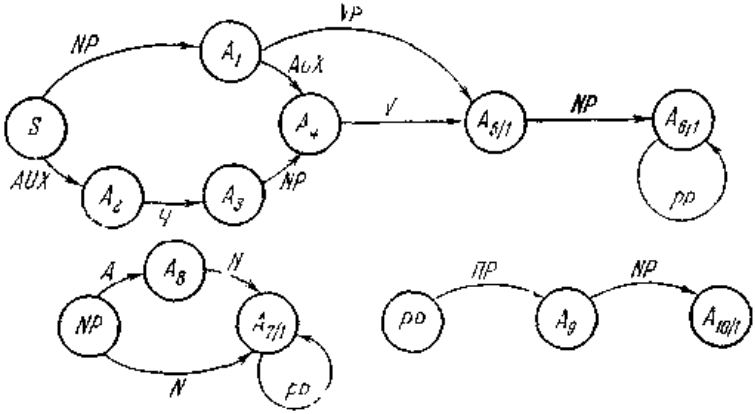


Рис. 11.11. Фрагмент грамматики, представленный в виде рекурсивной сети переходов. S — начальное состояние; A_5, A_6, A_7, A_{10}— конечные состояния; AUX — вспомогательный глагол; Ч — частица; PP — предложная группа существительного; ПР — предлог; AUX, Ч,V, A, N, ПР лексические (терминальные) категории.

Метки на дугах могут быть как терминальными символами, так и наименованиями состояний.

Дуга, помеченная наименованием состояния, интерпретируется следующим образом:

1) наименование состояния на конце дуги запоминается в стеке;

2) управление передается к состоянию, которым помечена дуга (без изменения обрабатываемого символа входной строки).

Будем в дальнейшем называть операции, перечисленные в пунктах 1 и 2, *погружением.*

Если текущее состояние является конечным, то происходит выталкивание последнего запомненного в стеке состояния и передача управления в это состояние. Критерием приемлемости входной строки является попытка вытолкнуть пустой стек, когда последний символ входного предложения обработан. Наименования состояний, появляющихся на дугах в этой модели, соответствуют наименованиям нетерминальных конструкций, которые могут быть обнаружены во входных предложениях.

На рис. 11.11 приведен пример рекурсивной сети переходов для подмножества русского языка. Конечные состояния имеют вид $A_{i/1}$. Приведенная РСП принимает, например, такие предложения, как: «Большой Джон будет строить красивый дом у белой реки», «Будет ли Михаил строить кооператив в Москве?» и т. п.

Нетрудно установить, что **РСП эквивалентна недетерминированному автомату с магазинной памятью и имеет <u>мощность контекстно-свободной грамматики</u>.** Как было уже указано, КС-грамматика не является адекватным механизмом для описания естественного языка. Увеличим возможности РСП, определив понятие **расширенной сети переходов** (РАСП), которая способна выполнять трансформационные преобразования, не вводя понятия **трансформационной компоненты**. Основное свойство, которое трансформационная грамматика добавляет к КС-грамматике — это способность передвигать, копировать и устранять фрагменты синтаксической структуры (так, что их положение в глубинной структуре отличается от положения в поверхностной структуре) и совершать эти действия в зависимости от контекста. Мы можем ввести эквивалентные свойства в РСП, добавив к каждой дуге:

1) произвольные *условия* (переход в состояние, указываемое дугой, разрешается только при выполнении этих условий);

2) произвольные *действия* (действия выполняются, если осуществляется переход к состоянию, на которое указывает дуга).

Будем называть это расширение РСП *расширенной сетью переходов* (РАСП).

РАСП строит частичное структурное описание предложения при переходе из одного состояния сети в другое. Куски этого частичного

описания хранятся в **регистрах (со стековой структурой)**, и их содержимое автоматически проталкивается (погружается), когда рекурсивное применение вызывает переход к более нижнему уровню. *Действия,* связанные с дугами, изменяют содержимое этих регистров в терминах их предыдущего содержимого, содержимого других регистров, текущих входных символов и (или) результатов нижних уровней вычисления. Кроме того, регистры могут использоваться для хранения указателей (флагов), отражающих особенности прохождения через сеть. Указатели могут опрашиваться *условиями,* связанными с дугами.

Каждое конечное состояние связывается одним или несколькими условиями, при выполнении которых осуществляется переход на предыдущий уровень. Каждому из этих условий ставится в соответствие функция, результат вычисления которой возвращается как итог при переходе на верхний уровень.

Чтобы рассмотрение РАСП было более конкретным, приведем в табл. 11.4 формальное определение языка, в котором РАСП может быть представлена.

Таблица 11.4

Формальное определение языка для описания расширенных сетей переходов

```
<сеть переходов> → (<множество дуг> <множество дуг> *)
<множество дуг> → (<состояние> <дуга> *)
<дуга> → (КАТ <имя категории> <условие> <действие> * <переход>) |
         (ПОГР <состояние> <условие> <действие> * <переход>) |
         (УСЛ<произвольн.метка><условие><действие>*<переход>)|
         (ВЫТ <форма> <условие>)
<действие> → (ПРТЕК <регистр> <форма>) |
             (ПРПОГР <регистр> <форма>) |
             (ПРВЫТ <регистр> <форма>)
<переход> → (ПУ <состояние>) |
            (ПП <состояние>)
<форма> → (ЗНАЧ <регистр>) |
          Δ |
          (СВОЙСТВО <свойство>) |
          (ДЕР <фрагмент> <регистр> *) |
          (СПИСОК <форма> *) |
          (ОБЪЕДИН <форма> <форма>) |
          (ИМЯ <произвольная структура>)
```

Определение дадим в форме расширенной бэкусовской нотации. Будем обозначать вертикальной чертой | альтернативные варианты, операцию * будем использовать как индекс, присутствие которого указывает на возможность вхождения конституента произвольное число раз. Нетерминальные символы грамматики будем представлять в виде текста из строчных букв, заключенного в угловые скобки, а терминальные (исключая *, Δ, |) заглавными буквами, не заключенными в скобки. Символом Δ будем обозначать наименование регистра, содержащего текущее слово входного предложения.

Дадим пояснения к определениям. Первая строка в таблице указывает, что сеть переходов есть множество дуг (или несколько таких множеств), заключенное в круглые скобки. Множество дуг состоит из наименования состояния, за которым следует любое число дуг. Мы будем предполагать, что РАСП представляется в виде списочной структуры. Тогда сеть переходов есть список, элементы которого — множество дуг. Множество дуг в свою очередь есть список, элементами которого являются наименование состояния и дуги.

Дуги представляются в виде <u>списков четырех разновидностей</u>. Условия и действия, связанные с конечными состояниями, представляются как псевдодуги. Первый элемент списка (дуга) указывает тип дуги (КАТ — задает категорию терминального наименования, которым помечена дуга, ПОГР — указывает, что данный тип дуги вызывает операцию погружения, УСЛ — задает условие, ВЫТ — указывает на необходимость выполнения операции выталкивания стека). Третий элемент определяет произвольное условие, при выполнении которого осуществляется переход к состоянию, указанному в пятом элементе. Четвертый элемент определяет действие, осуществляемое при переходе к состоянию, на которое указывает дуга.

Опишем более подробно типы дуг. Дуга типа КАТ — дуга, которой следуют, если текущее входное слово является лексической (терминальной) категорией, указанной во втором элементе дуги.

Дуга типа ПОГР используется для обработки дуг, помеченных наименованием состояния (что вызывает операцию погружения)

Дуга типа УСЛ (условие) используется для задания произвольного условия, определяющего возможность следования к состоянию, указываемому дугой.

Дуга типа ВЫТ (выталкивание) является пустой дугой и предназначена для того, чтобы иметь возможность связать с конечными состояниями определенные условия (выполнение которых необходимо для подъема на более верхний уровень) и действия (возвращаемые на верхний

уровень). Представления условий и действий в виде информации, связанной с пустой дугой, дает возможность упорядочить выбор ВЫТ по отношению к другим дугам данного состояния.

Состояние, к которому осуществляется переход, указывается в пятом элементе дуги. Переходы бывают двух видов:

1) с выбором в качестве текущего обрабатываемого символа очередного символа входной строки (предложения);

2) без изменения текущего обрабатываемого символа. По аналогии с программированием первый переход будем называть передачей управления (ПУ), а второй—переходом к подпрограмме (ПП).

Действия и *формы,* встречающиеся в сети, представляются в виде «польской нотации», нотации, в которой функция представляется как список, заключенный в скобки, первым элементом его является наименование функции, а остальными элементами — аргументы функции.

Мы определим **три типа *действия***, которые присваивают значение *формы* наименованию регистра. Действия различаются по тому, на каком уровне они выполняют присваивание (ПРТЕК—присваивание на текущем уровне, ПРПОГР — присваивание на уровне погружения, т. е. уровне более глубоком, чем текущий, ПРВЫТ — присваивание на уровне выталкивания).

Формы, так же как и условия, можно записывать в виде функций языка программирования (например, LISP). Приведенные в табл. 11.4 типы форм составляют базовое множество, достаточное для того, чтобы проиллюстрировать основные свойства РАСП.

ЗНАЧ есть функция, чьим значением является содержимое указанное в форме регистра. Значением формы Δ является текущее слово входного предложения. В случае ПОГР значением формы Δ является уровень, получаемый при возврате из ПОГР.

СВОЙСТВО есть функция, которая определяет значение свойства, указанного в форме для текущего слова входного предложения.

Форма ДЕР (дерево) используется для построения фрагмента или полного дерева (структуры) разбора. Фрагмент задается использованием скобок, наименований символов и указателей параметров, изображаемых знаком + (см. ниже пример). Значением формы ДЕР является фрагмент дерева, в которое на место параметров подставлены значения регистров, указанных в форме. Параметры ставятся в соответствие регистрам следующим образом: содержимое первого регистра замещает первый символ +, содержимое второго регистра — второй символ и т. д. Кроме того, форма Δ (если она присутствует в ДЕР) заменяется на соответствующее ей значение. Три

оставшиеся формы предназначены также для построения структур. Форма СПИСОК — создает список из значений аргументов, ОБЪЕДИНИТЬ — объединяет два списка в один, ИМЯ — производит как значение (невычисленные) аргументы формы. Отметим, что три последних формы являются встроенными функциями языка LISP (LIST, APPEND, QUOTE).

При выполнении анализатора для конкретной грамматики для создания более гибкой системы (чем описанная) могут быть введены дополнительные форматы дуг, увеличена гибкость функции ДЕР и т. п. Вообще целесообразно **множество условий, действий и форм оставить открытым для расширений в ходе эксперимента**. Заметим, что формат дуг и действий вместе с произвольными выражениями LISP для изображения условий и форм обеспечивает модель, эквивалентную по мощности машине Тьюринга и, следовательно, полную в теоретическом смысле.

В табл. 11.5 приведен пример записи во введенном формализме фрагмента грамматики, введенной на рис. 11.11.

Таблица 11.5

Фрагмент грамматики, представленной в расширенной сети переходов

```
((S (ПОГР NP | Усл
      (ПРТЕК S Δ)
      (ПРТЕК ТИП (ИМЯ ПОВЕСТВ))
      (ПУА1))
      (КАТ AUX Усл
      (ПРТЕК AUX Δ)
      (ПРТЕК ТИП (ИМЯ ВОПР))
      (ПУА2)))
(A1   (ПОГР VP | Усл
      (ПРТЕК AUX ø)
      (ПРТЕК VP Δ)
      (ПУА5))
      (КАТ AUX Усл
      (ПРТЕК AUX Δ)
      (ПУА4)))
(A2   (КАТ Ч Усл
      (ПУА3)))
(A3   (ПОГР NP | Усл
      (ПРТЕК S Δ)
      (ПУА4)))
(A4   (КАТ V | Усл
      (ПРТЕК V Δ)
      (ПУА5)))
(A5   (ВЫТ (ДЕР (S+++| (VP+)) тип, S, AUX, Δ) Усл)
      (ПОГР NP | Усл
      (ПРТЕК VP (ДЕР (VP (V+)Δ) V))
      (ПУА6)))
(A6   (ВЫТ (ДЕР (S++++), тип, S, AUX, VP) Усл)
      (ПОГР PP | Усл
      (ПРТЕК VP (ОБЪЕДИН (ЗНАЧ VP (Список Δ)))
      (ПУА6))))
```

Рассмотрим действие РАСП, введенной в табл 11.5 для предложения «Будет ли Джон строить дом?»

В качестве пояснения приведем содержимое регистров в соответствующих состояниях:

S: AUX:=будет;

ТИП: = ВОПР;

A2:

A3: *S:=(NP* Джон);

A4: *V* =строить;

A5: *VP:=(VP{V* строить)(*NP* дом));

A6: (S ВОПР(*NP* Джон) будет *(VP(V* строить) *(NP* дом))).

Полученную в состоянии A6 списочную структуру, соответствующую глубинному представлению исходного предложения, изобразим на рис. 11.12.

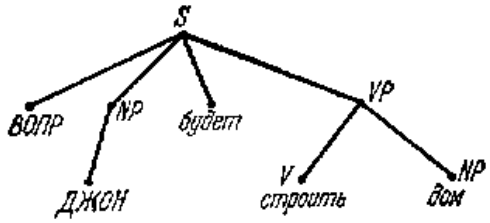


Рис. 11.12. Глубинная структура предложения.

В анализаторах НС-грамматик полученное структурное описание предложения соответствует алгоритму разбора, т. е. построение структуры происходит одновременно с применением грамматических правил. В РАСП эти процессы разделены. Это дает возможность некоторым конституентам, найденным в ходе разбора, появляться в окончательной структуре несколько раз или ни разу, и их место может отличаться от места в поверхностной структуре. Кроме того, структурное описание, присвоенное некоторой конституенте в ходе разбора, может быть впоследствии изменено. Эти свойства плюс способность вставлять проверку произвольных «условий» позволяет строить глубинную структуру в то время, когда анализатор выполняет переходы, соответствующие поверхностной структуре предложения.

Перечислим основные свойства РАСП, которые делают их привлекательными для использования в качестве модели естественного языка.

1. Наглядность. НС-грамматики являлись очень популярной моделью грамматики, несмотря на их неадекватность для управления некоторыми свойствами естественного языка, во многом благодаря их наглядности. Теория трансформационных грамматик описывает практически все свойства естественного языка, но в том виде, в котором она существует, эта теория потеряла наглядность НС-грамматик. В трансформационной грамматике (ТГ) эффект отдельного правила связан с его взаимодействием с другими правилами, и требуется сложнейший анализ для определения эффекта и целей данного правила. РАСП, обеспечивая мощность ТГ, во многом сохраняет наглядность НС-грамматик.

2. Генеративная мощность. Как уже указано выше, РАСП имеет мощность машины Тьюринга, и при этом операции, выполняемые в РАСП, являются «естественными» для анализа языка. РАСП в отличие от ТГ (ориентированной на генерацию) с одинаковым успехом может использоваться как для генерации (формирования ответа), так и для распознавания (синтаксического анализа) предложений.

3. Эффективность представления. РАСП в отличие от НС-грамматик имеет средства для явного указания общих частей многих правил, что дает более эффективное представление. Объединение общих частей позволяет не только более компактно представлять грамматику, но и устраняет излишнюю обработку при разборе предложения (за счет уменьшения количества сопоставлений при определении применимости правил в течение разбора).

Отметим, что в РАСП возможно объединение подобных частей.

Пусть требуется представить в виде сети следующие подобные правила: *S→ABCDK, S→PBCTK*. На рис. 11.13 приведен пример объединения подобных частей этих правил.

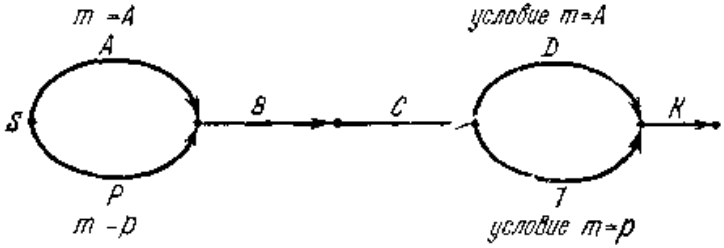


Рис. 11.13. Объединение подобных частей в расширенной сети переходов.

Для обоих правил строится единая сеть. Правила различаются благодаря действиям на дугах *A* и *P* и условиям на дугах *D* и *T*. Процесс объединения подобных частей делает грамматику более компактной, но увеличивает время разбора (за счет внесения дополнительных действий и условий).

4. Эффективность анализа. Как уже указывалось, эффективность вытекает из объединения общих частей и способности изменять построение фрагмента дерева (без повторного просмотра входного предложения) или откладывать его построение.

5. Гибкость экспериментирования. Гибкость достигается за счет того, что множество операций оставлено открытыми и может пополняться в ходе экспериментирования. Следует отметить, что явное построение структуры предложения (с помощью действий) позволяет использовать РАСП не только для построения глубинных структур предложения, но и для других видов представления таких, как грамматики зависимостей, падежные грамматики.

Рассмотрим алгоритмы разбора, которые можно использовать при представлении грамматики в виде РАСП. Как мы уже указывали, **РСП эквивалентна автомату с магазинной памятью**. Поэтому многие

существующие алгоритмы для НС-грамматик могут быть непосредственно сведены к РСП. Кроме того, один из наиболее эффективных алгоритмов для КС-грамматик (п.11.4 1) может быть приспособлен с небольшими модификациями для использования в РСП.

Использование алгоритма Эрли для РАСП (а не для РСП) является более сложным делом. Действительно, для переходов, зависящих от содержимого регистров, трудно определить эквивалентные конфигурации и объединить их при дальнейшей обработке. Кроме того, использование регистров и действий при построении структур усложняет задачу выбора представления для объединенных конфигураций. Поэтому прямолинейно расширить алгоритм Эрли не удается. Однако, если в РАСП делать явные различия между **<u>флаговыми регистрами</u> (т. е. регистрами, содержащими только условия, выбранные из конечного словаря)** и **<u>регистрами содержания</u> (содержащими произвольные структуры)**, и ограничить условия и действия, приписанные дугам, таким образом, что:

1) действия могут ссылаться только на флаговые регистры и символы во входной строке;

2) на действия и условия наложить ограничения по времени обработки,

3) существует только один регистр содержания, изменяемый функцией ДЕР или конечным состоянием (содержимое его не анализируется), тогда можно построить версию алгоритма Эрли с указанными ранее временными характеристиками. Однако если мы ослабим перечисленные условия, то временная граница превзойдет (Kn^3) (например, одно условие и действие на дуге может потребовать больше, чем n^3 шагов).

Для многих применений при анализе фреймов нет необходимости получать представления всех возможных разборов входного предложения. В частности, в нашем случае более важно выбрать «наиболее вероятный» разбор в данном контексте, но сделать это наиболее быстрым образом. В таких применениях последовательный подход (с подходящим механизмом для выбора анализа, рассматриваемого первым) более предпочтителен, чем параллельный, так как он в большинстве случаев позволяет избежать необходимости следовать всем альтернативам. Повторный анализ тех же самых подстрок входной строки при альтернативных разборах можно устранить путем запоминания предыдущих результатов.

Успех последовательного подхода во многом зависит от качества механизма, осуществляющего выбор «наиболее вероятного» разбора. В РАСП некоторые из таких механизмов могут быть добавлены

естественным образом. Упорядочением выбора дуг, исходящих из данного состояния, можно навязывать соответствующий порядок анализа. При создании грамматики можно подбором упорядочения дуг пытаться выделить наиболее вероятные ситуации. Кроме того, повторением в РАСП одной и той же дуги в виде нескольких дуг с разными условиями возможно сделать упорядочение этих дуг зависящим от особенностей обрабатываемого предложения, в частности, сделать зависящим от семантических свойств слов, встречающихся в предложении.

Заканчивая рассмотрение методов синтаксического анализа, укажем, что **алгоритмы разбора** могут рассматриваться как **процедуры поиска на графе**. Процедура поиска начинается с пустого дерева разбора. Каждый узел дерева поиска соответствует частному разбору предложения, а дуга соответствует добавлению нового уровня к дереву разбора. Различные методы синтаксического анализа отличаются способом, которым они добавляют новые дуги к дереву разбора. Алгоритмы «сверху—вниз» добавляют дуги на вершину дерева разбора и работают в направлении «вниз», в то время как процедуры «снизу — вверх» добавляют дуги в основание дерева и работают в направлении «вверх». Последовательные алгоритмы разбора соответствуют поиску «сначала в глубину», а параллельные — поиску «сначала в ширину». Однако хорошо известно, что **большей эффективностью обладают стратегии, управляемые эвристиками**.

Приведем примеры подобных эвристик:

1) избегать выполнения одного и того же частного разбора (т. е. разбора подстроки входного предложения) более одного раза;

2) избегать определенных разборов, которые не могут привести к успеху (определяются просмотром вперед, разделением предложения на части и т. п.);

3) устранять семантически нелепые анализы (проверкой части анализа на соответствие базовым данным, обычно при анализе «снизу—вверх»);

4) использовать оценочные функции и другие механизмы управления поиском наиболее перспективного пути.

11.5. Семантическая интерпретация

11.5.1. Общие сведения о семантической интерпретации

Входом для этапа семантической интерпретации (СИ) является глубинная структура обрабатываемого предложения, точнее

структуры, так как **на этапе синтаксического анализа не удается (в общем случае) получить однозначную интерпретацию предложения**. **Задачей СИ явтяется понять входное предложение**. **Под словом «понять» мы подразумеваем не только выражение смысла предложения в известных нам понятиях, но и установление связи данного предложения с известными нам фактами**. Поэтому мы определяем задачи этапа семантической интерпретации следующим образом:

— получить **однозначное представление входного предложения** (интерпретировать его) в терминах базовых данных;

— **объединить входное предложение** с базовыми данными.

Этап СИ может выполняться как параллельно с процессом синтаксического анализа, так и после его завершения. В первом случае подпрограммы интерпретации добавляются к некоторым или всем правилам грамматики и выполняются во время применения грамматического правила при синтаксическом анализе. Достоинством этого подхода является то, что результат семантической интерпретации может направлять синтаксический анализ.

Второй подход имеет то преимущество, что глубинная структура, полученная на уровне синтаксического анализа, может просматриваться многократно и в произвольном порядке, что облегчает задачу СИ.

Следует отметить, что еще **не существует общей теории процесса семантической интерпретации**. Процессы СИ существенно зависят от ряда факторов и в первую очередь от внутреннего представления базовых данных, определяющего общность или ограниченность вывода в системе (см. п. 11.3.3 и п. 11.3.4).

В данном параграфе мы сначала опишем общую постановку задачи СИ, а затем рассмотрим ее конкретные реализации отдельно для **представления в виде семантических сетей (системы с ограниченной логикой) и в виде формул исчислении предикатов (системы с общим выводом).**

В связи с распространенностью в естественных языках явления многозначности слов, словосочетаний, предложений, раскрытие многозначности при обработке текстов на ЭВМ приобретает особое значение. Раскрытие многозначности некоторых предложений невозможно без ссылок на контекст или без обращения к общим знаниям о мире. Например, предложение «Он пошел в парк с девушками» может быть понято как:

1) Он и девушки пошли в парк, или 2) Он пошел в парк, где есть девушки. Вероятно, двузначность данного предложения может быть раскрыта ссылкой на контекст. Однако только общие знания о мире

могут раскрыть двузначность следующих выражений: «Гастроли балета на льду» и «Гастроли балета на Кавказе».

Значительно большее количество предложений является многозначным для машины, чем для человека, так как ЭВМ имеет более простые алгоритмы для раскрытия многозначности и меньшие знания о контексте и мире. Например, приведенные выше предложения (о гастролях) будут двузначными для ЭВМ и скорее всего однозначными для всех людей. Существующие системы распознают только подмножество естественного языка, и это дает возможность раскрыть некоторые из многозначностей. **Оставшиеся многозначности устраняются обычно одним из двух способов: 1) *методом семантических категорий;* 2) *ссылками на базовые данные.***

Семантические категории расширяют традиционную грамматическую классификацию слов введением подклассов. Отсутствие подобных подклассов приводит к катастрофическому возрастанию количества интерпретаций. Например, **если слова A, B, C и D имеют по три значения, то предложение, содержащее все эти слова, может иметь 3×3×3×3=81 интерпретацию. Человек не строит все интерпретации, он выделяет наиболее разумные.** Известно, что слово «коса» может обозначать следующие объекты: сельскохозяйственный инструмент, выступающий мыс, длинные волосы, уложенные определенным образом. С другой стороны, «зеленый» может обозначать: цвет, неспелый, неопытный. Однако, когда человек видит фразу «зеленая коса» он не рассматривает 9 интерпретаций, так как он знает, что «зеленый» в смысле «неспелый» применяется только к овощам и фруктам, а в смысле «неопытный» только к людям. Таким образом, чтобы ввести эту информацию в ЭВМ, необходимо расширить традиционную грамматическую классификацию слов на **классы**. Кроме обычных классов: **существительное, прилагательное и т. д., вводят классы «одушевленный», «неодушевленный», «человеческий», «абстрактный», «физический» и т. п** Используя подобную информацию, удается выделить бессмысленные комбинации слов при выборе интерпретации.

Типичной системой, использующей такую информацию, является программа, разработанная Глазерфельдом. Анализ предложения осуществляется «снизу— вверх». В начале анализа каждому слову в предложении присваивается список «индексных категорий» из словаря (соответствуют описанным выше классам и подклассам). В течение анализа, каждой более высокой вершине дерева разбора назначается своя **«индексная категория»** на основании «индексных категорий»

составляющих данной вершины. Назначение категории осуществляется специальной процедурой — **«реклассификации»**.

Недостатком метода семантических категорий является тот факт, что с их помощью не могут быть раскрыты все многозначности. Например, они не могут раскрыть двузначность предложения «Он пошел в парк с девушками», так как для раскрытия двузначности необходима **контекстуальная информация**. Другим их недостатком является увеличение и усложнение словаря в связи с детальной классиффикацией слов. При **методе семантических категорий информация в словаре будет частично дублировать информацию в базовых данных системы**. Желание объединить эту информацию приводит к методу раскрытия многозначности путем ***ссылки на базовые данные***. Например, в приведенном предложении семантический интерпретатор можем сформулировать вопрос к базовым данным: «Идут ли к ним девушки»? Ответ на этот вопрос может определить выбор интерпретации. В общем случае **метод раскрытия многозначности заключается в том, что все имеющиеся интерпретации сопоставляются с базовыми данными и устраняются интерпретации, противоречащие базовым данным**. Однако и после этого может остаться более одной интерпретации. Это наиболее трудный случай.

Можно указать несколько подходов к решению этой задачи:

1) выбрать интерпретацию, которая соответствует части базовых данных с наибольшими связями между вершинами;

2) выбрать интерпретацию, при объединении которой с базовыми данными к ним будет добавляться минимальное количество новых вершин и дуг;

3) выбрать наиболее разумную интерпретацию, используя некоторую «вероятностную» меру «разумности»;

4) запомнить все интерпретации в базовых данных и пытаться устранять многозначность за счет поступления новой информации (в частности, запрашивая необходимую информацию).

Отметим, что при обработке естественного языка **процессы СИ, представления фактов, выполнения выводов и формирования ответа** на вопросы могут быть существенно упрощены, если идентичные предложения, выражаемые различными поверхностными структурами, будут представлены **единой концептуальной конструкцией**. Одной из причин разнообразия поверхностных структур при единстве смысла является разнообразие «поверхностных глаголов», описывающих одну и ту же ситуацию. Шенк исходит из существования «глубинных» (канонических) глаголов, унифицирующих в глубинных структурах смысл многих поверхностных глаголов.

До сих пор мы, рассматривая вопросы семантической интерпретации, по умолчанию предполагая, что на вход системы поступают отдельные предложения. Задача значительно усложняется, если требуется понять смысл связанного текста.

Перейдем к рассмотрению процессов СИ с учетом структуры представления базовых данных.

11.5.2. Семантическая интерпретация в системах с ограниченной логикой

При описании процесса СИ с ограниченной логикой мы будем основываться на реальной вопросно-ответной системе, воспринимающей естественный (английский) язык. Выбор реальной ВОС преследует цель показать состояние практики в решении одного из наиболее сложных этапов по обработке естественного языка— семантической интерпретации. Несмотря на то, что описываемая система воспринимает английский язык, мы там, где это возможно, будем для удобства читателей приводить примеры на русском языке. В случаях, где примеры отражают специфику английского языка, мы будем приводить английский текст, а в скобках давать его русский перевод.

В рассматриваемой системе входом для СИ является каноническая структура, полученная в результате обработки ограниченного естественного языка вариантом расширенной сети переходов (РАСП). Действия РАСП были подробно описаны в п. 11.4.3, поэтому мы сейчас остановимся только на механизме, позволяющем осуществлять перевод входного текста в каноническую структуру. Основой этой операции является включение в словарь определенной информации (табл. 11.6). В словаре каждый поверхностный глагол содержит основной вход, представляемый в виде L-«ГЛАГОЛ», где L— признак основного входа, а «ГЛАГОЛ» — конкретный глагол в инфинитиве. Список свойств поверхностного глагола содержит различную информацию: **ссылку на каноническую форму данного поверхностного глагола, различные формы глагола, А-правила и Ф-правила** (см. п.11.7). А-правила используются при анализе входного предложения и соотносят поверхностные (предложные и беспредложные — обозначаются БП) существительные, используемые с данным поверхностным глаголом, глубинным структурам (например, в смысле Филмора), связанным с соответствующим каноническим глаголом. А-правила представлены в виде **списка триплетов**; первый элемент в каждом триплете указывает предлог (или отсутствие его), с которым используется поверхностное существительное. Второй элемент

триплета есть список семантических категорий (о назначении семантических категорий см. п. 11.5.1), к одной из которых должно относиться поверхностное существительное для того, чтобы удовлетворять глубинному падежному отношению (deep case relation), задаваемому третьим элементом триплета. Поясним структуру А-правила на примере: JOHN BOUGHT THE CAR FROM MARY (ДЖОН КУПИЛ АВТОМОБИЛЬ У МЭРИ).

Как видно из табл. 11.6, поверхностному глаголу BUY (ПОКУПАТЬ) соответствует канонический глагол EXCHANGE (ОБМЕНИВАТЬ). Глагол ОБМЕНИВАТЬ определяется следующими глубинными отношениями (падежами): ПОКУПАТЕЛЬ (BUYER); ПРОДАВЕЦ (SELLER); ПОЛУЧЕННОЕ (THINGBT); ОТДАННОЕ (THINGGIVEN); МЕСТО (LOC); ВРЕМЯ (TIME).

Таблица 11.6

Фрагмент словаря

L-BUY (ПОКУПАТЬ) WORD CLASS (КЛАСС СЛОВА) CANNON-VB (КАНОН. ГЛ) INF (ИНФИНИТИВ) SG3 (ЕД. ЧИСЛО 3 ЛИЦО) PAST (ПРОШЕДШЕЕ) -EN -ING	 VERB (ГЛАГОЛ) EXCHANGE (ОБМЕНЯТЬ) BUY BUYS BOUGHT BOUGHT BUYING
P-RULES (А-Правила)	(ОК (HUMAN ORGANIZATION) BUYER) (БП (ЧЕЛОВЕК ОРГАНИЗАЦИЯ) ПОКУПАТЕЛЬ) (ОК (PHYSOBJ) THINGBT) (БП (ФИЗИЧ. ОБЪЕКТ) ПОЛУЧ.) (FROM (HUMAN ORGANIZATION) SELLER) (У (ЧЕЛОВЕЧ. ОРГАНИЗАЦИЯ) ПРОДАВЕЦ) (FOR (MONEY) THINGGIVEN) (ЗА (ДЕНЬГИ) ОТДАННОЕ) (AT (PLACE) LOC) (В (МЕСТО) МЕСТО) (IN (PLACE) LOC) (ОК (DAY TIME) TIME) (IN (DAY TIME) TIME))
G-RULES (Ф-правила)	((BUYER ACTIVE THINGBT (FROM SELLER) (FOR THINGGIVEN)) (THINGBT PASSIVE(FROM SELLER) (FOR THINGGIVEN)))
L-COST (СТОИТЬ) G-RULES	 ((THINGBT ACTIVE (BUYER) (THINGGIVEN)))

Продолжение табл. 11.6

L-PAY (ПЛАТИТЬ) G-RULES	((BUYER ACTIVE (SELLER) (THINGGIVEN) (FOR THINGBT)))
L-WHEN (КОГДА) WORD CLASS SEM	Q WORD (ВОПР. СЛОВО) (DAY DAYPART) (ДЕНЬ, ЧАСТЬ ДНЯ)
EXCHANGE SURF-VB (ПОВЕРХН. ГЛ.)	(L-BUY L-SELL L-PAY L-COST)

Из табл. 11.6 мы можем увидеть, что глагол BUY (ПОКУПАТЬ) для выражения глубинного отношения SELLER (ПРОДАВЕЦ) использует предлог FROM (У) и, кроме того требует, чтобы семантическая категория поверхностного существительного относилась к классу ЧЕЛОВЕЧЕСКИЙ или ОРГАНИЗАЦИЯ. Очевидно, что слово МЭРИ в рассматриваемом примере удовлетворяет обоим условиям и, следовательно, выражает отношение SELLER.

Предполагается, что в словаре каждое существительное и вопросительное слово (Q WORD) имеет специальное свойсво (SEM), указывающее семантическую категорию, к которой существительное или вопросительное слово принадлежит. Результатом разбора входного предложения с применением РАСП являются две (в случае вопроса три) вспомогательные структуры. Например, для предложения WHO BOUGHT A CAKE AT THE NEW BAKERY FROM THE BAKER? (КТО КУПИЛ ПИРОЖНОЕ В НОВОЙ БУЛОЧНОЙ У БУЛОЧНИКА?) мы получим три вспомогательные структуры:

Глагольная составляющая:

(CANON-VB EXCHANGE MODAL (TENSE PAST MOOD
INTERROG CASE AFFIRM))

(КАНОН. ГЛ. ОБМЕНЯТЬ МОДАЛ. (ВРЕМЯ ПРОШ.
НАКЛ. ИЗЪЯВИТ. ПАДЕЖ УТВЕРД)).

Составляющая существительных:

(OK(PHYSOBJ) (TOK L-CAKE DET INDEF NBR S))

(AT(PLACE) (ТОК L-BAKERY DET DEF NBR S
MOD (AGE L-NEW)))
(FROM (HUMAN) (ТОК L-BAKER DET DEF NBR S))
(БП (ФИЗИЧ. ОБ.) (ОБОЗН. L-ПИРОЖНОЕ
АРТИКЛЬ НЕОПР. ЧИСЛО ЕДИНСТВ.))
(В(МЕСТО)(ОБОЗН. L-БУЛОЧНАЯ АРТИКЛЬ ОПРЕД.
ЧИСЛО ЕДИНСТВ. МОД.(ВОЗРАСТЬ-НОВЫЙ)))
(У (ЧЕЛОВЕЧ.) (ОБОЗН. L-БУЛОЧНИК АРТИКЛЬ
ОПРЕД. ЧИСЛО ЕДИНСТВ.)).

Составляющая вопросительного слова:
(OK(HUMAN) (ТОК L-WHO))
(БП (ЧЕЛОВЕЧ.) (ОБОЗН. L-КТО)).

Глагольная составляющая указывает на канонический глагол, определяющий предложение, и на характеризации глагола (модальность МОДАЛ). Составляющие существительного определяют предложные и беспредложные (БП) существительные в порядке их появления во входном предложении. **Каждая составляющая представляет собой триплет**. **Первый элемент триплета** указывает синтаксическую форму (предлог или его отсутствие), в которой существительное появилось в предложении. **Второй элемент** определяет семантическую категорию существительного (скопированную из словаря). **Третий элемент** является списком, описывающим свойства данного существительного из входного предложения. Составляющая вопроситечьного слова представляется в виде триплета со структурой, аналогичной триплету существительного.

Окончательная стадия отображения входного предложения в каноническую структуру состоит в сопоставлении составляющих существительных и вопросительного слова с А-правилами поверхностного глагола входного предложения. Сопоставление заключается в поиске для каждой составляющей существительного А-правила, у которого первый элемент триплета совпадает с первым элементом составляющей, а вторые элементы образуют непустые пересечения. Глубинное отношение, указываемое при этом третьим элементом А-правила, связывается с третьим элементом составляющей (если оно еще не было связано с ранее сопоставляемой составляющей). Когда заканчивается процесс сопоставления составляющих существительных, переходят к сопоставлению составляющей вопросительного слова. Для вопросительного слова сопоставление делается аналогичным образом среди оставшихся несопоставленными отношений. Отличие состоит в том, что результирующей структуре назначается псевдоотношение Q (QUESTION— ВОПРОС), а

подходящие отношения (в нашем примере это одно отношение— BUYER) собираются в список ARGS (АРГУМЕНТЫ).

Таким образом, **<u>каноническая структура</u>, полученная для рассматриваемого входного предложения, имеет вид**:

(CANNON-VB EXCHANGE MODAL (TENSE PAST
MOOD INTERROG CASE AFFIRM)
SELLER (ТОК L-BAKER DET DEF NBR S)
(THINGBT (ТОК L-CAKE DET INDEF NBR S)
LOC (ТОК L-BAKERY DET DEF NBR S MOD (AGE
L-NEW))
Q (ARGS (BUYER) ТОК L-WHO))
(КАНОН. ГЛ. ОБМЕН ЯТЬ МОДАЛ. (ВРЕМЯ ПРОШ.
НАКЛ. ИЗЪЯВИТ. ПАДЕЖ УТВЕРД.)
ПРОДАВЕЦ (ОБОЗН. L-БУЛОЧНИК АРТИКЛЬ
ОПРЕД. ЧИСЛО ЕД)
ПОЛУЧЕННОЕ (ОБОЗН. L-ПИРОЖНОЕ
АРТИКЛЬ НЕОПР. ЧИСЛО ЕД)
МЕСТО (ОБОЗН. L-БУЛОЧНАЯ АРТИКЛЬ ОПРЕД.
ЧИСЛО ЕД. МОД. (ВОЗРАСТ L-НОВЫЙ))
ВОПРОС (АРГ. (ПОКУПАТЕЛЬ) ОБОЗН. L-КТО)).

Изложенный способ <u>отображения входного предложения в каноническую структуру</u> рассчитан на простые активные предложения без вложенных конструкций. Подход может быть расширен в направлении включения:

1) более сложных А-правил, явно выражающих упорядоченность существительных в списке составляющих;
2) более сложного алгоритма сопоставления, распознающего относительные предложения;
3) более разнообразных синтаксических форм и семантических категорий;
4) включение пассивных конструкций и вложенных предложений.

Перейдем теперь к вопросу объединения информации входного предложения с базовыми данными. **<u>Базовые данные</u>, представленные в виде семантической сети, <u>хранят информацию об объектах и отношениях между объектами</u>.**

На рис. 11.14. приведен пример текущего состояния семантической сети.

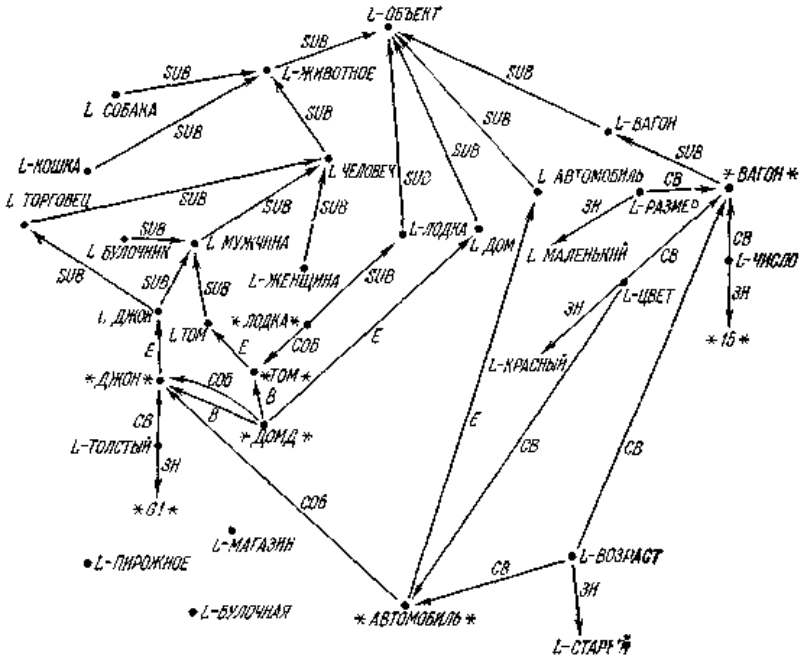


Рис. 11.14. Пример текущего состояния семантической сети.
SUB — ПОДМНОЖЕСТВО; Е — ЭЛЕМЕНТ; СОБ — СОБСТВЕННОСТЬ; В — НАХОДИТЬСЯ В; СВ — СВОЙСТВО; ЗН — ЗНАЧЕНИЕ.

В дальнейшем при описании сетей мы будем наравне с термином «вершины» использовать термин «узел». *Фактуальные вершины* будем представлять узлами, наименования которых начинаются и заканчиваются звездочкой (*), а *общие вершины*— узлами, наименованиям которых предшествует буква L. Рассмотрим узел *ДЖОН*, являющийся элементом L-ДЖОН и L-ТОРГОВЕЦ. L-ДЖОН является обозначением множества всех Джонов и подмножеством L-МУЖЧИНА (множества всех мужчин). L-ТОРГОВЕЦ есть множество всех торговцев. Следовательно, *ДЖОН* является узлом, представляющим конкретного человека, являющегося торговцем с именем Джон.

Этот Джон владеет конкретным домом *ДОМД* (что выражается отношением СОБСТВЕННОСТЬ). Джон находится в этом доме. Джон владеет старым красным автомобилем.

Для представления событий, происходящих во времени, используется понятие временной оси (рис. 11.15).

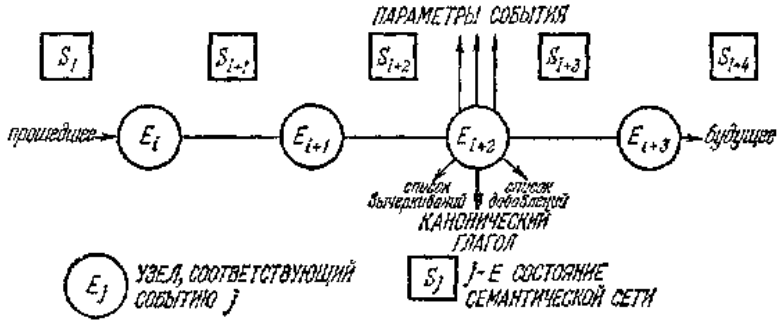


Рис. 11.15. Абстрактная временная ось.

Каждый узел на <u>временной оси</u> соответствует <u>событию (предложению в естественном языке)</u>. В начальном состоянии сеть содержит только некоторую информацию из словаря. **Состояние сети изменяется в соответствии с последовательностью событий, происходящих на временной оси.** С каждым событием связывается список отношений (механизм связи будет определен ниже), имеющих место до совершения события и более неудовлетворяющихся после совершения события (так называемый **список вычеркивания**) и список отношений, которые не имели место до рассматриваемого события (или по крайней мере не было известно, что эти отношения выполняются), но которые удовлетворяются после выполнения события (**список добавлений**). Отметим, что появление события на временной оси изменяет не только отношения, существующие в семантической сети, но и может привести к добавлению новых узлов, если во входном предложении были объекты, отсутствовавшие в сети.

Рассмотрим в качестве примера изменения, которые вызовет в сети (см. рис. 11.14) событие: ДЖОН ОБМЕНЯЛ С ТОМОМ СВОЙ АВТОМОБИЛЬ НА ЛОДКУ ТОМА. Отношения (СОБСТВЕННОСТЬ *ДЖОН**АВТОМОБИЛЬ*) и (СОБСТВЕННОСТЬ *ТОМ** ЛОДКА*), существовавшие в сети на рис. 11.14 больше не являются истинными и должны быть заменены на отношения (СОБСТВЕННОСТЬ *ДЖОН ** ЛОДКА*) и (СОБСТВЕННОСТЬ * ТОМ* «АВТОМОБИЛЬ*). Указанные изменения в сети изображены на рис. 11.16 (состояние 2).

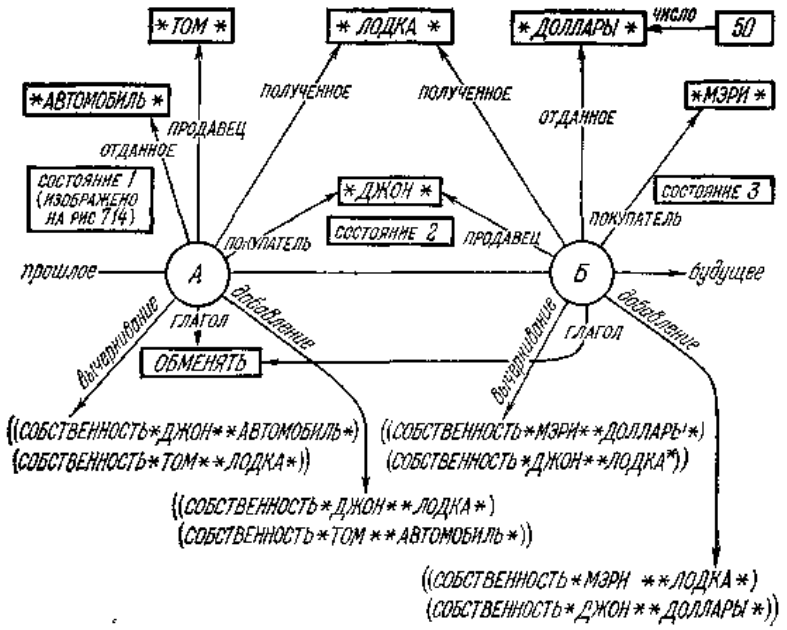


Рис. 11.16. Пример состояния временной оси при поступлении двух событий (вспомогательные узлы для простоты не показаны).

Так как события преобразуют одно состояние сети в другое, можно рассматривать их как операторы. Следуя этой трактовке, канонический глагол можно рассматривать как процедуральное событие (или оператор), которое преобразует состояние сети (являющееся неявным параметром) и множество явных параметров в новое состояние сети. Процедуральные события представляются аналогично операторам, используемым в системе STRIPS (см. п. 10.2).

Неотъемлемой частью любого оператора STRIPS является список предусловий, который используется пользователем для определения законной последовательности действий. Так как в рассматриваемой системе предполагается, что события следуют в хронологической последовательности, то исчезает необходимость в списке предусловий для определения последовательности событий (операторов). Существует однако следующая проблема. Пусть на вход поступили предложения: ДЖОН КУПИЛ ЧАСЫ В МАГАЗИНЕ, ЗАТЕМ ДЖОН КУПИЛ ПИРОЖНОЕ В НОВОЙ БУЛОЧНОЙ. Заметим, что Джон покупал часы в магазине, а в следующем событии он покупал пирожное в булочной. До того как событие по покупке пирожного

может иметь место, Джон должен перейти из магазина в булочную (предусловие для события в булочной). Таким образом, мы видим, что необходимость в удовлетворении предусловиям остается даже при заданной последовательности событий. Предусловия определяют неспецифицированные вспомогательные события, которые необходимы для выполнения специфицированных (указанных в тексте) событий.

Сущность этих неспецифицированных вспомогательных событий в значительной степени определяется предусловиями специфицированных событий. Так, в приведенном примере до того, как ДЖОН КУПИЛ ПИРОЖНОЕ В НОВОЙ БУЛОЧНОЙ необходимо выполнить некоторое событие, устраняющее отношение (В*ДЖОН ** МАГАЗИН*) и добавляющее отношение (В*ДЖОН ** БУЛОЧНАЯ *). (В общем случае событию ОБМЕНЯТЬ должно предшествовать событие, переносящее участников обмена в место обмена и т. д.)

Таким образом, процедуральное событие приводит к вызову вспомогательного события и созданию на временной оси соответствующих им двух узлов. Процедуральное событие подставляет списки добавления и вычеркивания неспецифицированного события, которое преобразует текущее состояние сети в состояние, удовлетворяющее предусловиям главного события. То, что такое событие должно существовать, следует из заданной последовательности специфицированных событий.

Заметим, что описанный способ обработки <u>процедурального события</u> является одним из возможных вариантов представления события в виде <u>скелета (фрейма</u>).

Рассмотрим на примере, как осуществляется вызов и обработка процедурального события. Пусть входное предложение имеет вид:

AT THE NEW BAKERY JOHN BOUGHT A CAKE
FROM THE BAKER
(В НОВОЙ БУЛОЧНОЙ ДЖОН КУПИЛ ПИРОЖНОЕ
У БУЛОЧНИКА).

После этапа синтаксического разбора и приведения к канонической структуре оно примет вид:

(КАНОН. ГЛ. ОБМЕНЯТЬМОДАЛ. (ВРЕМЯ ПРОШ.
НАКЛ. ИЗЪЯВИТ. ПАДЕЖ УТВЕРДИТ.)
ПРОДАВЕЦ (ОБОЗН. L-БУЛОЧНИК АРТИКЛЬ
ОПРЕД. ЧИСЛО ЕД.)
ПОКУПАТЕЛЬ (ОБОЗН. L-ДЖОН ЧИСЛО ЕД.)
ПОЛУЧЕННОЕ (ОБОЗН. L-ПИРОЖНОЕ АРТИКЛЬ
НЕОПР. ЧИСЛО ЕД.)
МЕСТО (ОБОЗН. L-БУЛОЧНАЯ АРТИКЛЬ

ОПРЕД. ЧИСЛО ЕД. МОД. (ВОЗРАСТ L-НОВЫЙ))).

Канонический глагол указывает, какая процедура должна быть вызвана (табл. 11.7).

Таблица 11.7

Представление процедурального события «обменять»

```
ПРОЦЕДУРА ОБМЕНЯТЬ (продавец, покупатель, полученное,
                                        отданное, место);
СПИСОК ВЫЧЕРКИВАНИЙ ВСПОМОГАТЕЛЬНОГО СОБЫТИЯ:
        ((В продавец *) (В покупатель *) (В полученное *)
         (В отданное *) (СОБСТВЕННОСТЬ * полученное)
         (СОБСТВЕННОСТЬ * отданное));
СПИСОК ДОБАВЛЕНИЙ ВСПОМОГАТЕЛЬНОГО СОБЫТИЯ:
        ((В продавец место) (В покупатель место)
         (В полученное место) (В отданное место)
         (СОБСТВЕННОСТЬ продавец полученное)
         (СОБСТВЕННОСТЬ покупатель отданное));
СПИСОК ВЫЧЕРКИВАНИЙ ГЛАВНОГО СОБЫТИЯ:
        ((СОБСТВЕННОСТЬ покупатель отданное)
         (СОБСТВЕННОСТЬ продавец полученное));
СПИСОК ДОБАВЛЕНИЙ ГЛАВНОГО СОБЫТИЯ:
        ((СОБСТВЕННОСТЬ покупатель полученное)
         (СОБСТВЕННОСТЬ продавец отданное)).
```

Параметры процедуры (возможно не все) в явном виде вырабатываются на стадии разбора. В приведенном примере это: продавец, покупатель, полученное, отданное, место. Остальным параметрам процедуры, не получившим на стадии разбора конкретных значений, присваивается значение NIL. Затем с каждым параметром должен быть связан некоторый узел сети. Если подходящего узла в сети не существует, то такой узел должен быть создан. Эту функцию выполняет специальная программа НИС (найти или создать), чьими аргументами являются параметры выбранной процедуры. Например, если артикль параметра является неопределенным, то НИС просто создает новый узел в сети, удовлетворяющий списку свойств параметра. Так, для А САКЕ, упомянутого во входном предложении, НИС создает новый узел, не заботясь о том, связан он или нет с некоторым понятием ПИРОЖНОЕ, существующим в сети. (Впоследствии необходимо определить различные узлы, выражающие одну и ту же сущность. Объединение этих узлов выполняется специальной функцией.) Если у параметра определенный (или специфированный) артикль, то НИС пытается найти в сети

соответствующий этому параметру узел. Если найдено больше одного узла, то выбирается тот, который упоминался последним. Если не найдено ни одного узла, то создается новый узел.

Значением свойства ОБОЗН, указанного в списке свойств параметра, является наименование узла, представляющего множество объектов. Значением параметра является элемент из этого множества. В нашем примере для параметра ПРОДАВЕЦ наименованием множества является L-БУЛОЧНИК (именующий множество всех булочников). Следовательно, значением параметра ПРОДАВЕЦ должен быть элемент из множества L-БУЛОЧНИК. Заметим, что слова типа L-БУЛОЧНИК, L-ДЖОН, L-ПИРОЖНОЕ введены и в словарь (для целей разбора), и в сеть перед началом функционирования системы. Кроме того, в сеть введены и определенные примитивные отношения между подобными словами, например, (ПОДМНОЖЕСТВО L-ДЖОН L-МУЖЧИНА). Всякий раз, когда упоминается некоторый *ДЖОН*, он становится элементом множества L-ДЖОН, являющегося в свою очередь подмножеством множества L-МУЖЧИНА.

Параметры указанного входного предложения будут обработаны следующим образом:

1) Программа НИС ищет в сети *x* такое, что для него справедливо отношение (ЭЛЕМЕНТ *x* L-БУЛОЧНИК). Не найдя такого *x,* программа создает узел * БУЛОЧНИК *, и этот узел становится значением параметра ПРОДАВЕЦ. (В течение этого процесса отношение (ЭЛЕМЕНТ * БУЛОЧНИК * L-БУЛОЧНИК) вводится в сеть).

2) Программа НИС ищет такое *x,* что (ЭЛЕМЕНТ *x* L-ДЖОН). Она находит узел «ДЖОН», который и становится значением параметра ПОКУПАТЕЛЬ.

3) В связи с тем, что у параметра ПОЛУЧЕННОЕ неопределенный артикль, НИС создает новый узел *ПИРОЖНОЕ* такой, что для него выполняется отношение (ЭЛЕМЕНТ * ПИРОЖНОЕ* L-ПИРОЖНОЕ), и он является значением параметра ПОЛУЧЕННОЕ.

4) В связи с присутствием у параметра МЕСТО модификатора (прилагательного) работа программы НИС усложняется. НИС ищет *x* такое, что (ЭЛЕМЕНТ *x* L-БУЛОЧНАЯ) и (ВОЗРАСТ НОВЫЙ *x).* Не найдя такого узла, НИС создает его (*БУЛОЧНАЯ*) и делает значением параметра МЕСТО.

Как только значения параметров определены, система вызывает процедуру ОБМЕНЯТЬ для того, чтобы обработать поступившее событие. Важно отметить, что отношения, вводимые в сеть в процессе определения значений параметров, не вводятся на списки вычеркивания и добавления узлов временной оси.

Итак, обращение к процедуральному событию (табл. 11.7) имеет вид: ОБМЕНЯТЬ (* БУЛОЧНИК ** ДЖОН ** ПИРОЖНОЕ * NIL * БУЛОЧНАЯ *), где NIL присваивается параметру процедуры ОТДАННОЕ, отсутствующему в поступившем событии. Эффект, вызываемый обращением к процедуре, представлен на рис. 11.17.

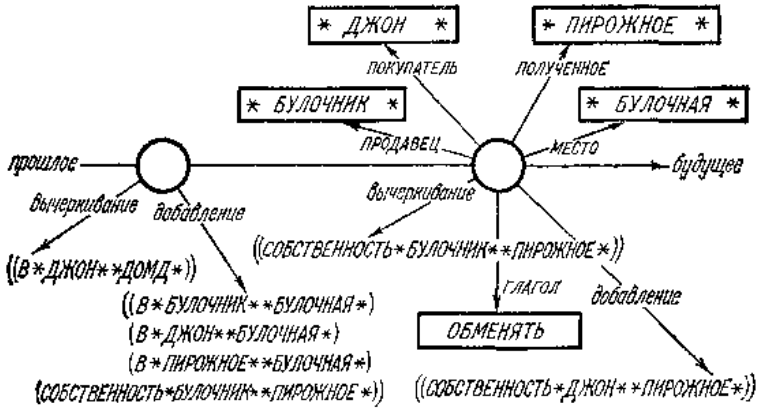


Рис. 11.17. Узлы временной оси, созданные процедуральным событием.

Рассмотрим этапы выполнения процедуры:

1) На временной оси создается узел, соответствующий вспомогательному событию. Так как природа события неизвестна, с узлом не связывается ни канонический глагол, ни параметры события.

2) Рассматривается список вычеркиваний вспомогательного события (табл. 11.7). Первым рассмотрим триплет (В ПРОДАВЕЦ *). Делается попытка сопоставить это отношение с отношениями, существующими в сети. Звездочка, стоящая на месте третьего элемента триплета, позволяет устанавливать соответствие с любым объектом сети. Параметру «продавец» при вызове процедуры сопоставлен узел *БУЛОЧНИК *. Итак, нам надо искать отношение в виде (В*БУЛОЧНИК*—). Такого отношения в сети нет, и поэтому никакие действия не предпринимаются. Следующим рассматривается отношение (В ПОКУПАТЕЛЬ*). Подстановка фактических параметров (при вызове процедуры) преобразует его в (В* ДЖОН * —). Этому отношению в сети (рис. 11.14) соответствует отношение (В*ДЖОН **ДОМД*). Следовательно, (так как мы рассматриваем список вычеркиваний) отношение (В * ДЖОН * * ДОМД *) вычеркивается из сети и устанавливается в список вычеркиваний узла, представляющего на временной оси вспомогательное событие. Остальным отношениям,

перечисленным в списке вычеркиваний процедуры, не удается сопоставить отношения сети, и поэтому они не вызывают никаких действий.

3) Рассматривается список добавлений вспомогательного события (табл. 11.7). Первое рассматриваемое отношение (В ПРОДАВЕЦ МЕСТО) после подстановки фактических параметров преобразуется в отношение (В*БУЛОЧНИК ** БУЛОЧНАЯ *). Это отношение добавляется к сети и к списку добавлений вспомогательного узла временной оси. Аналогичным образом добавляется следующие отношения (В * ДЖОН ** БУЛОЧНАЯ *), (В * ПИРОЖНОЕ ** БУЛОЧНАЯ *) и (СОБСТВЕННОСТЬ*БУЛОЧНИК ** ПИРОЖНОЕ*). Триплеты, содержащие параметр «отданное», не вызывают никаких действий, так как этот параметр неопределен поступившим событием.

4) На временной оси устанавливается узел, соответствующий главному событию. С данным узлом связываются параметры процедуры ОБМЕНЯТЬ и канонический глагол ОБМЕНЯТЬ.

5) Рассматривается список вычеркиваний главного события (табл. 11.7) и вырабатывается отношение (СОБСТВЕННОСТЬ * БУЛОЧНИК ** ПИРОЖНОЕ *), устраняемое из сети и заносимое в список вычеркиваний главного события.

6) Обрабатывается список добавлений главного события и добавляется отношение (СОБСТВЕННОСТЬ * ДЖОН * * ПИРОЖНОЕ*).

Описанной процедурой первоначальное состояние сети (рис. 11.14) преобразуется во вспомогательное состояние, в котором Джон и булочник находятся в булочной, и булочник владеет пирожным. Затем это вспомогательное состояние преобразуется в состояние, в котором Джон и булочник остаются в булочной, но владельцем пирожного становится Джон.

Итак, мы рассмотрели основные этапы семантической интерпретации в системах с ограниченной логикой при представлении базы данных в виде семантической сети. Перейдем теперь к описанию СИ при представлении базовых данных в исчислении предикатов.

11.5.3. Семантическая интерпретация в системах с общим выводом

Описание СИ систем с общим выводом дадим на базе вопросно-ответной системы. Основанием для выбора нами этой системы является следующее:

1) законченность системы (т. е. в системе реализованы все задачи диалога);

2) система разработана для представления фреймов;

3) система позволяет продемонстрировать альтернативные подходы в области грамматики, внутреннего представления и взаимосвязи этапов семантической интерпретации и синтаксического анализа (СА).

В связи с тем, чго в данной системе этапы СИ и СА выполняются параллельно (а не последовательно), описанию СИ необходимо предпослать краткие сведения о синтаксическом анализе

Синтаксический анализ системы основан на трансформационной грамматике (п. 11.2.3) подмножества английского языка, включающего повелительные, повествовательные и вопросительные предложения.

Задачи трансформационных правил состоят в преобразовании пассива в актив, вопросительных предложений в повествовательные, отображении множественных форм существительных и глаголов в единые формы.

Так, например, при обработке предложения с пассивной конструкцией «THE TALL BOX WAS PUSHED BY JOHN» (высокий ящик толкался Джоном), задачей трансформационной компоненты является определение необходимости применения трансформации (пассив→актив), которая переведет данное предложение в активную форму: «JOHN PUSHED THE TALL BOX» (Джон толкал высокий ящик). Затем к работе приступает базовая компонента. Анализатор базовой компоненты записывается в виде правил продукции, имеющих следующий вид:

$L1$*:* $\alpha \mid \rightarrow \beta \mid \gamma * L2$*;*

где $L1$ и L2— метки, α и β есть строки, → указывает операцию замещения, | — разделитель, γ есть последовательность семантических продукций, * указывает операцию «ЧТЕНИЕ» (выбора очередного слова из входной строки и размещение его на вершине стека синтаксического анализатора), ; — знак, разъединяющий продукции. $L1$, →, β, γ,* являются необязательными символами, в то время как |, α, $L2$ являются обязательными для каждой продукции. Рассмотрим, как осуществляется выполнение продукции. Предполагается, что в начале работы в верхний уровень синтаксического стека устанавливается первый (левый) символ входного предложения. Пусть управление передано продукции, помеченной меткой $L1$. Символ α правила $L1$ сопоставляется с верхними уровнями стека. Если сопоставление закончилось успешно, то:

1) часть стека, соответствующая α, заменяется на β (свободный класс переменных становится связанным в смысле исчисления предикатов);

2) выполняется последовательность γ (последовательность семантических продукций правила);

3) если * присутствует в продукции *L*1, то выполняется операция «чтения»;
4) совершается переход к продукции, с меткой *L2*.
Если сопоставление строки α правила *L*1 со стеком неудачно, то управление передается следующей продукции в последовательности. Операция сопоставления понимается в смысле сопоставления с образцом. Образец (α-строка правила) может содержать терминальную константу, класс переменных, определенных в терминах терминальных констант или в терминах булевской комбинации других классов. Образец может иметь вид $1, который сопоставляется с произвольной одиночной составляющей. Образец $ сопоставляется с любым числом произвольных составляющих.
Продукции трансформационной компоненты имеют такую же форму, как продукции базовой компоненты, за исключением того факта, что сопоставление осуществляется не со стеком, а с самим предложением (просмотром его слева направо). Любой элемент образца, взятый в кавычки, указывает на то, что сопоставление выполняется на уровне букв частного слова, а не на лексическом уровне. Таким образом могут быть выполнены проверки на множественное число и на стандартные суффиксы и префиксы.
Семантические продукции идентичны синтаксическим по форме и выполнению. Отличие состоит в том, что в семантических продукциях не используется операция * и, кроме того, семантические продукции выполняются на семантическом стеке.
В табл. 11.8 приведены примеры предложений ограниченного английского языка, используемые при обращении к отправителю.

Таблица 11.8

Образцы перевода предложений естественного языка в формулы исчисления предикатов

Предложения	Формулы
Повествовательные	
1) Все люди смертны	$\forall x$ (ЕСТЬ (x, человек) $\rightarrow$ ЕСТЬ (x, смертен))
2) Существуют высокие представители мужского пола, не являющиеся мальчиками	$\exists x$ (ЕСТЬ (x, высокий) $\wedge$ ЕСТЬ (x, мужчина) $\wedge \sim$ЕСТЬ (x, мальчик))
3) У Джона две руки	ИМЕТЬ (Джон, рука, 2)
4) Любой зеленый объект является высоким, узким ящиком	$\forall x$ (ЕСТЬ (x, зеленый) $\rightarrow$ ЕСТЬ (x, высокий) $\wedge$ ЕСТЬ (x, узкий) $\wedge$ ЕСТЬ (x, ящик))
5) Высокий ящик толкался Джоном	$\exists s, x, y, z$ (РАВНО (S, ТОЛКАТЬ (ДЖОН, x, y, z, $S_{нач}$)) $\wedge$ ЕСТЬ (x, высокий) $\wedge$ ЕСТЬ (x, ящик) $\wedge$ ВРЕМЯ (S, прошл.))
Вопросительные	
6) Является ли Джон мужчиной?	ЕСТЬ (Джон, мужчина)
7) Сколько пальцев имеет Джон?	$\exists x$ (ИМЕТЬ (Джон, палец, x))
8) Есть ли слева от Вас какие-нибудь ящики?	$\exists x$ (ЕСТЬ (x, ящик) $\wedge$ СЛЕВА (x, Отправ))
9) Когда вы будете толкать ящик?	$\exists s, t, x, y, z$ (РАВНО (S, ТОЛКАТЬ (Отправ., x, y, z, $S_{нач}$)) $\wedge$ ЕСТЬ (x, ЯЩИК) $\wedge$ ВРЕМЯ (s, t) $\wedge$ БУДУЩЕЕ (t))
Повелительные	
10) Остановиться 11) Повернуться кругом 12) Двигаться на 10 футов	Осуществляется обращение к подпрограммам, реализующим указанные стандартные действия.
13) Идти к большой красной призме	$\exists s, x$ (У (Отправ., x, s) $\wedge$ ЕСТЬ (x, большой) $\wedge$ ЕСТЬ (x, красный) $\wedge$ ЕСТЬ (x, призма))
14) Толкать черный ящик на вершину платформы	$\exists s, x, y, z$ (ТОЛКАТЬ (x, s) $\wedge$ ЕСТЬ (x, черный) $\wedge$ ЕСТЬ (x, ящик) $\wedge$ НА (x, y) $\wedge$ ЕСТЬ (y, вершина) $\wedge$ ПРИНАДЛЕЖИТ (y, z) $\wedge$ ЕСТЬ (z, платформа))
15) Собрать все кубы в центре комнаты	$\forall x \exists s, y, z$ (В (x, y, s) $\wedge$ ЕСТЬ (x, куб) $\wedge$ ЕСТЬ (y, центр) $\wedge$ ПРИНАДЛЕЖИТ (y, z) $\wedge$ ЕСТЬ (z, комната))
16) Исследовать комнату Джона	$\exists s, x$ (ИССЛЕДОВАТЬ (x, s) $\wedge$ ЕСТЬ (x, комната) $\wedge$ ПРИНАДЛЕЖИТ (x, Джон))

Для удобства читателей предложения переведены на русский язык. В таблице 11.8 приведены формулы исчисления предикатов, соответствующие предложениям естественного языка.

Повествовательные предложения (примеры 1-5 табл. 11.8) переводятся в аксиомы системы, вопросительные предложения (примеры 6—9 табл. 11.8) переводятся в утверждения, которые должны быть доказаны вопросно-ответной системой. Простые повелительные предложения (примеры 10-12 табл. 11.8) непосредственно переводятся в наименования стандартных программ, вызывающих выполнение отправителем определенных действий. Сложные повелительные предложения (примеры 13-16 табл. 11.8) рассматриваются системой как утверждения, возможность выполнения которых должна быть определена, т. е. они аналогичны вопросам.

Большинство предикатов, введенных в примерах, очевидны, однако, для ясности рассмотрим один пример (уже упоминавшийся ранее): «Высокий ящик толкался Джоном» (THE TALL BOX WAS PUSHED BY JOHN). Синтаксическая компонента, получив данное предложение, определяет, что к предложению необходимо применить трансформацию, переводящую его в активную форму «Джон толкал высокий ящик» (JOHN PUSHED THE TALL BOX). Затем базовая компонента распознает, что глагол «толкать» использован в предложении в прошедшем времени, и устанавливает в соответствующее значение предикат ВРЕМЯ. Прилагательное «высокий» и существительное «ящик» представляются в предикате ЕСТЬ. Полученная в результате формула может быть проинтерпретирована примерно так. **Существуют состояние S, объект *x* и координаты *y* и *z* такие, что S есть состояние, полученное из начального ($S_{нач}$) применением к нему оператора (Джон толкает объект *x* из координат *y* и *z*), причем *x* относится к классу высоких объектов и к классу ящиков. Кроме того, состояние S имело место в прошлом.**

Способ представления предложений в виде формул исчисления предикатов не является единственно возможным.

Сэндуолл указывает три возможных способа. Проиллюстрируем их на примере предложения «Джон продал лодку Мэри».

1) Глагол используется как предикатный символ. После трансляции входное предложение будет иметь вид ПРОДАЛ (Джон, Лодка, Мэри), где ПРОДАЛ является трехместным отношением, а Джон, Лодка, Мэри — константы. Если мы хотим указать дополнительные сведения о происходящем событии (например, время, место), то это может быть сделано введением предиката, содержащего большее количество аргументов.

2) Главным отличием этой нотации является использование небольшого множества базовых отношений и получение на их основе более сложных отношений благодаря введению понятия «события».
Рассматриваемое предложение после трансляции может иметь вид:
(∃ еПОДЛЕЖАЩЕЕ (Джон, е) ∧ СКАЗУЕМОЕ (продал, е) ∧ ДОПОЛНЕНИЕ (лодка, е) ∧ КОСВ.
ДОПОЛН. (Мэри, е) ∧ РЕАЛЬНОСТЬ (е)),
где «е» интерпретируется как событие, в результате которого «Джон продал лодку Мэри». Первые четыре отношения используют для того, чтобы описать данное событие, а предикат РЕАЛЬНОСТЬ используется для того, чтобы определить, что указанное событие фактически имеет место. В данной нотации можно представить и более сложные предложения, например: «Джон верит, что Петр продал лодку Мэри».
(∃ е ∃ р ПОДЛЕЖАЩЕЕ (Джон, е) ∧ СКАЗУЕМОЕ
(верящий, е) ∧ ДОПОЛНЕНИЕ (р, е)
∧ РЕАЛЬНОСТЬ (е) ∧ ПОДЛЕЖАЩЕЕ (Петр, р) ∧
СКАЗУЕМОЕ (продавший, р) ∧ ДОПОЛНЕНИЕ
(Лодка, р) ∧ КОСВ. ДОПОЛН. (Мэри, р)),
где «р» интерпретируется как событие: «Петр продал лодку Мэри».
Предикаты, используемые для описания события, существенно зависят от конкретной системы, так, если в системе используются глубинные падежи (см. п. 11.2.3), то вместо приведенных в примере предикатов ПОДЛЕЖАЩЕЕ, СКАЗУЕМОЕ, ДОПОЛНЕНИЕ и т. д. будут использоваться предикаты, отражающие глубинные отношения.
3) Третий способ отличается тем, что отношения между словами предложения выражаются не предикатами, а функциональными символами.
ЕСТЬ (Джон, Кому Юбъект (продавший, лодку), Мэри)).
Данное выражение можно представить в виде бинарного дерева (рис. 11.18), отражающего структуру предложения.

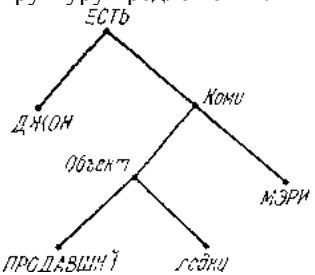


Рис 11.18. Представление структуры предложения в виде бинарного дерева.

В данном методе (так же как и в предыдущем) используются объекты (Джон, лодка, Мэри) и свойства (продавший). Отношение ЕСТЬ определено на **двух аргументах: первый — объект, второй — свойства**. Функции «Объект», «Кому» являются двухместными. **Первый аргумент этих функций должен быть свойством, второй объектом, а результат является свойством**. Подобным образом могут быть определены более сложные высказывания, содержащие кванторы, сравнительную степень прилагательных, придаточные предложения и т. п.

Первый метод широко используется. Причины его популярности вызваны тем, что для людей является естественной идентификация **понятия «предикат» из традиционной грамматики с понятием «предикат» в исчислении предикатов**. Однако этот метод имеет и очевидные недостатки. В нем затруднено выражение ссылок на предшествующий контекст, необходимо введение большого числа предикатов и функций при описании объектов реальной сложности.

Второй метод используется в нескольких вопросно-ответных системах. Этот метод близок к падежной грамматике Филмора (см. п. 11.2.3). Этот подход позволяет одному предложению ссылаться на другое и не требует введения большого числа предикатов.

Третий метод имеет некоторое сходство с глубинными структурами трансформационных грамматик (см. п. 11.2.3). Можно найти большое сходство между деревом на рис. 11.18 и традиционным С-маркером для предложения в естественном языке.

Второй и третий методы имеют преимущества перед первым и много общего между собой. Они вводят относительно немного функций и отношений, которые являются базовыми для **отражения лингвистического и концептуального состава языка.**

Итак, можно подытожить, что исчисление предикатов обладает достаточной выразительной мощностью для представления «глубинной структуры» предложений естественного языка.

Следует отметить различие в понимании термина «глубинные структуры» в лингвистике и при использовании в качестве глубинных структур формул исчисления предикатов. **В лингвистике глубинная структура есть результат синтаксического анализа входного предложения до его объединения с базовыми данными, а в исчислении предикатов — после объединения.**

Основным достоинством использования исчисления предикатов в качестве глубинных структур является существование для них универсальных вычислительных методов.

Для реализации конкретной системы с использованием исчисления предикатов необходимо решить две задачи:

1. Сформировать каноническое множество предикатов и функций, достаточных для описания среды, в которой будет функционировать ИСФ. Множество должно быть достаточно малым, но полным.
2. Сформулировать аксиомы, выражающие свойства понятий, введенных в пункте 1.

В таблице 11.9 приведен экспериментальный список основных понятий, используемых для описания существенных свойств действий.

Таблица 11.9

Каноническое множество предикатов, используемых для описания действия

Предикаты	Вопросы
1. АГЕНТ (x, y)	Кто совершил действие?
2. ДЕЙСТВИЕ (x, y)	Что он сделал?
3. ОБЪЕКТ (x, y)	Для кого или чего он это сделал?
4. Описание окружения:	В каком контексте он это сделал?
ВРЕМЯ (x, y)	Когда он это сделал?
МЕСТО (x, y)	Где он это сделал?
ОБСТОЯТЕЛЬСТВО (x, y)	При каких обстоятельствах он это сделал?
5. Модальность:	Как он это сделал?
СРЕДСТВО (x, y)	С помощью какого инструмента или метода он это сделал?
СПОСОБ (x, y)	С помощью чего он это сделал?
6. Причинность:	Почему он это сделал?
ПРИЧИНА (x, y)	Какая причина побудила его сделать это?
ЦЕЛЬ (x, y)	С какими намерениями он это сделал?
СОСТОЯНИЕ (x, y)	В каком состоянии (настроении) он это сделал?

Для того, чтобы назначение конкретного предиката было более понятно, рядом с предикатом приведены вопросы в естественном языке. Не утверждается, что этот список в принципе полон, но для заданных целей он достаточен. Аксиомы (фреймы) должны описывать, по крайней мере, информацию трех видов.

1. Геометрические отношения окружающей среды.

2. Правила, описывающие ограничения на возможности фрейма воспринимать окружающую среду и и выполнять манипуляции в ней.
3. Информацию, извлеченную из повествовательных предложений естественного языка, полученную в ходе диалога.

Следует отметить, что формулировка аксиом (фреймов) является довольно сложной задачей**. Можно предложить следующие неформальные шаги при выполнении аксиоматизации (фреймизации).**

1. Записать наблюдения об окружении, которое мы хотим описать. Выразить все в корректном исчислении предикатов.

2. Проверить на непротиворечивость.

3. Проверить на избыточность, т. е. проверить, не выводимы ли некоторые аксиомы из других аксиом?.

Следует отметить, что наличие избыточности может ускорить вывод, поэтому добиваться отсутствия избыточности не всегда целесообразно.

4. Проверить на полноту, т. е. <u>проверить, введены ли все аксиомы, необходимые для вывода</u>. Общий метод осуществления проверки на полноту состоит в подборе проверяющих примеров (теорем, которые должны быть выводимы в предлагаемой аксиоматике) и проверке вручную, что каждая из них может быть доказана.

Проверяющие примеры могут быть «позитивные» и «негативные». Позитивные примеры — это те, которые используются при прямом выводе. Их задача — показать, что то, что должно быть выводимо, выводится. Негативные примеры должны показывать, что то, что не должно быть выводимо, выводится.

Следует отметить, что предложенные шаги имеют много общего с отладкой программы на вычислительных машинах. Однако аксиомы проще записывать, отлаживать и объединять в общую систему.

Остановимся теперь на вопросе объединения некоторого факта (повествовательное предложение), переведенного в формулу исчисления предикатов, с базовыми данными системы.

Объединить новый факт (*Ф*) с базовыми аксиомами *(А)* можно только в том случае, если этот факт не противоречит аксиомам.

Заметим, что если исходное предложение имеет после синтаксического и семантического этапа более одной интерпретации, то из них выбирается та, которая не противоречит базовым аксиомам (если такая интерпретация существует) .

Проверка на непротиворечивость состоит в определении невыполнимости множества $A \bigcup \sim\Phi$, где A — исходное множество аксиом, $\sim\Phi$ — отрицание объединяемого факта. Практически во всех системах дедуктивного вывода, основанных на исчислении

предикатов, для определения невыполнимости используются модификации принципа резолюции.

При определении невыполнимости возможны три исхода.

1. Множество формул $A \cup \sim\Phi$ невыполнимо (получен пустой дизъюнкт), что обозначает выводимость формулы Φ из базовых аксиом.

2. Множество формул $A \cup \sim\Phi$ выполнимо ($R_{i+1}=R(R_i(A))$), что означает невыводимость формулы Φ из аксиом.

3. Процедура резолюции не дает ответа за время, отведенное на доказательство (из-за зацикливания или из-за ограниченности времени).

В первом случае утверждение Φ, несмотря на его выводимость из A, иногда целесообразно добавить к A, так это может ускорить получение некоторых выводов.

Второй случай обозначает, что Φ невыводима в A. Известно, что

если замкнутая формула Ф теории А невыводима в А, то теория А', полученная из А добавлением ~Ф *в качестве аксиомы, непротиворечива.*

На основании этого утверждения формула $\sim \Phi$ может быть добавлена к A.

Если произошел третий исход, то мы можем попытаться проверить выводимость ~Ф из A (т. е. установить невыполнимость $A \cup \Phi$). При этом опять же возможны три указанных исхода. Если имеет место первый случай, то это значит, что формула $\sim \Phi$ выводима из A и, следовательно, Φ противоречит A. В этом случае вопрос о раскрытии противоречия должен решать человек.

Второй случай указывает на то, что утверждение Φ должно быть добавлено к A.

Если имеет место третий случай, то это обозначает, что система не в состоянии установить (либо в принципе, либо в отведенное время), противоречит ли Φ исходному множеству A. Вероятно, наиболее разумное решение — выдать сообщение об этом факте и ждать указаний о присоединении (или неприсоединении) формулы Φ к исходному множеству аксиом.

11.6. Вывод ответа

11.6.1. Доказательство и извлечение ответа в системах с общим выводом

Входом для этапа дедуктивного вывода является некоторая формула *Ф* исчисления предикатов, полученная из вопросительного или повелительного предложения исходного языка и рассматриваемая как теорема, подлежащая доказательству. Задачей этапа является определение, следует ли данная теорема из базового множества аксиом *(А),* то есть определение невыполнимости множества $A \cup$ ~Ф. Как уже было указано в п. 11.5.3, при определении невыполнимости по методу резолюции возможны три исхода:

1) А $\cup$ ~*Ф* невыполнимо, следовательно, *Ф* выводима из *А;*
2) А $\cup$ ~*Ф* выполнимо, т. е. *Ф* не следует из *А*;
3) процедура не дает ответа.

Первый исход соответствует утвердительному ответу на поставленный вопрос. Второй исход соответствует ответу «для вывода недостаточно информации». Третий исход соответствует ответу «не знаю». В случае третьего исхода мы можем попытаться получить иной ответ на поставленный вопрос, либо увеличив время, отведенное системе на доказательство, либо выводя следование ~*Ф* из *А*, путем определения невыполнимости множества формул $A \cup$ *Ф*. В последнем случае опять возможны три исхода. Если установлена невыполнимость $A \cup$ *Ф*, т.е. ~*Ф* следует из *А*, на поставленный вопрос надо отвечать отрицательно. Выполнимость $A \cup$ *Ф*, как и прежде соответствует ответу «для вывода недостаточно информации». В связи с ограниченностью времени, отводимого на доказательство, и неразрешимостью исчисления предикатов, процедура доказательства может не дать ответа (третий исход). Часто одно из доказательств будет найдено, и ответ на вопрос получен.

Остановимся теперь на том, как извлечь из доказываемой теоремы информацию, являющуюся развернутым ответом на вопрос.

Для извлечения ответа требуется преобразовать граф опровержения с пустым дизъюнктом в корне в граф, называемый *модифицированным графом,* у которого в корне находится некоторое утверждение, могущее служить ответом. Преобразование состоит в том, что каждое предложение, возникающее из отрицания теоремы (часто называемой предположением), превращается в тавтологию (путем добавления к

нему отрицания этого предложения). Затем в соответствии с первоначальной структурой графа опровержения выполняются те же самые резолюции, что и раньше, до тех пор пока в корне не будет получено некоторое утверждение.

Так как при указанном преобразовании каждое предложение, возникающее из отрицания предположения, превращается в тавтологию, то модифицированный граф представляет собой доказательство того факта, что утверждение, расположенное в его корне, логически следует из аксиом и тавтологий. Поэтому оно также следует только из одних аксиом.

Приведем простейший пример, поясняющий принцип извлечения ответа.

П р и м е р 11.1. Пусть высказаны следующие утверждения: «Куб номер 2 находится всегда в том же месте, где куб номер 1» и «Куб номер 1 находится в месте *а»*. И задан вопрос: «Где находится куб номер 2?».

Указанные предложения после перевода их в исчисление предикатов примут вид аксиом:

$\forall x$ (В(Куб1, x)→В(Куб2, x)) аксиома 1

В (Куб1, *а)* аксиома 2

и предположения: $\exists x$В(Куб2, x), которое должно быть выведено из имеющихся фактов.

В приведенном примере предикату *В* придана интерпретация «находится в определенном месте».

На рис. 11.19 приведен граф опровержения для нашего примера.

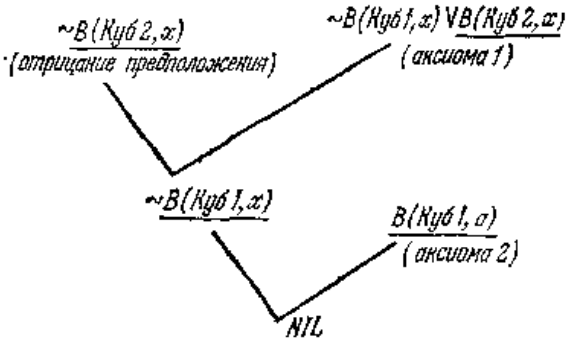


Рис. 11.19. Граф опровержения для примера 11.1.

Граф опровержения строится обычным путем.

Сначала отрицается формула, которую предстоит доказать. В рассматриваемом случае это приведет к формуле $\forall x$(~В (Куб2, *x)).* Затем это отрицание добавляется к множеству аксиом *(S),* и все члены этого расширенного множества преобразуются в форму предложений.
Далее с помощью принципа резолюции показывается, что это множество неудовлетворимо.
Будем выделять литеры, подвергающиеся унификации при образовании каждой резольвенты в графе опровержения. На рис. 11.19 эти литеры подчеркнуты. Подмножество литер в предложении, подвергающееся унификации в процессе построения графа опровержения, назовем *множеством унификации.* Каждая резольвента в модифицированном графе опирается на множества унификации, которые в точности соответствуют множествам унификации исходного графа опровержения.
Для извлечения ответа из этого графа построен модифицированный граф доказательства (рис.11.20).

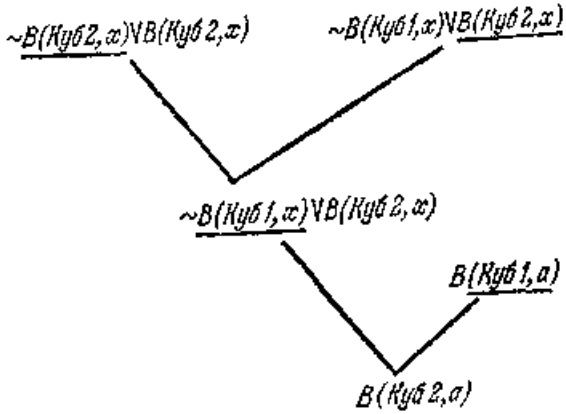


Рис. 11.20. Модифицированный граф доказательства для примера 11.1.

Смысл модификации графа состоит в том, чтобы извлечь тот частный случай переменной, относящейся к квантору существования, который служит ответом на вопрос. В нашем примере мы в качестве ответа получаем предложение В (Куб2, *а),* которое переводится в обычную форму исчисления предикатов (из формы предложений). Затем эта формула переводится в естественный язык, например, в виде предложения «Куб номер 2 находится в месте *а*».
Хотя указанный метод и прост, но у него есть несколько тонких моментов, которые мы разъясним на примерах.

Пример 11.2. Покажем, как преобразовать в тавтологию более сложные предложения, возникающие при отрицании предположения. Пусть в виде предложений дано следующее множество аксиом:

$$\sim A(x) \vee F(x) \vee G(f(x)),$$
$$\sim F(x) \vee B(x),$$
$$\sim F(x) \vee C(x),$$
$$\sim G(x) \vee B(x),$$
$$\sim G(x) \vee D(x),$$
$$A(g(x)) \vee F(h(x)).$$

Требуется доказать, исходя из этих аксиом, предположение
$(\exists x \exists y((B(x) \wedge C(x)) \vee (D(y) \wedge B(y))))$.
Отрицание предположения переводит его в два дизъюнкта:

$$\sim B(x) \vee \sim C(x),$$
$$\sim B(x) \vee \sim D(x).$$

На рис. 11.21 приведен граф опровержения для примера 11.2.

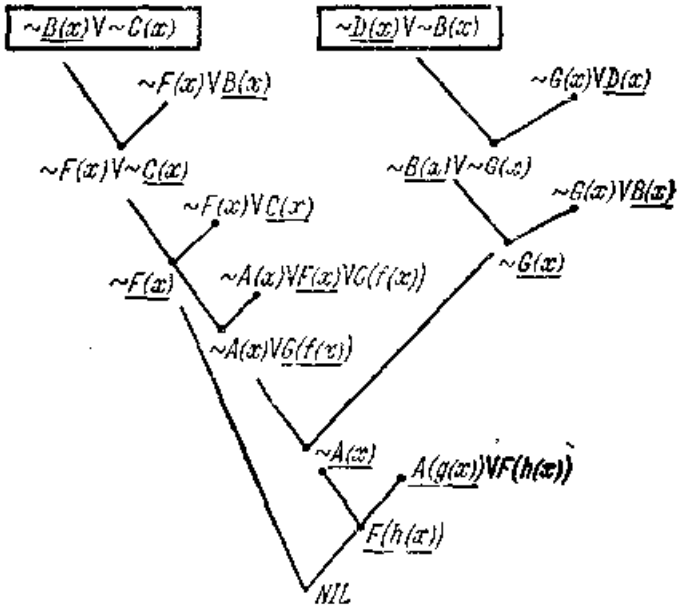


Рис. 11.21. Граф опровержения для примера 11.2.

Для преобразования графа следует превратить предложения, соответствующие отрицанию предположения, в тавтологии, добавив к ним их отрицания. Отрицания в данном случае (рис. 11.22) не являются предложениями (дизъюнктами), так как в них содержатся конъюнкции. Однако мы будем обращаться с этими конъюнкциями как с единой литерой и действовать формально, как будто наша формула является дизъюнктом. Мы можем себе это позволить, так как ни один из элементов рассматриваемой конъюнкции не может оказаться ни в одном множестве унификации.

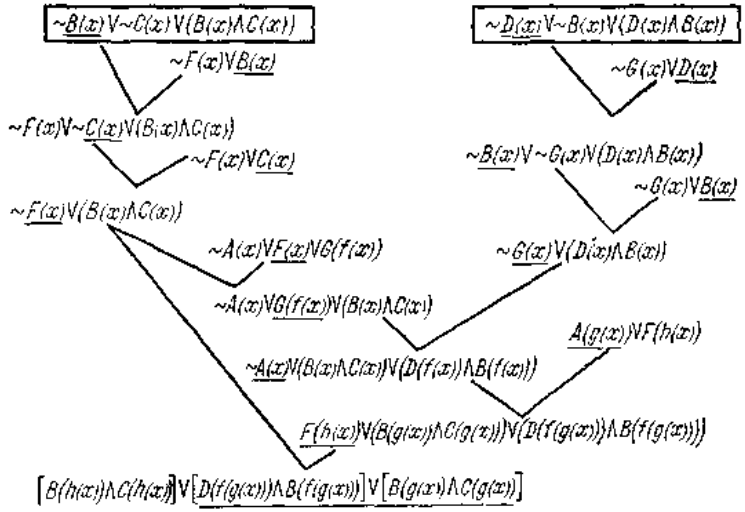


Рис. 11.22. Модифицированный граф доказательства для примера 11.2.

На рис. 11.22 приведен модифицированный граф. Мы видим, что ответное утверждение, расположенное в корневой вершине имеет форму, подобную форме предположения.

В общем случае, если предположение высказано в дизъюнктивной нормальной форме, то утверждение, получаемое в процессе извлечения ответа, представляет собой дизъюнкцию выражений, каждое из которых имеет форму либо всего предположения, либо одного или нескольких дизъюнктов этого предположения. Потому мы и говорим, что такое утверждение можно использовать в качестве «ответа» на вопрос, представляемый исходным предположением.

В случае, когда предположение, которое следует доказать, содержит переменные, относящиеся к квантору всеобщности, возникают дополнительные трудности. При отрицании такие переменные,

переходят в переменные, относящиеся к квантору существования, а это приводит к необходимости введения сколемовых функций. Возникает вопрос: «как интерпретировать эти функции, если они появляются в качестве термов в ответном утверждении»?.

Если мы будем применять описанный выше метод модификации к предположениям, содержащим квантор всеобщности, то мы будем получать ответы частного вида.

Пример 11.3. Пусть задана аксиома ($\forall x \forall u(P(x, u, x) \vee P(a, u, u))$) и требуется доказать предположение ($\exists w \ \forall v \ \exists y P(w, v, y)$).

На рис. 11.23, *а* представлен граф опровержения для примера 11.3. Граф доказательства, построенный по указанному ранее способу, приведен на рис. 11.23, *б*. Ответ, полученный в графе доказательства, имеет частный вид.

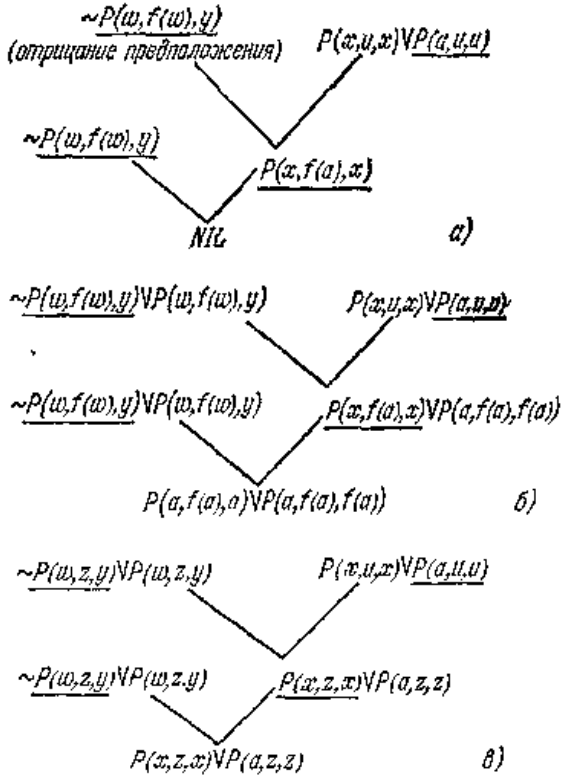


Рис. 11.23. Графы доказательств для примера 11.3.

Действительно, если из некоторого множества аксиом A мы можем доказать формулу $P(f(x))$, используя принцип резолюции, то это значит, что $A\rightarrow (\forall xP(f(x)))$ является теоремой. Если f не встречается в A, то формула $A\rightarrow (\forall f \forall xP(f(x)))$ также доказуема и, следовательно, $A\rightarrow(\forall zP(z))$ есть теорема. Применив указанные действия к ответу на рис. 11. 23, *б* получим $(\forall z(P(a, z, a) \vee P(a, z, z)))$. Однако и этот ответ не является наиболее полным. Можно показать, что в процессе извлечения ответа всегда можно заменять сколемовые функции, возникающие при отрицании предположения, новыми переменными.

В модифицированном доказательстве в эти новые переменные не будет делаться никаких подстановок, так что они пройдут через доказательство без изменения и появятся в окончательном ответном утверждении.

На рис. 11.23, *в* приведен соответствующий граф доказательства. Ответ имеет вид $(\forall x \forall z(P(x, z, x) \vee P(a, z, z)))$.

В заключение перечислим основные этапы извлечения ответа.

1. Построение графа опровержения и выделение в нем множества унификации.
2. Подстановка новых переменных на место сколемовых функций, появляющихся в предложениях, образованных из отрицания предположения.
3. Преобразование предложений, полученных из отрицания предположения, в тавтологии.
4. Построение модифицированного графа доказательства, следуя структуре исходного графа опровержения. При образовании каждой резольвенты модифицированного графа используется множество унификации, определяемое множеством унификации графа опровержения.
5. Предложение, находящееся в корневой вершине модифицированного графа, представляет собой ответное утверждение, извлеченное в ходе описываемого процесса.

Очевидно, что ответное утверждение, построенное предложенным способом, зависит от опровержения, из которого оно было извлечено. Для одной и той же задачи может существовать несколько различных опровержений, и обычно у нас нет возможности установить, будет ли ответное утверждение, построенное на его основе, наиболее общим. Эта трудность представляет, вероятно, лишь теоретический интерес, так как ответы, получаемые работающими программами, являются вполне удовлетворительными.

Итак, мы рассмотрели способ получения ответа в системах с общим выводом, теоретически не накладывающих ограничений на тематику диалога. Однако в связи с тем, что существующие методы семантической интерпретации требуют заранее задать множество предикатов и функций, определяющих тематику диалога, и в связи с ограниченностью времени и памяти, выделяемых на получение ответа, системы с общим выводом на практике не обладают универсальностью. Как следствие этого факта, системы с ограниченной логикой, не обладающие универсальным выводом, оказываются на практике конкурентоспособными по сравнению с системами с общим выводом. В следующем параграфе мы рассмотрим на примере системы с ограниченной логикой, введенной в п. 11.5.2, методы извлечения ответа из семантической сети.

11.6.2. Вывод ответа в системах с ограниченной логикой

Системы с ограниченной логикой не имеют универсальных процедур вывода, и их тематика задается при создании систем разработкой процедур, отвечающих на требуемые типы вопросов. Так, например, система, описанная в п. 11.5.2, отвечает на вопросы трех типов. В указанной системе базовые данные представлены в виде семантической сети, отражающей текущее состояние внутреннего мира системы, и в виде последовательности событий на временной оси, отражающей факты, сообщенные системе (рис. 11.15).

Для ответа в момент E_j на вопрос об отношениях, существовавших между узлами семантической сети в некоторый прошедший момент E_k, $k<j$, система должна вернуть сеть в состояние, соответствующее событию E_k. Возврат осуществляется путем движения по временной оси в направлении от события E_j к событию E_k. При этом, при прохождении через каждое событие E_i, $k \le i < j—1$, отношения, перечисленные в списке добавлений этого события, устраняются из семантической сети, а отношения, перечисленные в списке вычеркиваний этого события, добавляются к сети. После ответа на вопрос семантическая сеть возвращается в состояние, соответствующее событию E_j.

Приведем примеры типов вопросов, на которые умеет отвечать система. Примером первого типа вопросов является: КАКОЙ МУЖЧИНА КУПИЛ ПИРОЖНОЕ В БУЛОЧНОЙ? Ответ на этот вопрос приводит к исследованию временной оси. Для приведенного примера осуществляется поиск по временной оси в направлении прошлого до тех пор, пока не будет найдено событие, определяемое

каноническим глаголом ОБМЕНЯТЬ. При этом параметр МЕСТО должен иметь значение *БУЛОЧНАЯ*.

Когда такой узел найден, проверяется, является ли значением параметра ПОЛУЧЕННОЕ элемент из множества L-ПИРОЖНОЕ. Если условие выполняется, то проверяется, является ли значением параметра ПОКУПАТЕЛЬ элемент из множества L-МУЖЧИНА. Если и это условие выполняется, то узел, являющийся значением параметра ПОКУПАТЕЛЬ, принимается в качестве основы для ответа на вопрос. Ответ формируется либо на основании этого узла (это соответствует неполному ответу), либо на основании узла-события, являющегося родителем данного узла. Подробности формирования ответа приведены в п.11.7.

Примером вопроса второго типа является предложение: ЧТО БЫЛО СОБСТВЕННОСТЬЮ БУЛОЧНИКА ДО ТОГО, КАК ДЖОН КУПИЛ ЧТО-ТО В БУЛОЧНОЙ? В результате разбора предложение будет разделено на две части. Часть, соответствующая событию «ДЖОН КУПИЛ ЧТО-ТО В БУЛОЧНОЙ» обрабатывается аналогично вопросу первого типа, т. е. сеть возвращается в состояние, непосредственно предшествующее этому событию. Затем система ищет в полученной сети такое *x,* что для него справедливо отношение (СОБСТВЕННОСТЬ * БУЛОЧНИК * *x).* Если такое *x* обнаруживается, то оно используется при формировании ответа на вопрос.

Примером вопросов третьего типа является предложение «ЧТО ЯВЛЯЛОСЬ СОБСТВЕННОСТЬЮ БУЛОЧНИКА ДО ТОГО, КАК ПИРОЖНОЕ СТАЛО СОБСТВЕННОСТЬЮ ДЖОНА?». Для того чтобы ответить на вопрос этого типа, сеть возвращается назад до тех пор, пока не будет получено состояние, в котором истинны отношения (СОБСТВЕННОСТЬ * ДЖОН* *x)* и (ЭЛЕМЕНТ *x* L-ПИРОЖНОЕ). Затем производится движение по временной оси в прошлое до тех пор, пока одно из этих отношений не перестанет быть истинным, а отношение (СОБСТВЕННОСТЬ x БУЛОЧНИК * *x)* не становится истинным для некоторого *x.* Если такое *x* найдено, то оно используется для ответа на вопрос. На этом мы закончим рассмотрение методов получения ответов на вопросы. Напомним, что ответы, полученные на данном этапе, выражены во внутреннем представлении системы, и поэтому нашей очередной и заключительной задачей является перевод их во внешнее представление.

11.7. Формирование ответа в ограниченном естественном языке

Задача данного этапа заключается в переводе смысла некоторого высказывания, выраженного во внутреннем языке системы, в предложение, составленное по правилам выходного языка.

Методы, используемые на данном этапе, подобны методам на этапе синтаксического анализа с той только разницей, что здесь мы на основании грамматики порождаем предложение, а не распознаем его. В частности, представление в виде расширенных сетей переходов, используемое при синтаксическом анализе, с успехом применяется в большинстве систем при формировании ответа.

Поясним детали, возникающие при формировании ответа, на примере системы, приведенной в п. 11.5.2.

В результате работы этапа дедуктивного вывода, описанного в предыдущем параграфе, происходит выбор узла в семантической сети, определяющего смысл формируемого высказывания. Управление формированием ответа осуществляется на основе словарной информации и грамматики выходного языка, представленной в РАСП. В словаре у каждого канонического глагола есть свойство, значением которого является список, содержащий соответствующие ему поверхностные глаголы. Кроме того, каждый поверхностный глагол имеет *Ф*-правила (см. табл. 11.6), соотносящие глубинные структуры синтаксическим образам поверхностного глагола.

Проиллюстрируем эти свойства на примере сети, приведенной на рис. 11.16. Пусть выбран узел, отмеченный на рисунке буквой Б. Событие основывается на каноническом глаголе ОБМЕНЯТЬ, которому в словаре соответствуют несколько поверхностных глаголов (купить, продать, заплатить, стоить). Выбор одного из этих глаголов может быть осуществлен любым желаемым образом (возможен случайный выбор). Предположим, что выбран глагол КУПИТЬ. С глаголом «купить» в словаре связано два *Ф*-правила. Пусть выбрано первое правило

(BUYER ACTIVE THINGBT (FROM SELLER) (FOR THINGGIVEN))

(ПОКУПАТЕЛЬ АКТИВНЫЙ ПОЛУЧЕННОЕ (У ПРОДАВЦА) (ЗА ОТДАННОЕ)).

Это правило становится управляющим «предложением», которое должно быть разобрано генерирующей грамматикой.

Первый элемент в правиле указывает, что ему в соответствие должен быть поставлен узел, связанный с событием обменять (узел Б на рис.

11.16) глубинным отношением ПОКУПАТЕЛЬ. Этим узлом является *МЭРИ*. Итак, первым словом выходного предложения является МЭРИ. Второй элемент указывает на активный залог предложения. Пусть сформирован глагол КУПИЛ (BOUGHT). Третий элемент правила ПОЛУЧЕННОЕ указывает, что существительное должно быть сформировано на основании узла, удовлетворяющего отношению ПОЛУЧЕННОЕ в состоянии Б. Этому узлу соответствует слово ЛОДКА (THE BOAT).

Очередным элементом является (У ПРОДАВЦА). Наличие скобок в элементе указывает, что включение в выходную строку соответствующего ему существительного возможно, но не обязательно. В нашем случае результатом включения будет У ДЖОНА (FROM JOHN). Последний элемент в правиле (FOR THINGGIVEN) будет связан с узлом *ДОЛЛАРЫ* и приведет к генерации в выходной строке конструкции ЗА 50 ДОЛЛАРОВ (FOR 50 DOLLARS). Итак Ф-правило разобрано полностью, и на выходе получено предложение: MARY BOUGHT THE BOAT (FROM JOHN) (FOR 50 DOLLARS) (МЭРИ КУПИЛА ЛОДКУ (У ДЖОНА) (ЗА 50 ДОЛЛАРОВ)). Напомним, что конструкции в скобках не являются обязательными в выходном предложении. Не вдаваясь в детали, отметим, чго, выбрав в качестве поверхностного глагола PAY (ПЛАТИТЬ), мы бы сформировали предложение MARY PAID JOHN 50 DOLLARS FOR THE BOAT (МЭРИ ЗАПЛАТИЛА ДЖОНУ 50 ДОЛЛАРОВ ЗА ЛОДКУ).

Выбрав глагол COST (СТОИТЬ), мы бы получили предложение THE BOAT COST MARY 50 DOLLARS (ЛОДКА СТОИЛА МЭРИ 50 ДОЛЛАРОВ). В разговорной речи большинство ответов на вопросы не являются полными предложениями. Грамматика, основанная на РАСП, позволяет начинать работу с любого узла в РАСП, что приводит к возможности генерировать не обязательно все предложение, а например, только группу существительного.

Рассмотрим пример на рис. 11.16. Простейшим ответом на вопрос «WHO SOLD THE BOAT?» (КТО ПРОДАЛ ЛОДКУ?) является ответ «ДЖОН».

Система работает таким образом, что если в качестве узла, на базе которого строится выходное предложение, выбран узел на временной оси, то ответ является полным предложением. **Если выбран узел в сети, соответствующий существительному (например, *ВАГОН*, *ДЖОН* на рис. 11.14), то это существительное может быть использовано для получения неполного ответа.** Узлу *ВАГОН* соответствует конкретный объект в реальном мире, известный как «вагон» (точнее множество из пятнадцати вагонов), с которым связаны

значения: СТАРЫЙ, 15, КРАСНЫЙ, МАЛЕНЬКИЙ, являющиеся свойствами; ВОЗРАСТ, ЧИСЛО, ЦВЕТ, РАЗМЕР. Утверждение об узле *ВАГОН* во многом подобно узлу-событию. Но между этими узлами существует важное различие: **события происходят, совершаются, а утверждения являются статическими, неизменными до тех пор, пока не произойдет событие, изменяющее их истинность**.
Следует отметить, что случайная генерация модификаторов слова *ВАГОН* приводит к построению неправильных конструкций типа: КРАСНЫЙ, 15 ВАГОНОВ. Необходимо осуществить упорядочение модификаторов. Это может быть выполнено управляющей строкой словаря, руководящей генерацией группы существительного способом, аналогичным используемому при управлении генерацией предложения. Приемлемой строкой управления является (ЧИСЛО, РАЗМЕР, ВОЗРАСТ, ЦВЕТ).
В заключение отметим, что использование в качестве внутреннего представления исчисления предикатов не оказывает существенного влияния на изложенный метод формирования ответа. Справедливость этого замечания становится очевидной, если однозначно сопоставить каждой формуле исчисления предикатов некоторый граф.

11.8. Компьютеризация науки, ее проблемы и следствия

Развитие современных теорий предполагает анализ и осмысление фундаментальных изменений, происходящих в науке, культуре и образовании в связи с широким внедрением компьютерных технологий и персональных компьютеров. Обращение к этой проблеме будет осуществлено лишь в той мере, в какой позволит рассмотреть новые возможности изучения знания и «знания о знании», а также выявить новые способы описания присутствия человека и социокультурной составляющей в разных формах «представления знания». Реализовать это возможно, опираясь на исследования в области когнитивной науки, где **знание и информация являются главным предметом**. Представление знания, как оно исследуется в когнитивной науке, не только предполагает предметное его содержание, но и определяет интерпретативную деятельность субъекта, социокультурную обусловленность его знания и поведения, а также фиксирует другие связи и отношения, представленные в традиционных эпистемологических структурах лишь опосредованно.

11.8.1. Эпистемология и когнитивная наука

Когнитивная наука (когнитология) сформировалась в 60-70-х годах XX века (Гарвард, США) в качестве дисциплины, исследующей методом компьютерного моделирования функционирование знаний в интеллектуальных системах. Когнитивную науку отличают междисциплинарность, использование компьютерной метафоры и исследование познания. Центральным для всей проблематики когнитивной науки является обращение к компьютеру, служащему самой наглядной и самой убедительной моделью того, как формируется, структурируется и «работает» знание, а также имитируются различные когнитивные процессы (например, обучения или получения экспертного знания и т. п.). Феномен знания исследуется в аспектах его получения, хранения, переработки, коммуницированния (передачи), выясняется, какими типами знания и в какой форме обладает человек, как «представлено» знание в его голове и как он его использует.

Важную роль играет лингвистика, которая выступает для когнитивной науки как важный источник материала об устройстве когнитивных структур. По отношению к «искусственному интеллекту» (ИИ) когнитология является своего рода «теорией интеллектуальных машин и механизмов», т. е. сконструированных человеком компьютерных устройств и лишь через них — их естественных прообразов - людей познающих. Это объясняет различную природу эксперимента в психологии и когнитивной науке и определяет существование в последней компьютерной метафоры.

Традиционные проблемы гносеологии, эпистемологии, философии и методологии науки получили новое видение и интерпретацию. «Когнитивизм знаменовал появление новой парадигмы научного знания, и с ним в историю науки пришло новое понимание того, как следует изучать знание, как можно подойти к проблеме непосредственно не наблюдаемого — прежде всего к проблеме внутреннего представления мира в голове человека...» (Кубрякова Е.С., Демьянков В.З. и др. Краткий словарь когнитивных терминов. М., 1996. С. 61). Эксплицитно выраженные знания составляют лишь незначительную часть общей базы знаний человека. Согласно современным подходам, такая база есть самоорганизующаяся и саморегулируемая система. Она включает следующие компоненты:
- языковые знания — грамматика (с фонетикой и фонологией),

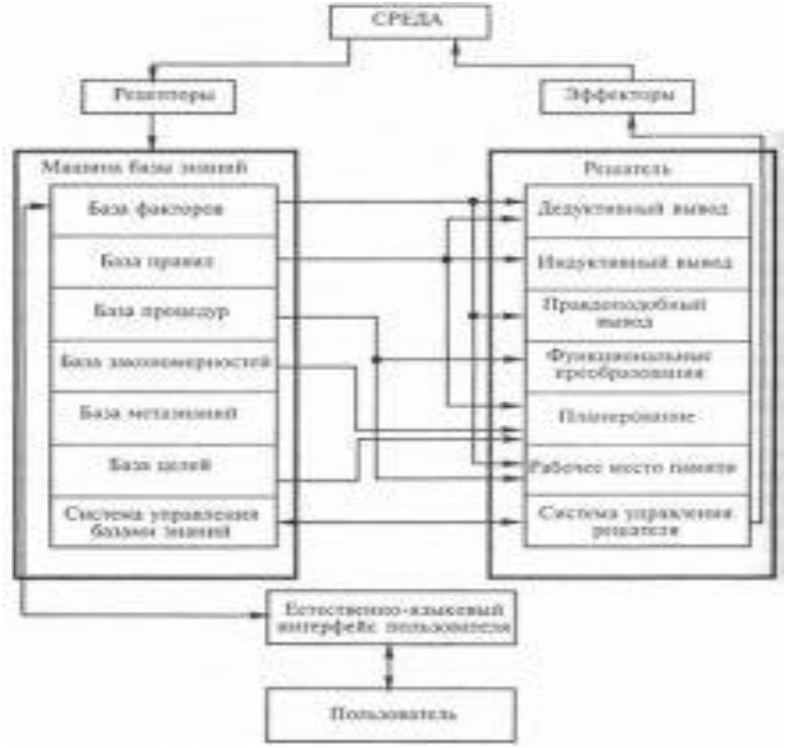
дополненная знанием композиционной и лексической семантики;
- знание об употреблении языка;
- знание принципов речевого обучения;
- внеязыковые знания — о контексте описываемой ситуации, об адресате коммуникации (в том числе знание о поставленных адресатом целях и планах, его представления о говорящем, об окружающей обстановке, знание своих умений);
- общефоновое знание, т. е. личностная картина мира (Осуга С. Обработка знаний. М., 1989. С. 9-10).

При соотнесении эпистемологии и когнитивной науки необходимо различать **знание и информацию**, что упрощенно можно свести к формуле: **информация — это знание минус человек; информация — знаковая оболочка знания.** Под **компьютерным представлением знания** **принято понимать информацию, хранимую в машине, формализованную в соответствии с определенными структурными правилами, которые компьютер может автономно использовать при решении проблем с помощью заложенных в нем алгоритмов типа логического вывода.** Информационная модель знания (как записанная в компьютере, так и вербализованная в тексте) является лишь намеком на представленное знание, по которому человек способен творчески воссоздать само знание. **Следует отметить принципиальное отличие той информации, которая служит для получения знаний человеком, от информации, изучаемой в теории информации**. Когнитивное знание открывает человеку дополнительные возможности размышления и действия, увеличивает его свободу. **Информация как управляющий сигнал уменьшает неопределенность допускаемых состояний управляемой системы**. Знание — личное достояние знающих, перенимающих его друг у друга как образцы действия в процессах познания. Этого нельзя сказать об информации, которая в противоположность знанию не является достоянием конкретной личности, она равно доступна всем, хотя возможности превратить ее в знание у каждого свои, опирающиеся на личный опыт и способности.

Такое различие создается исключительно присутствием человека, способного извлечь из информации, записанной на бумаге или закодированной в компьютере, нечто, позволяющее реализовать человеческую свободу выбора. Пользователь получает представление о ряде возможных точек зрения, соответственно, возникает та самая неопределенность, которая является необходимой предпосылкой для выбора. Вечная философская тема — диалектика свободы и

необходимости — специфически проявилась в исследованиях по представлению знаний. Для правильного понимания свободы важно выявить ее связь с объективной неопределенностью развития. Мысль о том, что свобода — это осознанная необходимость, была высказана Спинозой еще в XVII веке, принимавшим концепцию абсолютного и однозначного детерминизма. Свобода человека оказывается здесь иллюзорной: ни человек, ни общество в целом не имеют свободы выбора действий и поступков, а лишь осознают предопределенный ход событий. Действительная свобода возможна лишь при наличии объективной неопределенности, когда принятое решение, понимание, осознание происходящего могут изменить ход событий, сами детерминируют социальный процесс. В этом случае нельзя спрятаться за некой безличной необходимостью, свобода объективно соотносится с этическими «параметрами» — нравственной ответственностью человека, осуществляющего выбор и принимающего решение. В когнитивных науках, таким образом, становится необходимой этическая рефлексия, основанная на принципе свободы личности.

Отличие традиционной гносеологии от разделов теории познания, имеющих дело с использованием компьютеров, состоит в том, что первая концентрируется на процедуре описания, обращаясь к высказываниям и правилам для получения знания. **«Компьютерная» теория познания делает центром своего внимания регуляцию, обращается к нормативным предложениям, использует знания для продуцирования правил.** Сегодня развитие теории познания классическими гносеологическими средствами не всегда возможно, изменяются инструментарий гносеолога, требования к его профессиональной подготовке. Философия становится дисциплиной, сопричастной экспериментальной деятельности, осуществляемой при разработке программ искусственного интеллекта. Выяснилось, что именно в этой сфере возможна проверка самых тонких и абстрактных гипотез о природе человеческого разума. Сегодня здесь на первый план вышла проблема порождения знания, и это потребовало пересмотра базовых концепций ИИ. По ходу поиска обнаружилось, что идеи Локка Лейбница, Канта, Гуссерля, Хайдеггера — это концептуальные модели, которые могут быть «экспериментально» проверены в рамках программы ИИ, что позволяет по-новому решать философские споры о природе разума и познания. Так, по существу, экспериментально была доказана несостоятельность представления Локка о душе как «чистой доске» (tabula rasa), на которой только опыт записывает какое-либо содержание. Выяснилось, что возможности



универсальных распознающих устройств, не имеющих тех эмпирических знаний-предпосылок, которыми обладает любой человек, ограничены. На самом деле «чистая доска», на которую записываются данные опыта, — те «образы», которые необходимо распознать иногда только по намеку, как это делает человек, — должна иметь весьма сложную структуру, включающую множество априорных знаний о мире. При разработке программ ИИ экспериментально подтвердилась также огромная роль скрытых, неявных знаний, не выраженных в языке, но хранящих в себе жизненный опыт. **В литературе достаточно определенно высказывается мнение о том, что в настоящее время эксперименты на компьютерах, а не на людях — самый верный шаг на пути проверки гипотез о мышлении и познании, различных аналогов мыслительной деятельности.**

Такие эксперименты, разумеется, не могут рассматриваться в качестве полного доказательства предлагаемых гипотез, но принимаются как

серьезный аргумент в их пользу. Программа искусственного интеллекта как своего рода «экспериментальная философия» делает фигуру эпистемолога столь же необходимой для компьютерной эпохи, как и математика-программиста. Эпистемология впервые за всю историю получает прямой выход в сферу конструктивной инженерной и технологической деятельности. Меняется характер связи эпистемологии с практикой.

12. Методы и средства образования фреймов

12.1. Метод формального образования фреймов

В процессе образования фреймов выделим два этапа: инфологический и даталогический.

На **инфологическом этапе** образования фреймов рассматриваются вопросы, связанные со смысловым содержанием фреймов. **Даталогический этап** образования фреймов направлен на решение вопросов представления фреймов в памяти ЭВМ. Даталогическое образование фреймов, в свою очередь, будем подразделять на **логическое** и **физическое**.

Таким образом, процесс образования фреймов и базы фреймов (БФ) представляет собой сложный многоуровневый процесс, охватывающий все аспекты их использования: от удобства обращения к базе фреймов пользователей до конкретного представления фреймов в ЭВМ.

Весь процесс образования фреймов можно представить в виде последовательности операций, начинающихся с описания признаков и свойств объктов (истин) предметной области и заканчивающихся схемой внутренней модели бызы фреймов.

На первом этапе образования фреймов (инфологическом) изучается предметная область и проводится ее описание. С этой целью идентифицируются все типы истин (сущностей, предметов), представляющие интерес для введения в БФ, определяются связи между этими истинами, а также выявляются ограничения. Для описания предметной области будем используют концепцию моделей фреймов: «истина —связь» или «инфологическую модель фрейма». Эти описания позволяют обеспечить взаимодействие между пользователями и лицами, опразовывающими фреймы. Образование фреймов начинается с предварительной структуризации предметной

области. Обычно для облегчения этого процесса составляется перечень вопросов, на которые требуется ответить:
какие типы истин входят в состав предметной области?
каковы имена каждого типа истины?
каково значение (семантика) каждого типа истины?
какими признаками и свойствами обладает каждый тип истины?
какие атрибуты истин представляют интерес?
каковы имена каждого атрибута?
На основе собранной информации о типах объектов (истин) выявляются типы существующих связей для систематизации собираемой информации, а также предлагается использовать вопросы следующего плана:
какие типы связей (отношений) могут иметь место между каждой парой типов объектов (истин)?
каковы имена каждого типа связи?
каково значение каждого типа связи?
Собранную информацию оформляют в виде специальных диаграмм (фреймов). Для обозначения объектов (истин) можно использовать прямоугольники, а для атрибутов — овалы, соединяя их с соответствующими объектами (истинами) ненаправленными ребрами. Для обозначения связей можно использовать ромбы, которые также соединяют их с объектами (истининами) ребрами.
Пример. Рассмотрим графическую диаграмму (фрейм), соответствующую описанию предметной области производства телевизоров (рис. 12.1).

Рис. 12.1. Пример графической диаграммы (фрейма)

При создании описания предметной области разработчик фрейма разбивает ее на ряд локальных областей, которые потом объединяются. Выбор размеров локальных областей в общем случае является произвольным. Но для удобства образования фрейма в одной локалькой области рекомендуют использовать не более 6—7 объектов (истин).

Моделирование локальных представлений. Для представления информации в модели «истина-связь» конструктивными элементами модели являются: *истина, атрибуты (признаки, свойства)* и *связи.*

Основным элементом локального представления некоторого явления, процесса или объекта предметной области (фрейма), о котором необходимо собрать информацию, является *истина.* При формулировании истин следует различать такие понятия, как **тип** и **экземпляр.**

Понятие **тип истины** относится к набору однородных предметов или явлений, выступающему как целое. **Экземпляр истины** относится к конкретному элементу набора. Например, типом истины может быть ТЕЛЕВИЗОР, а экземпляр истины— РУБИН. На концептуальном этапе образования фрейма необходимо сформулировать истины, требуемые для описания локального представления фрейма. При этом возникает проблема ее выделения в качестве конструктивного элемента, так как некоторая информация может быть представлена как атрибут, истина или связь. Например, тот факт, что конкретный студент учится в университете, может быть выражен истиной СТУДЕНТ, либо связью УЧИТСЯ между истиной СТУДЕНТ и УНИВЕРСИТЕТ, либо как атрибут в истине ГРУППА УНИВЕРСИТЕТА.

При возникновении такой неоднозначности образования фрейма сущностей (истин) рекомендуется руководствоваться следующими правилами:

1. Необходимо выбирать вариант, более гибкий с точки зрения представления информации, т. е. позволяющий представлять не только всю часть некоторой информации, но и ее отдельные фрагменты (кванты).
2. Для моделирования порции (кванта) информации должна использоваться одна и только одна конструкция. Другими словами, следует избегать избыточности в использовании конструктивных элементов.

Другое важное положение, связанное с формулированием истин, касается **выбора наименованийистины**. Так как она представляет собой информационный факт (квант), то этому факту должно быть дано четкое наименование, что имеет важное значение для стадии объединения локальных представлений.

Выбор атрибута истины. Свойства истин определяются с помощью атрибутов.

Атрибут — это характеристика истины, имеющая имя. Атрибуты используются для определения того, какая информация должна быть собрана об истине при образовании фреймов. Примерами атрибутов для истины СТУДЕНТ могут быть: НОМЕР ЗАЧЕТНОЙ КНИЖКИ, ПОЛ, НОМЕР ГРУППЫ и т. д.

Несмотря на то что совокупность атрибутов не может служить основой для выделения истин, из множества атрибутов обычно выделяют несколько (или один) атрибутов, позволяющих однозначно распознавать экземпляр истины.

Атрибут (или совокупность атрибутов), значение которого единственным образом определяют экземпляр истины, будем называть **ключом.** Если для описания типа истин выбрана совокупность атрибутов, не содержащих ключа, то создается специальный атрибут, играющий роль ключа. В общем случае понятие может иметь несколько ключей.

Таким образом, атрибуты могут быть разделены на два класса: те, которые служат для идентификации экземпляров истины, т. е. являются ключами, и те, которые описывают свойства истин.

На этапе построения локальных представлений в процессе выбора атрибутов рекомендуется каждому ставить в соответствие следующие характеристики:

наименование, т. е. уникальное обозначение атрибута;

описание — словесное изложение смысла атрибута;

роль, т. е. конкретное использование атрибута.

Спецификация связей. После выделения истин, характеризующих предметную область, и соответствующих атрибутов локальное представление дополняется информацией, раскрывающей зависимости между экземплярами истин.

Одна из неформальных процедур для этого шага заключается в попарном объединении между собой всех экземпляров истин, входящих в рассматриваемое локальное представление, и установлении существования некоторой связи для каждой пары истин.

После их выявления определяются связи необходимые и избыточные. Каждой необходимой связи присваивается имя и определяются ее характеристики, которые включают тип связи (1 : 1, 1 :M, *M:M,M:*1*).*

Объединение моделей локальных представлений. В результате объединения локальных представлений получается единая глобальная информационная структура. Объединение может быть осуществлено на базе трех основополагающих подходов: **идентичности, композиции и обобщении**.

Идентичность позволяет объединять несколько истин путем объединения двух или более элементов синонимами.

Для проверки согласованности результата объединения локальных представлений на основе понятия идентичности предлагается следующее правило:

Если объект (истина) из одного локального представления идентичен объекту (истине) из другого представления, ни один из этих объектов

(истин) не должен в дальнейшем принимать участие в каком-либо другом объединении идентичности между этими двумя представлениями.

В нескольких локальных представлениях рассматривается один и тот же объект, но его отдельные составляющие могут различаться. Например, имеется два локальных представления ТЕЛЕВИЗОР (рис. 12.2, *а*).

Рис. 12.2. Объединение идентичности

В результате объединения идентичности вместо отдельных локальных представлений будет построено новое (рис. 12.2, *б)*.

Композиция (агрегация) позволяет рассматривать связь между элементами модели как новый элемент. Например, истина ФАКУЛЬТЕТ может быть рассмотрена как композиция истин КАФЕДРА, ДЕКАНАТ.

При объединении представлений композиция встречается в следующих двух формах.

1. В одном представлении композиционный объект (истина) определяется как целое, а в другом — в виде составных частей.

Например, в одном локальном представлении определены в качестве истины объект ТЕЛЕВИЗОР, а в другом — блоки: КИНЕСКОП, БЛОК ЯРКОСТИ, БЛОК РАЗВЕРТКИ, являющиеся составными частями объекта ТЕЛЕВИЗОР. Причем во втором представлении не указан явно тот факт, что вышеперечисленные блоки — составные части телевизора.

Простое объединение позволяет слить эти два локальных представления, не выражая явным образом, что ТЕЛЕВИЗОР является композицией частей КИНЕСКОП, БЛОК ЯРКОСТИ, БЛОК РАЗВЕРТКИ. Чтобы включить эту информацию в модель объединенного представления, необходимо выполнять объединение с использованием композиции.

2. Композиционный объект в одном локальном представлении до конца как единое целое не определен.

Например, в одном представлении определены КИНЕСКОП и БЛОК ЯРКОСТИ, а в другом БЛОК РАЗВЕРТКИ, БЛОК СИНХРОНИЗАЦИИ, БЛОК ЦВЕТНОСТИ, являющиеся составными частями объекта ТЕЛЕВИЗОР, который не назван ни в одном представлении. Для повышения возможностей совместного использования фреймов можно ввести в рассмотрение композицию ТЕЛЕВИЗОР (рис. 12. 3).

Рис. 12.3. Объединение композиции

Понятие **обобщения** близко к понятию композиции, но в отличие от последней, которая может быть представлена в виде составных частей, образующий некоторое «целое», **обобщение связано только с «целыми». Обобщение относится к типу абстракции, в которой группа подобных элементов воспринимается как родовой элемент**. При этом различия между отдельными элементами опускаются.

Например, УЧАЩИЙСЯ может быть воспринят как УЧАЩИЙСЯ школы, УЧАЩИЙСЯ ПТУ, УЧАЩИЙСЯ техникума.

Так же как и композиция, обобщение может встречаться в двух формах:

1. В одном локальном представлении определено некоторое множество объектов, которое может быть объединено общим для этих объектов родовым понятием, а само оно указано в другом локальном представлении.

Пример.

I представление	II представление
Цветные телевизоры	Телевизоры
Черно-белые телевизоры	

Здесь родовым понятием, объединяющим оба представления, будет ТЕЛЕВИЗОР.

2. Ни одно из объединяемых локальных представлений не содержит родового понятия.

I представление	II представление
Цветные телевизоры	Переносные телевизоры
Черно-белые телевизоры	

В этом случае установить наличие родовой связи между специфичными типами объектов можно только в процессе сопоставления объектов из различных локальных представлений.

Использование объединения обобщением позволяет повысить эффективность доступа пользователей к фреймам, хранящимся в БФ.

При формировании глобального представления фреймов путем комбинированного использования идентичности, композиции и обобщения можно отобразить в модели сложные связи, существующие между объектами и понятиями в предметной области.

Сам процесс объединения локальных представлений обычно носит интерактивный характер. В целях упрощения процесса объединения обычно осуществляют бинарное объединение, т. е. на каждом этапе объединяются только два локальных представления. Процесс попарного объединения повторяется для всех локальных представлений до тех пор, пока все они не будут интегрированы в одно глобальное (рис. 12.4).

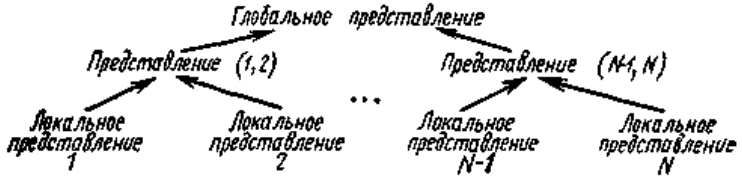


Рис. 12.4. Бинарное объединение

В процессе объединения могут выявляться противоречия между отдельными локальными представлениями. Противоречия обусловлены обычно неполнотой или ошибочностью спецификаций или же некорректностью требования. Так, например, в одном локальном представлении связь между объектами может быть отнесена к типу 1 : 1, а в другом — типу 1 *:M* или *M:*1. Большинство из противоречий такого вида может быть разрешено на этапе объединения путем выполнения соответствующих коррекций. Процесс объединения требует согласования и устранения всех выявленных противоречий.

После завершения объединения полученное глобальное представление служит исходной информацией для даталогического этапа проектирования БФ. Данный этап предусматривает полученное инфологическое описание предметной области отобразить в описание БФ.

Отображение одного описания в другое будет зависеть от принятой за основу модели фреймов, поддерживаемой соответствующей СУБФ.

СУБФ по виду поддерживаемой модели могут быть отнесены к одному из трех классов: реляционные, сетевые, иерархические.

В случае реляционной СУБФ описание фреймов предметной области трансформируется в реляционную схему. Объекты при этом отображаются в отношения БФ.

В случае сетевой СУБФ описание предметной области отображается в граф. Типы объектов (истин) представляются в типы записей, а типы связей — в типы наборов.

В случае иерархической СУБФ описание предметной области преобразуется в множество деревьев. Типы истин при этом также будут отображены в типы записей, а типы связей — в типы «исходный-порожденный».

Процесс этих преобразований является неформальным и во многом зависит от разработчика.

Принципы проектирования физической базы фреймов. Физическая организация фреймов оказывает существенное влияние на эксплуатационные характеристики проектируемой БФ, такие, как объем занимаемой памяти, время отклика базы на запрос пользователя и т. д.

Под проектированием физической БФ будем понимать <u>процесс</u> создания <u>эффективной ее структуры</u> на выбранной логической структуре.

Физическая база фреймов представляет собой совокупность совместно хранимых взаимосвязанных фреймов, состоящих из одного или нескольких типов хранимых записей.

Понятие структуры физической БФ включает: формат хранимой записи, структуру путей доступа к фреймам и размещение записей на физических устройствах.

Наиболее простой формой хранения фреймов в памяти ЭВМ является линейный список. **Линейный список** представляет собой конечное и упорядоченное множество объектов (параметров) $\{x[1], x[2], ..., x[n]\}$, структурные свойства которого связаны только с линейными относительным расположением элементов фреймов. Порядковый номер, расположенный в квадратных скобках, указывает на относительное положение параметров в списке.

Линейные списки используются в тех случаях, когда встречаются упорядоченные множества фреймов переменного размера и где операции включения, поиска, удаления элемента фрейма должны выполняться в произвольных местах.

Одномерный линейный список, используемый для хранения фреймов в памяти ЭВМ, называют также **вектором фреймов.**

Для линейного списка существует две возможности представления в ЭВМ: последовательное и связанное.

Последовательное представление. Является более простым и предполагает, что элементы списка размещаются в последовательных элементах памяти ЭВМ (рис.12. 5).

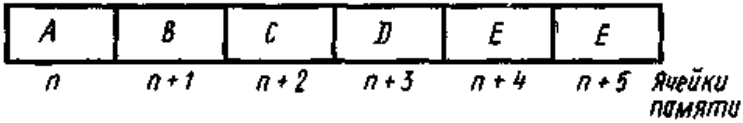


Рис. 12.5. Последовательное распределение памяти для представления линейного списка

При таком представлении списка возникают определенные сложности в реализации вставки нового элемента в середину. Например, чтобы включить в список между элементами *D* и *E* новый элемент *K,* необходимо изменить место элементов *E* и *F*. Точно так же удаление элемента из списка ведет к появлению в списке пустой ячейки и для его уплотнения необходимо осуществлять смещение оставшихся элементов.

Связное представление. Оно предусматривает задание для каждого элемента списка отношений следования и предшествования с помощью указателей, задающих связь между фреймами. При таком представлении каждая ячейка содержит элемент фрейма и указатель (адрес) на последующий элемент списка. При связанном распределении не требуется, чтобы список хранился в последовательных элементах памяти (рис. 12.6).

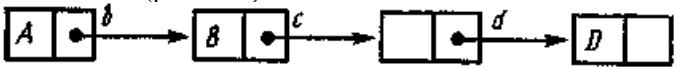


Рис. 12.6. Связанный список

Добавление или исключение некоторых фреймов в этом случае можно выполнить с помощью простой операции изменения значения указателя (рис. 12.7).

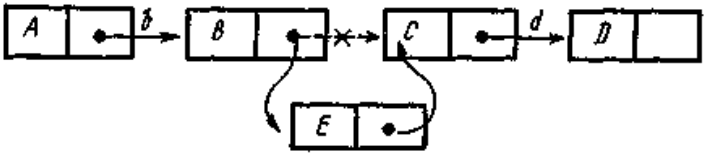


Рис. 12.7. Пример включения элемента в связанный список

Таким образом, использование связанных списков более удобно в случае динамически изменяющихся линейных структур.

Списки могут быть и двусвязанными (рис. 12.8).

Рис. 12.8. Двусвязанный список

При таком задании списка необходимо кроме прямого указателя для каждого элемента вводить в рассмотрение и обратный.

Структура линейного списка, представленная с помощью связанного распределения, называется также цепной структурой или цепью. Физическая последовательная и связанная структуры являются основными для большинства числа методов доступа к фреймам.

Методы доступа. Под методом доступа понимается совокупность технических и программных средств, обеспечивающих возможность хранения и выборки фреймов, расположенных на физических устройствах ЭВМ. В методе доступа выделяют два компонента: структура памяти и механизм поиска.

Наиболее широко используемыми методами доступа являются: последовательный; прямой произвольный; индексно-последовательный; индексно-прямой; основанный на использовании явных древовидных структур.

Последовательный доступ. Он реализует доступ к базе фреймов путем последовательного просмотра записей.

Рассмотрим случай, когда физически последовательная структура содержит несколько смежных записей. Записи могут быть неупорядочены или упорядочены по значениям первичного ключа. Обычно каждая запись располагается в отдельном блоке.

Среднее количество физических блоков $N_{ср}$, к которым осуществляется доступ при поиске произвольной записи, равно $N_{ср}= (1+N)/2$. Данное выражение справедливо как для упорядоченных, так и неупорядоченных записей при условии, что искомая запись существует. В случае если искомая запись отсутствует, то для неупорядоченного файла $N_{ср}=N$, т. е. будут проверены все записи. Таким образом, неупорядоченный файл является неэффективным, если приходится часто обращаться к поиску отсутствующей записи.

Поскольку внесение изменений в произвольном порядке в последовательную структуру требует большого количества операций по перемещению записей, режим внесения изменений строго ограничивают.

Прямой доступ. В том случае, когда имеется возможность выделить в памяти для каждой записи место, определяемое уникальным значением ее первичного ключа, можно построить простую функцию преобразования ключа в адрес, обеспечивающую запоминание и

выборку каждой записи в точности за один произвольный доступ к блоку.

Использование прямого доступа позволяет эффективно с точки зрения временных затрат осуществлять поиск понятий в библиотеке.

Методы прямого доступа подразделяются на две группы:

1. Доступ с помощью ключа, эквивалентного адресу;
2. Хэширование (метод произвольного доступа или рассеянной памяти).

Использование первой группы методов доступа возможно в тех случаях, когда в качестве атрибута в запись включается адрес памяти, в котором будет размещена запись. При работе с БФ этот атрибут будет использоваться в качестве ключа.

Методы второй группы широко используются для обеспечения быстрой выборки и обновления записей по заданному значению первичного ключа.

Основная идея хэширования заключается в том, что каждый экземпляр записи размещается в памяти по адресу, вычисляемому с помощью специальной **хэш-функции**. В этом случае экземпляры записей хранятся не в последовательных ячейках, выделенного блока памяти, а случайным образом рассеиваются по всему блоку. Причем каждая новая запись помещается в блок памяти таким образом, что ее присутствие или отсутствие может быть установлено без поиска по всему блоку.

Другими словами, ***в методе хэширования строится <u>отображение</u> множества <u>ключевых значений экземпляров</u> записей фреймов во множество <u>физических адресов</u>.*** В качестве хэш-функций обычно выбирается некоторое правило, преобразующее значение типа записи в адрес.

Рассмотрим на примере алгоритм построения хэш-функции.

Пусть в памяти машины для хранения экземпляров записей выделен блок *B* из 1024 слов. Экземплярами записей z_i, которые необходимо хранить в *B,* являются наборы символов длиной в два слова. Таким образом, в блоке *B* можно разместить 512 отличающихся записей z_i.

Предположим, что начальный адрес блока в памяти α. Хэш-адресом для данного случая может быть цепочка I_z из девяти битов, поскольку формула $\alpha+2I_z$ будет давать адрес, попадающий внутрь блока. Цепочка I_z может быть вычислена для каждой записи длиной в два слова по следующему алгоритму. Пусть z_i хранится в словах *a* и *b*. Тогда:

1. Умножим *a* на *b* и получим в результате *c* (произведение занимает два слова).
2. Сложим два слова, составляющие *c,* получим в результате *d.*
3. Возводим *d* в квадрат, получим в результате *e.*

4. Извлечем девять центральных битов из e и примем, что это и есть искомое значение I_z.

В зависимости от свойств битовых цепочек, представляющих экземпляры записей или их ключей, могут использоваться различные хэш-функции. При их подборе важно, чтобы хэш-функция равномерно распределяла все множество z_i по выделенному блоку памяти.

Обычно выбранная хэш-функция не может быть полной гарантией того, что различные экземпляры записей получат различные хэш-адреса. Всегда в процессе хэширования возникает ситуация, когда два (или более) экземпляра записей получают один и тот же адрес, что приводит к **коллизии.** Коллизия возникает, когда при добавлении новой записи в блоке памяти выясняется, что позиция записи, задаваемая хэш-адресом, уже занята записью, отличной от той, которую собираются туда поместить.

Для разрешения коллизий имеется много методов. К ним относятся методы последовательного сканирования и цепочки.

Метод последовательного сканирования. Он состоит в том, что при возникновении коллизии просматриваются последовательно (сканируются) участки памяти, отведенные для хранения записей, до тех пор, пока не будет найдена свободная позиция, куда и помещается запись.

Метод цепочки. Вместо хранения самих элементов блока можно хранить в нем указатели на связанные списки экземпляров, записей, имеющих одинаковый хэш-адрес.

После хэширования экземпляра записи (ее ключа), если участок памяти по вычисленному адресу свободен, запись размещается по этому адресу. Если же участок памяти по вычисленному адресу занят, то происходит обращение по указанию к следующему участку памяти (элементу списка), на который и помещается запись.

При поиске записей действия выполняются в той же последовательности. Вначале проверяется участок памяти по вычисленному адресу. Если там находится запись с другим значением ключа, то по указателю обращаются к следующей записи, и так до тех пор, пока не будет найдена необходимая запись.

Память, выделяемую для организации списков, называют **областью переполнения**.

Индексно-последовательный метод доступа. Он строится на основе упорядоченного физически последовательного файла и иерархической структуры индексов блоков, каждый из которых упорядочен по значениям первичных ключей подобно записям в файле данных.

Данный метод позволяет обеспечивать как последовательный, так и произвольный доступ к данным. С этой целью в рассмотрение

вводится новый параметр — индекс блока. **Индекс блока** — представляет собой упорядоченную таблицу значений первичных ключей, в которой каждый элемент блока содержит наибольшее значение ключа среди всех записей в указанном блоке. С каждым значением ключа в индексе связан указатель соответствующего блока (рис. 12.9).

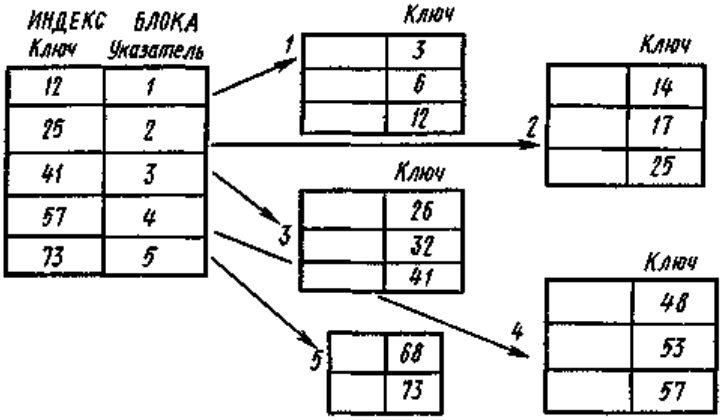


Рис. 12.9. Организация индекса блока

Использование упорядоченного индекса блока требует таких же записей данных в файле. Благодаря этой упорядоченности можно в каждом элементе индекса указать границы соответствующего блока: значение ключа представляет верхнюю границу, а указатель задает нижнюю границу, так как указывает адрес блока, точнее, адрес первой записи в блоке, содержащей наименьшее значение ключа.

В этом случае поиск экземпляра записи осуществляется посредством первоначального установления номера блока, в котором может находиться запись, а затем последовательного поиска в этом блоке, пока не будет установлено местонахождение искомой записи или ее отсутствие.

Данный метод позволяет обеспечивать быстрый доступ к записям баз фреймов и файлов большой размерности, в которых поиск по одному только индексу приводит к значительным затратам времени.

Индексно-произвольный метод доступа. Данный метод обеспечивает доступ к экземплярам записей на основе использования индекса. Поэтому метод называют также «метод доступа с полным индексом».

Полный индекс представляет собой такую организацию файла, при которой для каждого конкретного экземпляра записи предусмотрен соответствующий индекс. Этот индекс составляется из значения

первичного ключа и указателя экземпляра записи, содержащий это значение (рис. 12.10).

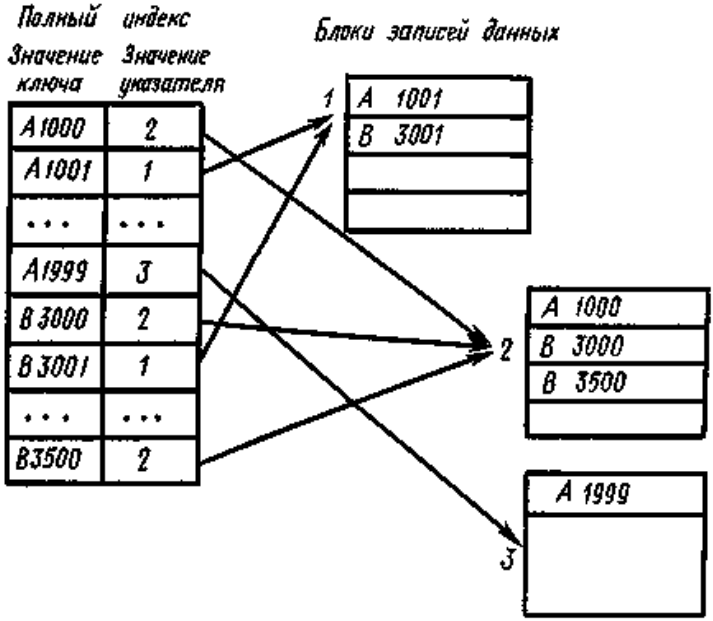


Рис. 12.10. Схема доступа с полным индексом

Обычно для ускорения поиска индексы упорядочиваются. При этом упорядочивание или физически близкое размещение хранимых записей не требуется.

После того как в индексе обнаружено искомое значение ключа, доступ к записи можно осуществить с помощью указателя, который хранится в индексе рядом со значением ключа. В этом случае, если искомое значение ключа в индексе не найдено, поиск завершается на уровне индекса. Допускается произвольный доступ к индексу посредством хэширования.

Метод доступа с полным индексом обеспечивает эффективную выборку одиночных записей, а также простоту операций обновления.

Методы доступа, основанные на использовании древовидных структур. Получили широкое распространение методы представления в памяти ЭВМ данных в виде древовидных сетевых структур и соответствующие методы доступа к ним. Эти методы стали конкурентами классических, таких, как индексно-последовательный метод или хэширование.

К основным древовидным структурам обычно относят: бинарное дерево и *B*-дерево. Бинарным (двоичным) деревом называется древовидный граф с двоичным ветвлением, т.е. с таким ветвлением, когда из каждой вершины (кроме концевых) выходят две дуги.

Бинарные деревья представляют собой вид структур баз фреймов, обеспечивающий как произвольную, так и последовательный выбор фреймов.

Важную роль проектирования древовидных структур фреймов играет понятие сбалансированного дерева. Прежде чем его рассмотреть, введем некоторые дополнительные понятия. Уровень вершины *i* определяется длиной пути от корневой вершины *T* до вершины *i*. Корневая вершина *T* имеет нулевой уровень. Ветвь дерева определяется его максимальным уровнем.

Дерево называется сбалансированным, если разница уровней любых двух конечных вершин не превышает единицы. Построение сбалансированного дерева делает равновероятным обращение к любой из его вершин, что позволяет минимизировать среднюю длину доступа.

Механизм поиска по бинарному дереву основан на том, что каждая вершина дерева помечена отдельным ключом.

Ключи упорядочены следующим образом: $k_{i_1} \leq k_i \leq k_{i_2}$, где k_i—ключ *i* -й вершины; k_{i_1} — ключ i_1-вершины, соответствующей левой дуге *i*-й вершины (—); k_{i_2} — ключ вершины, лежащей на правой дуге.

При поиске некоторого ключа *k* вначале просматривается корневая вершина дерева *T* и сравнивается ключ *k* с ключом k_T корневой вершины. В случае $k=k_T$ поиск завершен успешно. В том случае, когда $k<k_T$, поиск продолжается в левом поддереве, а при $k>k_T$ поиск продолжается в правом поддереве (рис. 12.11).

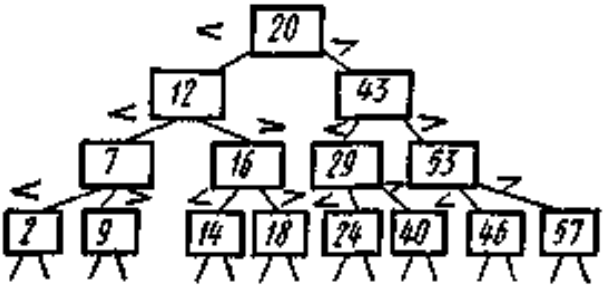


Рис. 12.11. Бинарное дерево поиска

Кроме бинарных деревьев в БФ используются так называемые *B*-деревья.
B-дерево представляет собой обобщение понятия бинарного дерева. В нем из каждой вершины могут выходить более двух ветвей. Обычно *B*-деревья используются только для организации индекса. Записи данных располагаются в отдельной области, для которой возможен произвольный доступ. Каждая вершина *B*-дерева состоит из совокупности значений первичного ключа, указателей индексов и ассоциированных фреймов. Указатели индекса используются для перехода на, следующий, более низкий уровень вершины в *B*-дереве. Ассоциирование фреймов фактически представляют собой совокупность указателей фреймов и служат для определения физического местоположения фреймов, ключевые значения которых хранятся в этой вершине индекса.
Имеется множество различных гибридных древовидных организаций, сочетающих в себе лучшие качества метода хэширования, бинарных деревьев, *B*-деревьев и их многочисленных вариантов.

12.2. Методы визуального отображения сцен

12.2.1. Структура и задачи визуального отображения сцен

Для формирования визуального отображения сцен необходимо иметь информацию об окружающей среде (о расположении предметов, об их форме, размерах и т. д.). Несмотря на богатство естественного языка, сообщать в нем эту информацию сложно, а скорее всего и невозможно. Стремление представить в удобной форме сведения об окружении приводят к введению в структуру ИСФ средств восприятия визуальной информации - визуально отображенных сцен .
В данном разделе мы опишем подсистему визуального отображения сцен (ПВОС). Назначение подсистемы заключается в том, чтобы на основании двумерного изображения и информации о дальности получить в терминах знаний трехмерное описание предъявленной структуры сцен. Процесс отображения (восприятия) изображения сцен (в дальнейшем: изображения) может быть условно разделен на следующие этапы.
1. Предварительная обработка изображения. В этот этап включают кодирование и аппроксимацию изображения, применение к изображению пространственно-инвариантных преобразований

(фильтрации, коррекции, сглаживания или обострения изображения) с целью повышения его качества.

2. Выделение на изображении областей и контуров и получение их формального описания. Задача данного этапа состоит в выделении на изображении, представленном в виде двумерной матрицы, яркостных точек, контуров и областей. Затем производится разбиение контура на сегменты, определение типа линии, соответствующей сегменту, **нахождение топологических и метрических свойств изображения.**

3. Выделение объектов, представленных на изображении. Этап заключается в семантической интерпретации линий и областей, выделенных на предыдущем этапе, с целью объединения их в связные объекты.

4. Определение трехмерной структуры сцены. Задача этапа состоит в определении пространственных отношений, связывающих объекты, представленные на сцене. Выполнение задачи осуществляется путем установления соответствия между изображением и сценой благодаря калибровке камеры, информации о дальности и т. п.

5. Опознавание выделенных в изображении объектов и составление описания сцены в соответствующих терминах.

В данном разделе не рассматриваются задачи первого и второго этапов, так как они лежат в стороне от основного содержания книги и являются темой самостоятельных исследований.

Рассматриваемые в этом разделе этапы касаются выделения из изображения семантической информации и связи ее с ИСФ. При изложении материала будем исходить из предположения, что определенная информация уже извлечена из изображения и представлена в символическом виде (а не в виде яркостных точек изображения). Например, будем предполагать, что **из изображения извлечена и описана (представлена в символической форме) информация о линиях и областях изображения.** Кроме того, как правило, будем предполагать существование некоторой априорной информации о классе анализируемых сцен, например, что все сцены состоят только из многогранных объектов.

Задача рассматриваемых в разделе методов — отобразить символическую информацию о вводимой сцене на накопленные знания системы.

Приведенное выше деление на этапы весьма условно, а последовательность их выполнения может отличаться от указанной выше. При организации подсистем визуального отображения сцен в ИСФ не используют фиксированную последовательность вызова подпрограмм, реализующих определенные функции, вместо этого

используют так называемый *гетерархичный принцип организации*. Это понятие не имеет еще четкою определения (термин «гетерархичный» можно трактовать как «с переменной иерархичностью»), но оно включает в себя организацию программ, обладающую приводимыми ниже свойствами.

1. *Система* должна быть *целенаправленной.* Процедуры на всех уровнях должны связываться с некоторыми определенными целями и должны быть довольно краткими. Цели, как правило, должны либо непосредственно вызывать несколько примитивных процедур, либо сводиться к небольшому числу подцелей. Как следствие система должна работать методом «сверху—вниз».
2. *Управление* является не централизованным, а *распределенным по всей системе.* Программные модули взаимо действуют как равноправные единицы.
3. От разработчика системы требуются минимально возможные знания о состоянии системы в момент вызова процедуры. Процедура сама должна знать и создавать условия, которые требуются для ее работы. Это свойство позволяет наращивать систему различным пользователям без детального знания всей системы (что является принципиальным требованием при разработке сложных программ).
4. Система должна обладать некоторыми знаниями о себе. Должны существовать программные модули, выражающие критику при обнаружении подозрительных ситуаций, и модули, предсказывающие неудачу примитивных подпрограмм. Средства связи между этими модулями, кроме обычных средств (передача данных), должны включать сообщения, аналогичные совету, поддержке, замечанию, недовольству, критике, вопросу, ответу и т п
5. Система должна обладать способностью делать пробные заключения и обнаруживать свои ошибки. Если обнаруживается, что предположение ошибочно, то система должна определять, какие факты в базовых данных являются наиболее проблематичными и какие изменения наиболее вероятны.

Графически такая система подобна сети, а не неизменной последовательности процедур. **Каждая процедура связывается с другой через множество возможных связок**, передающих управление. Какие из этих связок используются в конкретном случае, зависит как от контекста, определяемого решаемой задачей, так и от процедур, имеющихся в системе

Заметим, что при гетерархичной организации теряет смысл понятие подпрограмм верхнего и нижнего уровня, указывающего порядок вызова процедур. В данном случае целесообразно говорить об уровне процедур в смысле рода их работы Например, программа поиска

линии, работающая с изображением, представленным в виде яркостных точек, может в определенных случаях вызывать программу анализа модели предполагаемого класса объектов, являющуюся программой более высокого уровня.

В системах, использующих гетерархичную организацию, управление может осуществляться под влиянием окружения. Как отметил Саймон, это позволяет при довольно простой ведущей программе, но сложном окружении, демонстрировать интеллектуальное поведение.

Нас будут интересовать довольно сложные изображения, при обработке которых не удается ограничиться отнесением предъявленного изображения к одному из известных классов, а требуется получить описание изображения, причем число возможных описаний столь велико, что бессмысленно считать каждое из них определением отдельного класса. Под *описанием изображения* мы будем понимать перечень объектов, их свойств и взаимосвязей. Другими словами, мы хотим подчеркнуть, что изображения, с которыми нам приходится иметь дело, являются сложными и не могут быть полностью обработаны методами распознавания образов.

12.2.2. Формальное описание структуры понятия «сцена»

Рассмотрим примеры формализмов, которые могут быть использованы как основа для описания произвольных сцен.

12.2.2.1. Синтаксические описания.

Рассмотрим простейшую сцену, изображенною на рис. 12.12.

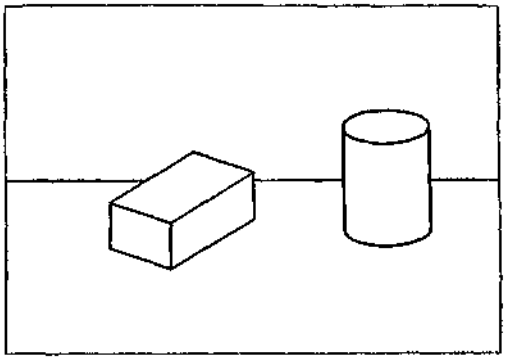


Рис 12.12. Пример простой сцены.

При кратком описании понятия «сцена» может быть охарактеризована как «параллелепипед и цилиндр». Более детально данная сцена может быть описана как «параллелепипед, расположенный слева от цилиндра». Можно продолжить детализацию описания, охарактеризовав, например, параллелепипед как совокупность трех граней и т. д. до любого желаемого уровня конкретности. *Указанный способ последовательного уточнения описания изображения будем называть* ***лингвистическим****,* так как он подобен процессу анализа предложения естественного языка. Для анализа изображения в этом подходе, так же как в случае анализа предложений, аналогичным образом вводится понятие грамматики.

Однако, в отличие от грамматик, используемых в естественных языках, ***грамматики для изображений являются не одномерными, а двумерными.*** В одномерных строках естественной операцией соединения символов является *операция конкатенации — размещения символов друг за другом, в двумерных строках такой естественной операции не существует*. Поясним эту мысль на примере рис. 12.12. Одно из возможных деревьев разбора для данной сцены приведено на рис. 12.13.

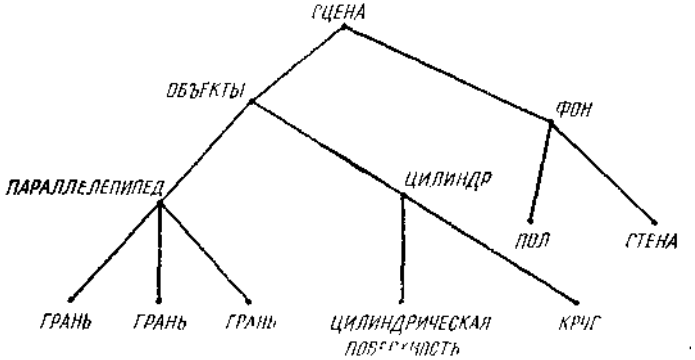


Рис. 12.13. Дерево разбора для сцены, приведенной на рис. 12.12.

Даже если мы точно определим терминальные (понятийные) вершины, дерево разбора будет только приблизительно описывать сцену. Например, три грани могут быть соединены множеством способов, из которых только некоторые дадут параллелепипед.

Существуют различные подходы к определению способа соединения символов в двухмерной строке (плоскости).

Наиболее прямолинейным является способ, полагающийся исключительно на описание границы некоторой фигуры, что дает возможность воспользоваться преимуществами естественного

упорядочения точек в одномерном множестве. В качестве примера этого подхода приведем описание четырехугольника (грани).

ЧЕТЫРЕХУГОЛЬНИК — ОТРЕЗОК + ОТРЕЗОК + ОТРЕЗОК + ОТРЕЗОК, где «+» обозначает конкатенацию, и предполагается, что *результирующая строка должна замыкаться на себя*. Выбор терминального (понятийного) символа (отрезка) в этом простом примере является очевидным. Однако для фигур, состоящих из гладких кривых, этот выбор является менее очевидным, и, кроме того, часто трудно определить, где заканчивается один терминальный (понятийный) символ и начинается другой. Проблема идентификации терминальных (понятийных) символов в изображении свойственна не только подходу, основанному на описании границ, но и любому синтаксическому методу.

Предложенный метод не решает всех проблем. Действительно для описания параллелепипеда, приведенного на рис. 12.12, в понятиях определенных выше четырехугольников необходимо конкретизировать операцию соединения четырехугольников. Одним из часто используемых способов является определение точек, в которых происходит соединение четырехугольников (рис. 12.14).

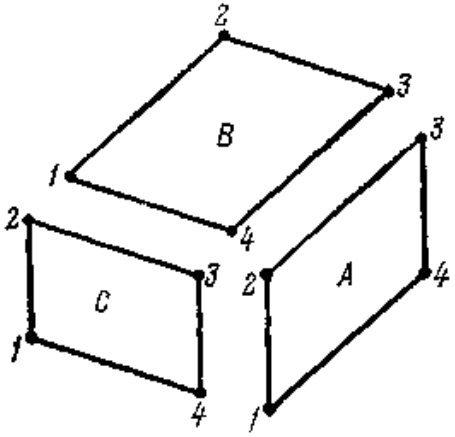


Рис. 12.14. Разложение параллелепипеда на грани

В этом случае синтаксическое описание параллелепипеда может иметь вид: «соединить точку 1 четырехугольника *A* с точкой 4 четырехугольника С» и т. д. для всех вершин.

Аналогичный подход к определению отношений между символами основывается на стандартизации точек соединения. Условимся, например, что каждый понятийный символ имеет две такие точки,

называемые «головой» и «хвостом». Будем считать, что операция конкатенации состоит в присоединении «головы» первого символа к «хвосту» второго путем их перемещения (без вращения) в плоскости изображения. «Хвостом» получившегося непонятийного символа будем называть «хвост» первого понятийного символа, а «головой» — «голову» второго понятийного символа. Так, например, если *b, c, d*— понятийные символы, то непонятийный символ $A=b+c+d$ будет иметь в качестве «хвоста» «хвост» *b,* а в качестве «головы» — «голову» *d.* В качестве примера такого способа задания грамматики опишем цилиндр, изображенный на рис. 12.12. При этом будем использовать понятийные символы, представленные на рис. 12.15.

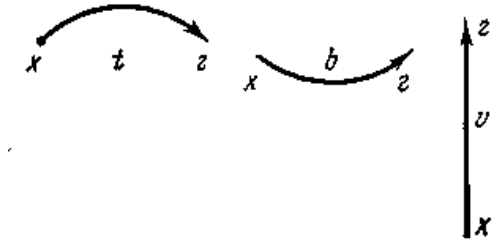


Рис 12.15. Множество понятийных символов для сцены, изображенной на рис.12.12

Операцию конкатенации будем обозначать символом «+», а символом «~» — операцию переобозначения у понятийного символа «головы» и «хвоста». Определим дополнительно операцию *, состоящую в присоединении «головы» символа *p* к «голове» символа *q* и «хвоста» *p* к «хвосту» *q.* «Головой» образованного символа будем считать точку соединения «голов» *p* и *q,* а «хвостом» — точку соединения «хвостов» *p* и *q.* В принятых обозначениях цилиндр можно записать следующим образом:

1. ЦИЛИНДР:: = СТОРОНА * КРУГ.
2. СТОРОНА:: = $\tilde{v}+b+v$.
3. КРУГ:: = $t*b$.

Рассмотрим на этом примере, как осуществляется синтаксический анализ изображения.

Нас в основном будет интересовать процесс распознавания изображения, а не порождения (см.11.2). Любая распознающая грамматика предполагает умение распознавать терминальные символы. Указанная операция в общем случае является довольно сложной, и поэтому при обработке двумерных строк (в отличие от одномерных) разбор «сверху—вниз» более предпочтителен, чем разбор «снизу—

вверх» (см.11.4), так как он определяет вид и возможное местоположение распознаваемого терминального символа.

В приводимом нами примере необходимо уметь распознавать вертикальные линии и два типа кривых линий.

Предположим, что нам предъявлено изображение и требуется определить, содержится ли в нем цилиндр. Применим процедуру разбора «сверху — вниз». Правила 1 и 2 говорят о том, что цилиндр должен включать в себя символ СТОРОНА и что этот символ должен содержать терминальный символ вертикального отрезка. Поэтому процедура будет осуществлять просмотр изображения с целью поиска вертикального отрезка. (Отметим, что при разборе символов одномерной строки требуется просто выбрать первый элемент строки.) Найдя вертикальный отрезок, мы будем рассматривать его нижний конец как «голову», а верхний как «хвост», так как в соответствии с правилом 2 мы ищем $\tilde{v}$. Из правила 2 видно, что терминальный символ b должен быть присоединен к «голове» вертикального отрезка, поэтому необходимо исследовать область изображения в районе нижнего конца отрезка $\tilde{v}$ с целью поиска кривой вида b. Если кривая b не найдена, то необходимо искать другой вертикальный отрезок. Если кривая b найдена, то в соответствии с правилом 2 необходимо искать в районе конца кривой b вертикальный отрезок. Если вертикальный отрезок найден, то в изображении опознан нетерминальный символ СТОРОНА. Затем в соответствии с правилом 1 осуществляется поиск символа КРУГ. Если символ КРУГ найден и его расположение на изображении соответствует операции *, то операция распознавания заканчивается успешно.

Как уже было указано ранее, использование метода разбора «сверху—вниз» позволяет направить процедуру распознавания терминальный символов. В нашем примере, за исключением распознавания первого терминальный символа и, указанная процедура применялась не ко всему изображению, а к его некоторой области. Этот целенаправленный аспект алгоритма разбора «сверху—вниз» не только значительно уменьшает количество вычислений, но и уменьшает вероятность обнаружения ошибочного терминального символа, т. е. относящегося не к исследуемой фигуре, а к некоторому другому объекту анализируемого изображения.

Заканчивая рассмотрение синтаксических методов, отметим два основных аспекта, возникающих при использовании некоторой грамматики. Первый состоит в уже упомянутой ранее проблеме распознавания в изображении терминальных символов. Природа процесса разбора такова, что ошибка в распознавании одного терминального символа может привести к существенно отличному

результату. Второй аспект касается целесообразности применения синтаксического анализа к тем или иным классам изображений. Одной из основных существенных сторон использования грамматики является рекурсивность, т. е. возможность в компактной форме представлять некоторые характеристики (признаки) входной строки. Таким образом, преимущества от использования грамматики могут проявиться на изображениях, состоящих из небольшого множества терминальных символов, образующих нетерминальные компоненты небольшим количеством рекурсивных правил. В изображениях, не удовлетворяющих указанному требованию, использование грамматики становится формальным способом исчерпывающего описания «в словах» полного содержания изображения, не дающего преимуществ в компактности.

12.2.2.2. Семантические сети

При анализе многих изображений структуру изображения целесообразно описывать «в словах», т. е. символически. Кроме синтаксических методов, приведенных выше, для этих целей удобно использовать *семантические сети, т. е. представлять структуру сцены в виде графа*. При этом узлы графа помечаются наименованиями частей сцены, а дуги — наименованиями семантических отношений, в которых находятся связываемые ими узлы. В качестве примера опишем в виде семантической сети сцену, изображенную на рис. 12.12. Структура графа будет подобна дереву разбора, представленному на рис. 12.13. Для описания сцены введем, например, следующие отношения: «часть», «тип», «смежный», «слева», «справа», «выше», «ниже». Вид одной из возможных семантических сетей, описывающих указанную сцену, приведен на рис. 12.16.

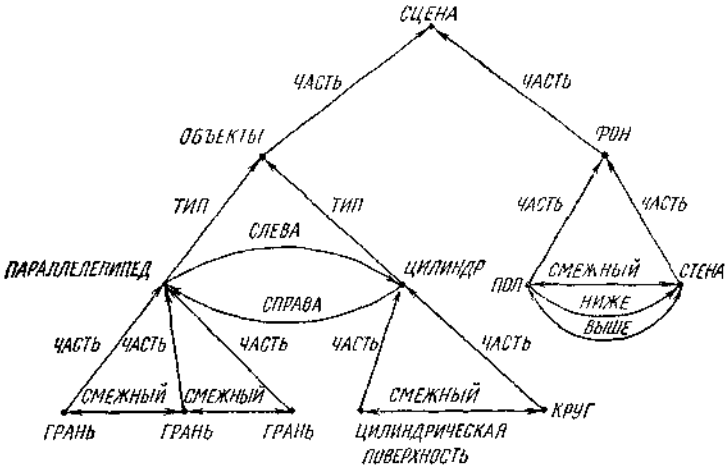


Рис. 12.16. Семантическая сеть для сцены, приведенной на рис. 12.12.

Дуги на рисунке направлены таким образом, что при чтении наименования узла у «хвоста» дуги, наименования дуги и наименования узла у «головы» дуги получается фраза естественного языка. Например, ЦИЛИНДР ТИП ОБЪЕКТОВ.

Как будет показано ниже, в виде семантических сетей удобно представлять не только описание анализируемой сцены, но и знание системы о понятиях (модели понятий), которые должны быть обнаружены на сцене.

12.2.3. Трехмерными модели объектов (истин)

Описание сцены, состоящей из трехмерных объектов (истин), можно осуществить одним из двух способов. Первый способ состоит в игнорировании трехмерной природы реальных объектов и описании сцены в понятиях двумерных конструкций. Второй способ описывает собственно трехмерные объекты, а не их изображения. Будем называть эти способы соответственно двумерным и трехмерным описанием сцены. Проиллюстрируем различие между способами на примере сцены, представленной на рис. 12.12. Двумерное описание может быть выражено, например, так: «три смежных четырехугольника, эллипс с примыкающей к нему криволинейной фигурой и три коллинеарных прямых отрезка». Трехмерное описание может быть определено,

например, такими словами: «параллелепипед и цилиндр, стоящие на полу перед стеной». И тот, и другой способы используются при распознавании изображений. Мы отдадим предпочтение трехмерному способу описания сцены, так как он является более общим, дает реальное представление об окружении.

В связи с тем, что отображение двумерного изображения на трехмерную сцену не является однозначным (существует бесконечное число трехмерных объектов, соответствующих одному изображению), трехмерное описание может быть извлечено из двумерного изображения только на основания какой-либо дополнительной информации об объектах, присутствующих на сцене. Эта информация задается в виде трехмерных моделей объектов окружения. **Трехмерное описание сцены вырабатывается как результат работы некоторой процедуры, интерпретирующей предъявленное изображение в понятиях трехмерной модели**. Если окружение является достаточно простым, то модели могут выражать только небольшое количество деталей сцены при довольно простой процедуре, сопоставляющей изображение модели. Так, например, для сцены, изображенной на рис. 12.12 модель могла бы быть описана так: «все ребра параллелепипеда являются прямолинейными отрезками, а у цилиндра некоторые ребра образованы криволинейными отрезками». Даже такой грубой модели было бы достаточно для интерпретации рисунка 12.12 в понятиях трехмерных объектов.

Для более сложного окружения будет требоваться более точная и полная модель. Часто, однако, возможно представлять сложные признаки, характеризующие изображения в виде простых признаков о моделях. В общем случае будет требоваться достаточно полное сравнение между трехмерными моделями и изображениями.

В частном случае, когда модель определяется конечным числом точек в трехмерном пространстве, а количество моделей конечно, для сопоставления изображения с моделью может быть использована следующая процедура. Предъявленное изображение сопоставляется с каждой моделью. При этом модель рассматривается со всех углов зрения. Для каждого угла зрения вычисляется проекция точек модели и производится сопоставление их с точками изображения. Модель, проекции которой в некотором положении наилучшим образом сопоставляются с изображением, считается моделью анализируемого изображения.

Проблема вычисления лучшего сопоставления между моделью и изображением может быть разделена на две подпроблемы:

1) идентификация точек модели и изображения, по которым производится сопоставление;

2) вычисления степени соответствия модели и изображения.

Первая подпроблема в общей постановке является очень сложной, и каких-либо глубоких результатов пока не получено. Однако в конкретных случаях, например, для сцен, состоящих только из многогранников, существуют элементарные геометрические свойства, достаточные для установления соответствия между изображением и моделью (или во всяком случае для существенного уменьшения неоднозначности соответствия). Для указанных ограничений (многогранники) вторая подпроблема может быть решена аналитически при полном описании трехмерной структуры каждой модели. Мы не будем останавливаться на описании этого метода, так как он требует полного описания моделей объектов (точнее, трехмерных координат всех вершин, выбранных для сопоставления) и применим только к простым сценам, что редко бывает в реальных ситуациях.

Ниже мы опишем методы, позволяющие анализировать сцены без полного знания моделей объектов, однако основанные на некоторых априорных знаниях о классе объектов, которые могут появляться на сцене. Мы ограничимся рассмотрением объектов, образованных многогранниками. Выбор довольно простого, но распространенного класса объектов вызван двумя причинами.

1. Указанное ограничение на класс рассматриваемых объектов практически не влияет на решение основной задачи — составление описания предъявленной сцены в понятиях знаний субъекта.

2. Только для указанных ограничений разработаны процедуры анализа изображений, практически не зависящие от вида конкретных объектов.

Для методичности изложения будем описывать этапы обработки изображения последовательно, но не следует забывать, что обработка осуществляется по гетерархическому принципу. Даже при восприятии простых реальных объектов возникают различного рода помехи (блики, тени и т. п.), не позволяющие выполнять этапы последовательно. В общих чертах процесс обработки изображения происходит следующим образом. Сначала выявляется из изображения наиболее достоверная и легко извлекаемая информацию, такая как размер и форма внешнего контура. Затем из множества всех моделей, имеющихся в системе, отбираются модели — кандидаты, не противоречащие первичной информации. Среди них выбирается наиболее подходящая модель, которая используется для того, чтобы направлять дальнейший процесс обработки изображения. Анализируя выбранную модель, предсказываются свойства, которые должны быть исследованы на изображении в первую очередь, и указывается место

их расположения. При обнаружении предсказанных свойств на изображении процесс предсказания продолжается либо до полного опознавания объекта, либо до выявления различия между предсказанием и изображением. При обнаружении различий следует определить, не вызваны ли они поворотом объекта в пространстве или загороженностью одного тела другим и т. п. Если различие относится к такому классу, то оно устраняется. Так, например, можно предположить, что эллипс на изображении соответствует окружности в модели. Приняв данное предположение, следует определить угол наклона наблюдаемой поверхности и произвести пересчет остальных видимых точек. Если различие является неустранимым, то это значит, что выбранная модель не соответствует анализируемой сцене. В этом случае следует выбрать из списка моделей-кандидатов очередную модель, наиболее соответствующую собранной к данному моменту информации.

Минский разработал теорию, на основании которой осуществляется не выбор новой модели, а получение ее из предыдущей путем трансформаций.

12.2.4. Разбиение сцены на отдельные объекты

Задача разбиения сцены на отдельные объекты является необходимой для сцен реальной сложности. Общий механизм, осуществляющий указанную операцию, пока не предложен. Однако при принятых нами ограничениях эта задача имеет эффективные решения.

12.2.4.1. Семантика линий.

Нас будет интересовать семантика, выражаемая линиями (отрезками), из которых состоит изображение. Будем предполагать, что сцена содержит только многогранники степени три, т. е. точно три плоскости пересекаются в каждой вершине многогранника. Заметим, что при этом предположении линия на изображении может иметь только одно из трех значений:

1) обозначать ребро, являющееся пересечением граней, образующих впадину (ребро—«впадина»);

2) обозначать ребро, являющееся пересечением граней, образующих выступ («выпуклое» ребро), у которого на изображении видны его грани;

3) изображать выпуклое ребро, у которого на изображении скрыта одна из граней.

Будем называть линии первого типа *впадинами* и помечать знаком «—», линии второго типа — *выпуклыми ребрами* («+»), а линии третьего типа — *скрытыми ребрами* (→). Направление стрелки → выбирается таким образом, что если смотреть в направлении стрелки, то видимая грань ребра находится справа. Все перечисленные типы ребер представлены на сцене, изображенной на рис. 12.17.

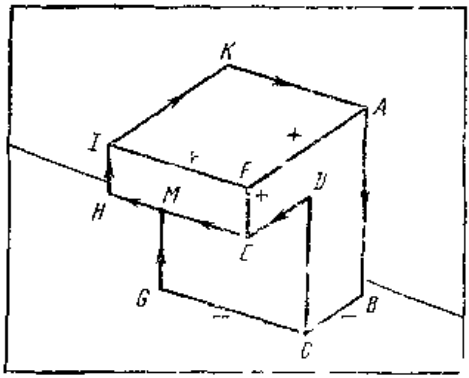


Рис. 12.17. Изображение объекта с помеченными ребрами

Например, линии *EF* и *AF* являются выпуклыми ребрами, линии *BC* и *CG* — впадинами, линии *EH, ЛВ*— скрытыми ребрами. При сделанных ранее предположениях значение линии остается неизменным по всей ее длине. Вторым важным наблюдением является тот факт, что все вершины делятся на четыре типа. Действительно, так как вершины образованы пересечением трех плоскостей, которые делят пространство на восемь частей (октантов), то вершины можно разделить на типы в зависимости от того, сколько из восьми октантов, смежных с данной вершиной, принадлежит объекту. Существуют вершины типа *1, 3, 5* и *7*. На рис. 12.18 приведены возможные типы вершин.

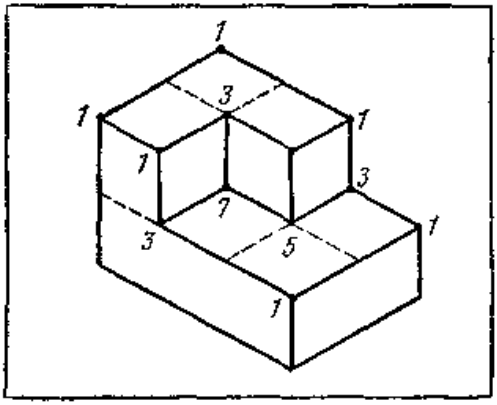


Рис. 12.18. Изображение объекта с помеченными вершинами.

Необходимо иметь в виду, что вершины одного и того же типа могут иметь различное представление на изображении (см. рис. 12.18). Это подчеркивает тот факт, что тип вершины является свойством самого объекта, а не его изображения. Изобразим все способы, в которых могут быть представлены вершины четырех типов. Следует иметь в виду, что вершина каждого типа может рассматриваться только из свободных октантов, т. е. октантов, не занятых объектом, которому принадлежит вершина. Заметим, что рассматривание объекта из различных точек одного октанта не приводит к существенному изменению изображения. Учитывая вышеизложенное, надо рассмотреть вершины типа *1, 3, 5, 7* соответственно с семи, пяти, трех и одной точек зрения. На рис. 12.19 приведен каталог возможных способов изображения четырех типов вершин.

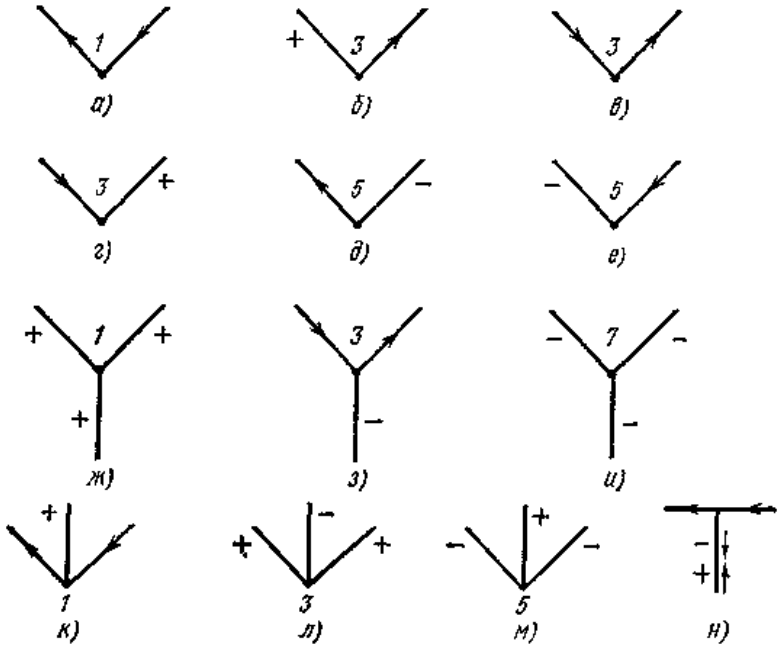


Рис. 12.19. Каталог возможных способов изображения четырех типов вершин.

Вершина типа *1* с семи возможных точек зрения дает только три различные изображения. Многие из способов представления вершин, приведенных в каталоге, могут быть найдены на рис. 12.17. Например, вершина *F* изображена на рис. 12.19, *ж,* вершина *К*— на рис. 12.19, *а,* вершина *А* —на рис. 12.19, *к* и т. д. На рис. 12.19, *н* показана так называемая *Т*-конфигурация, не соответствующая реальной вершине; она возникает, когда некоторая грань закрывает более удаленную часть объекта (см. точку *М* на рис. 12.17). Четыре возможные способа разметки ребер *Т*-конфигурации указаны на рис. 12.19, *н*. Будем ребро *Т*-конфигурации, которое может быть помечено разными символами, называть *основой (основанием),* а второе ребро — *перекладиной.*

Будем вершины, изображенные на рис. 12.19 *а—е* называть *V-вершинами,* вершины на рис. 12.19, *ж—и — вершинами типа Y,* а вершины на рис. 12.19, *к—м — вершинами типа W.*

Анализ каталога (рис. 12.19) показывает, что тип вершины и линий не может быть определен исследованием только локальной информации,

необходимо привлечение некоторой контекстной информации. Как это может быть сделано, мы покажем на примере рис. 12.20.

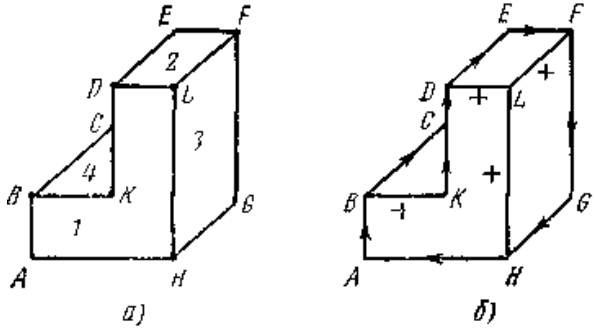


Рис.12. 20. Пример разметки вершин.

Заметим, что ребра внешнего контура некоторого изолированного тела могут быть размечены меткой → при обходе контура по часовой стрелке (см. рис. 12.20, б), так как при этом видимая грань тела всегда остается справа. Теперь займемся анализом вершин *B, D, F, H.* Каждая из этих вершин имеет тип *W* со стрелками на внешних ребрах. Анализ каталога показывает, что существует только одна вершина такого типа (см. рис. 12.19, *к).*

У этой вершины среднее ребро должно быть помечено знаком «+». Следовательно, ребра *BK, DL, LF, LH* должны быть помечены знаком «+». Точке *C* на рис. 12.20 соответствует вершина типа *T.* В соответствии с каталогом поперечина вершины типа *T* должна помечаться стрелкой, т. е. линия *KC* помечается стрелкой. В данном примере мы смогли формально и однозначно назначить метки каждой линии. Однако это не всегда возможно. Так, например, для изображения на рис. 12.17 вершины *B, C, G* не определить однозначно с помощью формального приема. Это отражает объективную причину: по изображению нельзя сказать, стоит ли объект на полу или висит над ним.

Формализуем описанную выше процедуру разметки следующим образом. Выберем некоторую вершину объекта и присвоим ей какую-либо разметку ребер из каталога. Разметка данной вершины наложит ограничения на вершины, смежные с ней. Будем повторять этот процесс до тех пор, пока не получим вершину, не удовлетворяющую каталогу, или не закончим разметку всего объекта. Представим данный процесс в виде дерева поиска. Узлам дерева поиска поставим в соответствие вершины изображения, а дугам, исходящим из данного узла, варианты разметки данной вершины, выбираемые из каталога.

Проиллюстрируем работу процедуры на примере объекта с отверстием, представленного на рис. 12.21.

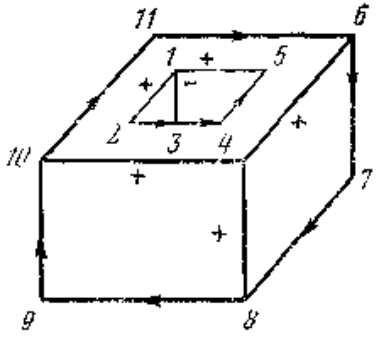


Рис. 12.21. Объект с квадратным отверстием

Выберем произвольный порядок обработки вершин, например, совпадающий с присвоенными им номерами. Вершине *1,* являющейся частью отверстия, на основании каталога может быть присвоено три способа разметки (рис. 12.19, *к, л, м).* Следовательно, из вершины *1* дерева поиска (см. рис. 12.22) будет исходить три дуги.

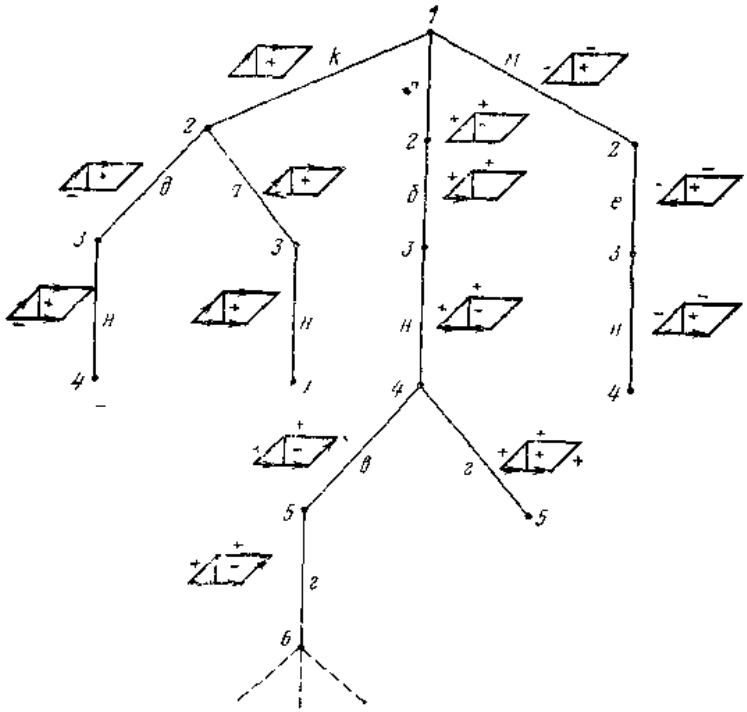


Рис. 12.22. Дерево поиска для объекта с квадратным отверстием.

Будем помечать дуги дерева поиска буквами, соответствующими выбранному из каталога варианту разметки. Для удобства рядом с дугой будем изображать отверстие с принятой на данном шаге разметкой. После рассмотрения вершины *1* выбираем вершину *2*. Способы ее разметки будут находиться в зависимости от разметки, присвоенной вершине *1*. В варианте *1к* (см. рис. 12.22) для разметки вершины *2* имеются две возможности (см. рис. 12.19, *д* и рис. 12.19, *а)*. Аналогичным образом раскрываются вершины дерева в вариантах *1л* и *1м*. Затем выбирается вершина *3*, имеющая тип *Т*. В варианте разметки *1к, 2д* для вершины *3* возможен только один способ разметки. Действительно, направление перекладины у вершины типа *Т* определяется каталогом однозначно (если смотреть в направлении стрелки, то ребро — основа должно находиться слева). Но тогда вершине *4* в каталоге не соответствует никакого способа разметки. Это означает, что данный вариант разметки является неприемлемым, и соответствующая ему ветвь дерева обрывается Продолжая этот

процесс, можно получить единственную разметку линий всей фигуры, изображенной на рис. 12.21. Напомним, что ранее мы установили возможность размечать линии внешнего контура фигуры по часовой стрелке. Это правило сократит перебор вариантов при разметкесоставгаихся вершин. Описанный алгоритм сводится к поиску пути в дереве**.**

12.2.4.2. Объединение областей в объекты.

Одна из основных проблем при анализе изображений многогранников состоит в выделении объектов. Существуют различные эвристические алгоритмы, направленные на решение этой задачи, однако пока нет ее полного теоретического анализа. Перед тем как перейти к изложению конкретных методов, сделаем некоторые предварительные замечания.
Предположим, что мы хотим на изображении, состоящем из линий, образующих многогранники (возможно, закрывающие друг друга), выделить отдельные объекты. Другими словами, нас интересует, какие из областей, выделенных на изображении, образуют некоторый многогранник. Если больше не наложено никаких ограничений, то задача не имеет однозначного решения. Действительно, даже для такого простого изображения, какое представлено на рис. 12.17, существует два варианта: либо многогранник стоит на плите (или висит над ней), либо он образует одно целое с плитой. В связи с тем, что однозначного решения не существует, мы можем надеяться в лучшем случае на метод, дающий на большинстве изображений решения, аналогичные решениям человека.
Для обоснования эвристик рассмотрим простой многогранник, изображенный на рис. 12.23.

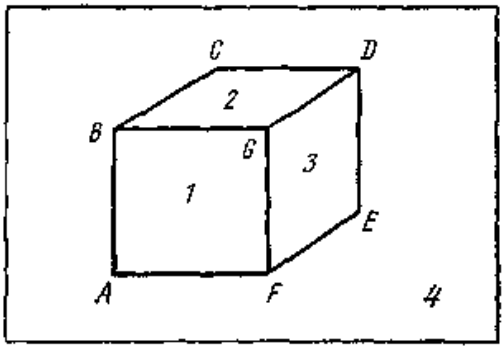


Рис. 12.23. Пример простой сцены

Предположим, что любой разумный метод выделения частей из данного рисунка должен определить, что области *1, 2, 3* объединяются в одну группу, а область *4* (фон) в другую. Если мы рассмотрим 7 видимых вершин многогранника, то здесь присутствуют: три *V*-вершины, три *W*-вершины и одна *Y*-вершина. Это вместе с фактом, что области *1, 2* и *3* должны объединяться вместе, предполагает введения следующих эвристических правил:

1) *Y*-вершина дает основания предполагать, что представленные в ней области должны быть объединены;

2) *W*-вершина дает основание предполагать, что две области, ограниченные ребрами, образующими между собой острые углы, должны быть объединены.

На рис. 12.24 изображены два расположенных друг на друге многогранника и находящаяся за ними треугольная призма.

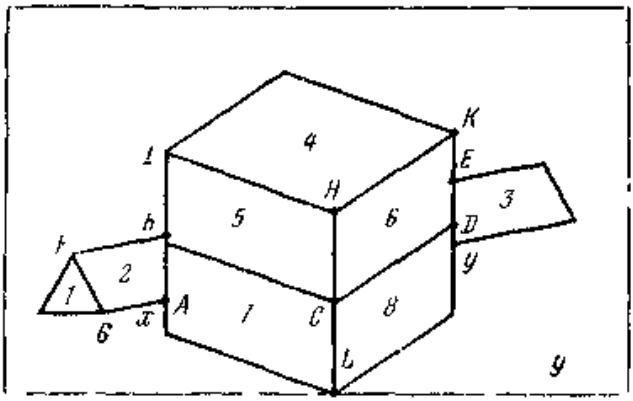


Рис. 12.24. Сцена, изображающая частично скрытый объект.

Вершина *C* называется вершиной типа ψ. Она наводит на мысль ввести следующее эвристическое правило:

3) *ψ* -вершина дает основания предполагать, что две верхние области должны образовать одну группу, а две нижних области — другую. (На рис. 12.24 ψ-вершина C дает основание для объединения областей *5* и *6* и областей 7 и *8.)*

Рассмотрим теперь на рис. 12.24 *T*-вершины *A* и *D*. Ранее мы отмечали, что перекладина *T*-вершины закрывает более удаленную часть сцены (это справедливо не только для многогранников степени три). Кроме того, факт, что основы *T*-вершин *A* и *D* параллельны, дает основание предположить, что области *2* и *3* принадлежат одной группе, а области

x и *y* в окрестности точек *C* и *D* принадлежат другой. Можно сформулировать следующее эвристическое правило:

4) параллельность оснований двух *T*-вершин дает возможность предполагать, что области, находящиеся по одну сторону от оснований, образуют одну группу, а области, расположенные по другую сторону, образуют другую группу.

На рис. 12.25 дана иллюстрация четырех перечисленных выше эвристических правил. Пунктирами обозначены связки, указывающие на объединяемые области.

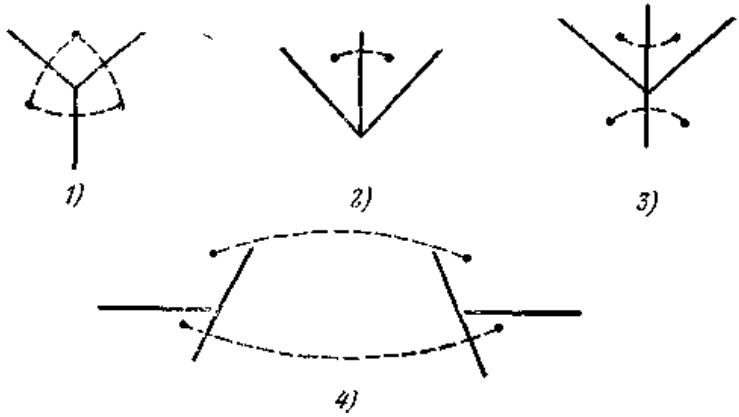


Рис. 12.25. Иллюстрация четырех эвристических правил объединения областей.

Приведенные правила не являются достаточными, так как были сформулированы только на основании совсем простых примеров. Кажется очевидным, что на основании только одной связки не следует объединять области. Целесообразно проанализировать отношения между всеми связками и только на основании этой информации объединять области.

Проиллюстрируем этот подход на примере сцены, содержащей усеченную пирамиду, лежащую на плите (рис. 12.26, *а).*

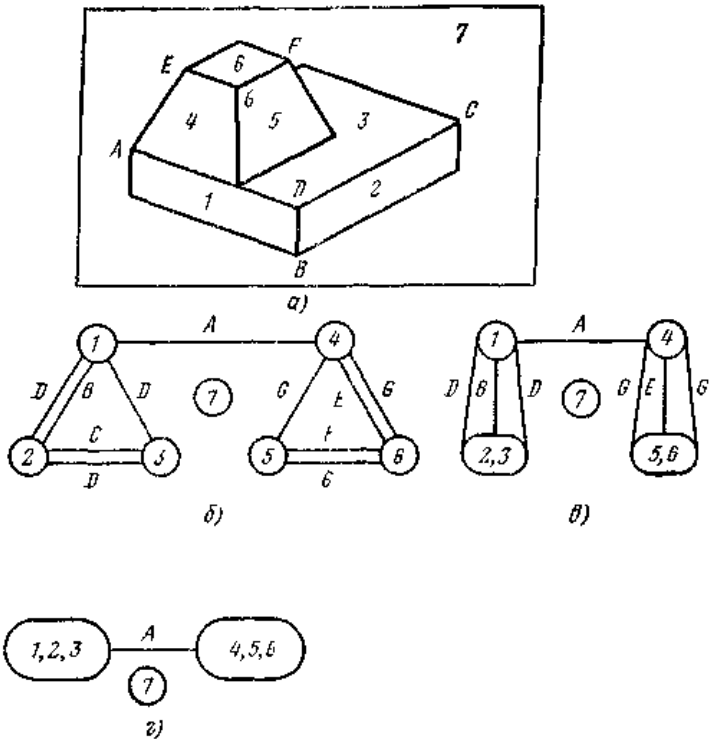


Рис. 12.26. Пример работы алгоритма объединения областей.

На рис. 12.26, *б* в виде графа представлена информация о связи областей сцены, изображенной на рис. 12.26, *а*. Узлы графа соответствуют областям изображения, а дуги — связкам, полученным с помощью применения приведенных ранее эвристических правил к некоторым вершинам (дугам присвоены имена этих вершин). Например, применив к вершине *D* эвристическое правило 1), получим связку между вершиной *1* и *2* (помеченную буквой *D)* и связки между вершинами *2* и *3, 1* и *3* (помеченные буквами *D).* Анализ рис. 12.26, *б* подтверждает подозрение, что одной связки не достаточно для объединения областей. Действительно, области *1* и *4* являются различными, но связываются связкой, полученной при обработке вершины *A*. Для того чтобы устранить указанное несоответствие, добавляется следующее эвристическое правило:

5) два узла объединяются, если между ними существует по крайней мере две связки. Любые связки из этих двух узлов к другим узлам остаются во вновь полученном графе.

Рис. 12.26, *в* изображает граф, полученный применением данного правила к узлам *2* и *3*, а затем к узлам 5 и *6* графа, представленного на рис. 12.26, *б*. Применив это же правило к узлам *1* и *(2, 3)* и узлам *4* и (5, *6)* графа, изображенного на рис. 12.26, *в*, получим окончательный вид графа рис. 12.26, *г*. К этому графу правило применить нельзя, так как никакие узлы не связаны двумя дугами. Таким образом, на рис. 12.26, *а* представлено три объекта *(1, 2, 3), (4, 5, 6)* и 7. Изложенный метод работает довольно успешно при распознавании многих сцен. Однако, можно привести примеры изображений, где этот метод дает неправильный результат. Например, фигура, изображенная на рис. 12.20, *а,* не будет опознана как одно тело, так как области *1* и *4* объединяются только одной связкой. Это несоответствие можно устранить, введя очередное эвристическое правило:

6) если в исходном графе некоторая вершина (область) связана только с одной вершиной, то эти вершины объединяются.

Однако и после введения этого правила можно привести пример (рис. 12.27), когда изображение будет неверно разделяться на отдельные объекты.

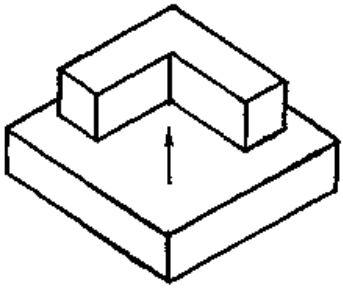


Рис.12. 27. Пример неверного объединения двух тел в одно на основе правил 1—6.

Рассмотренные выше методы не учитывали некоторых свойств физических тел и в первую очередь наличие теней. Обобщение изложенного выше способа разметки линий позволяет применить его для описания сцен, содержащих тени. Уолтц ввел одиннадцать типов линий (вместо трех, рассмотренных нами) и построил каталог, содержащий более тысячи возможных способов сочетания линий в вершинах (аналогичный каталогу, представленному на рис. 12.19). Процедура выделения тел, разработанная Уолтцом на основе

данного каталога, работает очень быстро, несмотря на большое количество потенциальных способов разметки вершин. Дело в том, что одновременно с увеличением числа способов разметки вершин увеличивается и количество ограничений, накладываемых на связанные вершины, что резко сокращает поиск. На рис. 12.28 приведен пример сцены, с которой программа Уолтца справляется довольно легко.

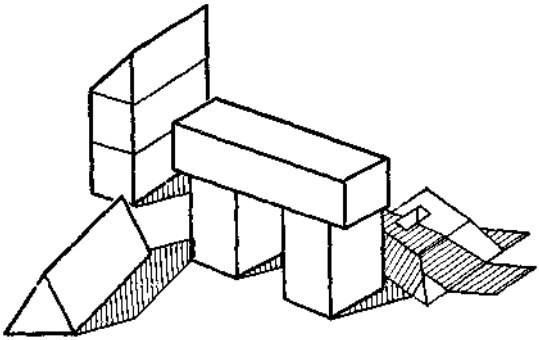


Рис. 12.28. Сцена с тенями, обрабатываемая программой Уолтца.

Подводя итоги перечисленным приемам, следует отметить, что они являются весьма общими в классе многогранников. Они не требуют ни того, чтобы многогранники ограничивались степенью три, ни какой-либо другой дополнительной информации о специфике объектов сцены. Однако эти методы не являются полными, т. е. они совершают ошибки при анализе некоторых сцен. Вообще говоря, довольно трудно характеризовать класс сцен, на которых эти методы работают безошибочно.

Цель применения описанных методов состоит в том, что после их работы сцена разлагается на тела, и это упрощает задачу опознающей программы. Действительно, теперь ей вместо осмотра всей сцены и поиска на ней участков, совпадающих с моделями, необходимо осуществлять сравнение только с выделенными областями, представляющими отдельные тела. Прежде чем перейти к описанию методов опознавания, мы рассмотрим в следующем параграфе способы получения пространственных характеристик выделенных объектов, также дающих дополнительную информацию для методов опознавания.

12.2.5. Монокулярное определение трехмерной структуры сцены

Определение трехмерной структуры видимой части объектов является важной частью анализа сцены. Оно необходимо как для опознавания объектов, если их модель является трехмерной, так и для образования визуального отображения понятия. Эта проблема в живой природе решается с помощью бинокулярного зрения. Однако при наличии двух изображений возникает задача отыскания на них идентичных пар точек.

В общем случае эта проблема не может быть решена по одному двумерному изображению. Однако при принятом предположении (сцена состоит только из многогранников) и благодаря некоторым дополнительным фактам можно получить информацию о третьем измерении.

Изобразим преобразование, выполняемое камерой в виде схемы, приведенной на рис. 12.29.

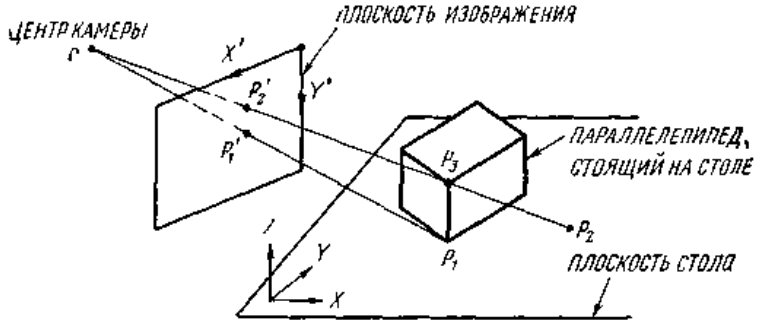


Рис. 12.29. Преобразование, выполняемое камерой.

Для любой точки $P_j=(X_j, Y_j, Z_j)$ реального мира существует единственная соответствующая ему точка изображения $P'_j = (X'_j, Y'_j)$. С другой стороны, каждой точке P'_j соответствует луч, проходящий через эту точку и центр камеры. Если камера соответствующим образом откалибрована, то каждой точке P'_j можно поставить во взаимнооднозначное соответствие некоторую точку P_j на поверхности стола. Она определяется как точка пересечения луча, исходящего из точки C, с плоскостью стола. Если мы примем естественную гипотезу, что объект, изображенный на рис. 12.29, опирается на плоскость стола, а не висит в воздухе, то мы можем по точке P'_1 однозначно определить координаты точки P_1.

Отметим, что если известна некоторая информация об объекте, например, такая, как размер ребра P_1P_3 или «ребро P_1P_3 вертикально поверхности стола» или «одна из образующих его граней вертикальна», то можно определить координаты точки P_3. Предположим, что об объекте, изображенном на рис. 12.23, нам известны координаты точек *A, F, E* и *G*. Так как три точки определяют положение плоскости, мы можем по известным координатам *A, F, G* определить положение плоскости *1* в трехмерном пространстве. Теперь становится возможным определить положение точки *B* как пересечение плоскости *1* с лучом, исходящим из центра камеры и проходящим через точку *B*. Подобным образом можно определить координаты точки *D* на основании известных координат точек *G, F, E*. Теперь нам известны координаты точек *B, G, D* и, следовательно, можно определить положение в пространстве плоскости 2, а по ней и координаты точки *C*. Итак, для изображенного на рис. 12.23 многогранника по координатам четырех точек *A, F, G* и *E* можно определить координаты оставшихся вершин, а следовательно, определить трехмерную структуру видимой части многогранника.

Формализуем приведенные выше рассуждения. Для простоты рассуждений будем полагать, что начало координат расположено в центре камеры. Тот факт, что точка *(x, y, z)* лежит в плоскости *v*, может быть выражен уравнением $V \cdot P = 1$, где *V* является вектором, перпендикулярным *v*, исходящим из начала координат и имеющим длину, обратную расстоянию от начала координат до плоскости, а *P* — вектор, исходящий из начала координат в точку *(x, y, z)*. Вектор *V* характеризует плоскость *v*. Если известно расположение луча, на котором лежит точка *P*, то это эквивалентно знанию положения единичного вектора *U* в направлении *P* и расстоянию α от начала координат до *P*. Учитывая вышеизложенное, можно записать, что

$$V \cdot \alpha\, U = 1 \quad \text{или} \quad V \cdot U = 1/\alpha = \lambda.$$

Применим этот анализ к изображению, представленному на рис. 12.23. Будем обозначать через V_i векторы, характеризующие плоскости граней *1, 2* и *3*, а через U_k — единичные векторы, направленные в вершину *K*. Тогда для плоскостей *1, 2* и *3* можно записать следующие уравнения:

$V_1 \cdot U_A = \lambda_A$, $V_2 \cdot U_B = \lambda_B$, $V_3 \cdot U_G = \lambda_G$,
$V_1 \cdot U_F = \lambda_F$, $V_2 \cdot U_C = \lambda_C$, $V_5 \cdot U_F = \lambda_F$,
$V_1 \cdot U_G = \lambda_G$, $V_2 \cdot U_G = \lambda_G$, $V_3 \cdot U_E = \lambda_E$,
$V_1 \cdot U_B = \lambda_B$, $V_2 \cdot U_D = \lambda_D$, $V_3 \cdot U_D = \lambda_D$.

Итак, мы имеем 12 линейных уравнений и 16 неизвестных *(λ_A, λ_B, λ_C, λ_D, λ_E, λ_F, λ_G* и 9 неизвестных, определяющих плоскости *1,2* и *3)*. Если уравнения линейно независимы, то однозначное решение может быть

получено, если будут зафиксированы любые четыре независимые переменные. В общем случае можно сказать, что мы имеем:

K = (число неизвестных) = 3× (число плоскостей)+
+ (число различных точек изображения).

M = (число уравнений) = $\sum_{\substack{\text{по всем} \\ \text{плоскостям}}}$ (число различных точек плоскости).

Таким образом, $L=K—M$ равно наименьшему числу точек изображения, информация о которых должна быть известна для того, чтобы однозначно определить трехмерную структуру многогранника.

Если на объекты сцены наложить ограничения, что они являются только многогранниками степени три, то можно получить интерпретацию описанной выше процедуры. Интерпретация позволяет последовательно определять местоположение в пространстве видимых граней объекта, что по сравнению с решением системы линейных уравнений дает возможность использовать информацию, получаемую в процессе вычислений, для направления вычисления. Интерпретация основывается на определенного рода дуальном графе, описывающем изображение многогранника. Узел дуального графа соответствует видимой грани многогранника, а дуга между двумя вершинами (гранями) соответствует ребру многогранника, разделяющему смежные грани (т. е. ребру, являющемуся общим для двух граней). Если две грани разделяются более чем одним ребром (все ребра толжны быть коллинеарны), то на графе изображается только одна дуга. Заметим, что из приведенного ранее метода разметки можно определить, является ли ребро общим для двух смежных, видимых граней. Действительно, если ребро помечено знаками «+» или «—», то оно является общим, если же оно помечено знаком → или ←, то оно не является общим для двух граней (одна грань закрывает другую).

На рис. 12.30, *а* и 12.30, *б* представлен многогранник и соответствующий ему дуальный граф.

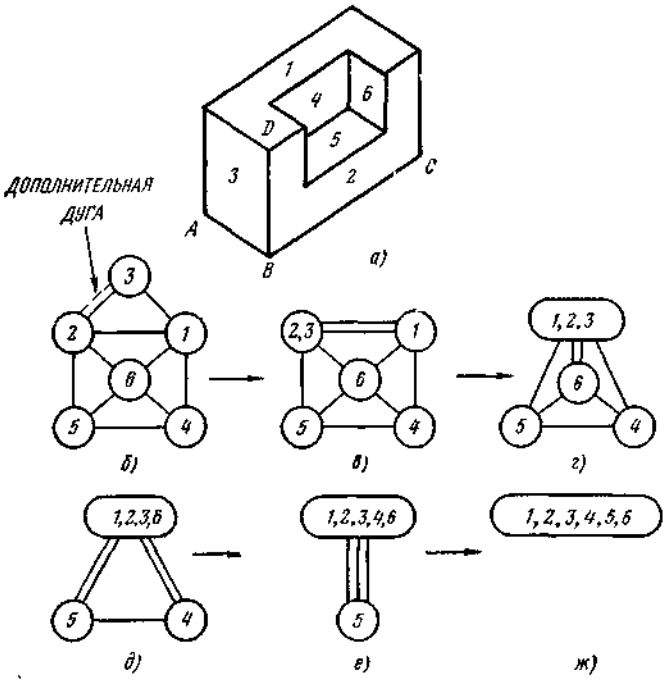


Рис. 12.30. Объединение дуг дуального графа с введением одной дополнительной дуги.

Покажем, как дуальный граф используется для определения последовательности перехода от одной грани многогранника к другой для определения его пространственной структуры. Предположим, что известны координаты вершин *A, B, C, D* (рис. 12.30, *а),* т. е. известно расположение в пространстве плоскостей 2 и *3,* Поверхности, имеющие известное местоположение в пространстве, будем объединять в одну вершину (см. рис. 12.30, *в),* связывая с ней все дуги, которые входили в объединенные вершины. Для того, чтобы инициировать процесс, введем дополнительное ребро между гранями, положение которых в пространстве известно (см. рис. 12.30, б). Из рис. 12.30, *в* видно, что грани 2 и 3 с известным местоположением связаны двумя дугами с гранью *1.* Следовательно, грань *1* граничит с гранями *2* и *3* по двум неколлинеарным ребрам, которые однозначно определяют положение грани *1.* Подобные рассуждения обосновывают следующее <u>*правило преобразования графа*</u>: *объединение в одну двух вершин, связанных более, чем одной дугой.* Применяя это правило многократно

к дуальному графу, изображенному на рис. 12.30, б, получим единственную вершину. Это обозначает, что для рассматриваемой фигуры четырех точек достаточно, чтобы определить пространственную структуру всей фигуры. В общем случае четырех точек может быть недостаточно для определения пространственной структуры тела. С точки зрения приведенного выше преобразования это будет означать, что к дуальному графу надо будет присоединить более чем одну дополнительную дугу, чтобы преобразовать граф к одной вершине. Другими словами, *для однозначного описания видимой пространственной структуры объекта необходимо определить (например с помощью дальномера) трехмерные координаты еще одной точки объекта.* Нетрудно показать, что пространственная структура видимой части многогранника может быть определена с помощью K+3 независимых известных точек, где K — количество добавляемых ребер. На рис. 12.31 приведен пример фигуры и соответствующего ей дуального графа, требующей для определения пространственной структуры пяти точек.

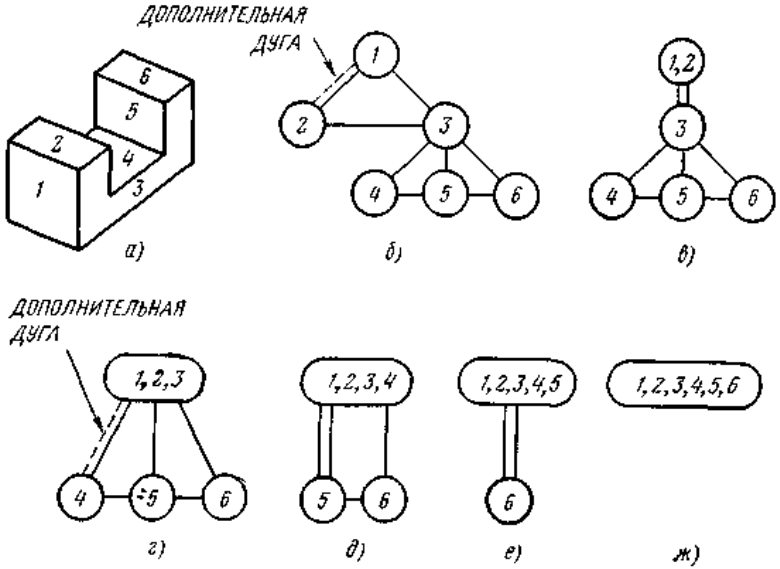


Рис. 12.31. Объединение дуг дуального графа с введением двух дополнительных дуг.

В п.12.2.1 предъявлены некоторые требования к структуре комплекса программ, осуществляющего восприятие визуальной информации. Мы отмечали, что последовательность вызова программ комплекса не должна быть фиксированной и заранее предопределенной, она должна определяться анализируемой сценой. В качестве примера рассмотрим сцену, представленную на рис. 12.32.

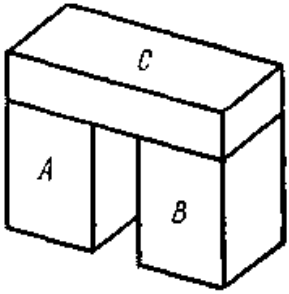


Рис. 12.32. Определение глубины многогранника *A* на основании глубины многогранника *B*

Здесь ребро, определяющее глубину многогранника *A*, не видно, и если полная модель многогранника *A* неизвестна программе, то на основании описанных выше методов длину ребра не определить. Человек при анализе этой сцены сделает естественное предположение, что многогранники *A* и *B* одинаковы и на основании этого определит размер невидимого ребра.

Финин предложил подход к решению подобных задач. Предложенную им процедуру удобно определить в терминах циклического выполнения последовательности: **«группирование—гипотеза—проверка».** Рассмотрим работу процедуры анализа на примере сцены, изображенной на рис. 12.32. Процедура пытается определить размеры объекта *A* и выясняет, что глубина его не видна из изображения. Тогда она относит объект *A* к какой-либо группе объектов. В рассматриваемом примере *A* и *B* могут быть отнесены к группе объектов, поддерживающих объект *C*. Затем процедура проверяет, существуют ли причины, мешающие отнести объекты *A* и *B* к одной группе? При этом процедура рассматривает примерно такие вопросы:

1. Относятся ли *A* и *B* к одному сорту объектов?
   Да, оба являются брусками.
2. Одинаковым ли образом ориентированы объекты *A* и B?
   Да.
3. Соответствуют ли видимые размеры объектов?
   Да.

4. Существуют ли причины сомневаться в том, что невидимое ребро объекта *A* отличается от аналогичного ребра объекта *B*?

Нет.

Если ответы аналогичны приведенным, то гипотеза применяется и анализ сцены продолжается.

Отметим, что группирование объектов является одним из способов использования контекста при анализе сцены.

Основанием для высказывания гипотезы о принадлежности объектов к одной группе может быть, например, следующая информация.

1. Объекты образуют стеки и ряды, связанные отношением «поддерживать» или «находиться перед».
2. Объекты выполняют общую функцию, например, колонны у арки или ножки у стола.
3. Объекты находятся в непосредственной близости.
4. Объекты относятся к одному типу.

Проверка гипотез осуществляется не только на основе измерения различий между объектами, но также на основе измерения, какие отклонения являются типичными и какие приемлемыми.

Описанная процедура требует дальнейшего уточнения и развития, однако общий подход заслуживает внимания.

12.2.6. Формирование моделей, опознавание объектов и описание сцены

На основании методов, аналогичных описанным в предыдущих параграфах, и некоторой априорной информации об объектах можно составить описание сцены.

Рассмотрим на примере сцены, изображенной на рис. 12.33, *а,* процесс получения ее описания.

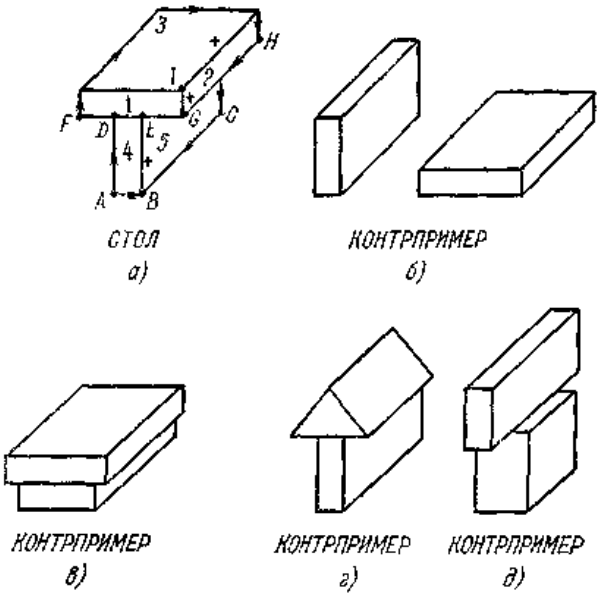


Рис. 12.33. Последовательность примеров, определяющих понятие СТОЛ.

Будем полагать, что предьявляемые сцены содержат только прямоугольные параллелепипеды (для краткости будем называть плитами). Столь жесткое ограничение на вид используемых объектов не отражает реальных возможностей машинных методов обработки изображений и введено исключительно с целью привести компактный пример. Будем также полагать, что программе, обрабатывающей изображение сцены, известны следующие понятия: СТОЯТЬ, ЛЕЖАТЬ, ПОДДЕРЖИВАТЬ. Например, программа может считать, что объект стоит, если его высота больше одного из горизонтальных размеров и объект лежит, если указанное условие не выполняется. Программа может определять понятие ПОДДЕРЖИВАТЬ как отношение одного объекта находиться под другим (при отсутствии других объектов между ними).

Как и прежде, будем основываться на гипотезе об опорной плоскости, т. е. будем считать, что любой объект поддерживается либо горизонтальной плоскостью стола, либо горизонтальной плоскостью другого объекта. В результате работы 1 и 2 этапов (см. п. 12.2.1) подсистемы зрительного восприятия изображение предъявленной сцены будет описано в терминах линий и областей. Применение к

полученному описанию алгоритма разметки линий даст результат, изображенный на рис. 12.33, *а.* На основании проведенной разметки линий, эвристические правила разбиения сцены на объекты обнаружат присутствие на сцене двух объектов: *(1, 2, 3)* и *(4, 5).*

На анализируемом изображении часть объекта, образованного гранями *4* и 5, закрыта объектом, составленным из граней *1, 2* и *3.* Поэтому при разбиении сцены на объекты объект *(4, 5)* будет представлен в виде неполного контурного рисунка. Для упрощения процесса опознавания можно применять эвристические процедуры, достраивающие неполные контурные рисунки на основании некоторых знаний о свойствах объектов. В рассматриваемом примере в связи с его простотой (все объекты являются плитами) в этом нет необходимости.

После этапа достраивания неполных контурных рисунков следует этап опознавания объектов и определения взаимоотношений между объектами. Для сопоставления трехмерной модели объекта с его двумерным изображением необходимо получить по контурному рисунку заключение о трехмерных свойствах сцены. В п. 12.2.5 рассмотрена возможность использования гипотезы об опорной плоскости для определения этой информации.

В рассматриваемом примере знание, что точки *A, B* и *C* лежат на известной опорной плоскости, дает возможность определить их трехмерные координаты.

В связи с тем, что плита является прямоугольным параллелепипедом, плоскость, в которой лежит грань *4,* перпендикулярна опорной плоскости, и, следовательно, ее положение в пространстве может быть определено. Если положение грани *4* известно, то можно определить координаты точек *D* и *E* как пересечение луча, исходящего из центра камеры, с плоскостью, в которой лежит грань *4.* На основании подобных рассуждений могут быть определены координаты точек *F, G, H, I.* Знание координат точек *A, B, C, F, G, H, I* дает возможность вычислить длины сторон параллелепипедов и сделать заключение, что плита *(4, 5)* находится в состоянии СТОЯТЬ (так как *AD>AB),* плита *(1, 2, 3)* находится в положении ЛЕЖАТЬ (так как *GI<FG* и *GI<GH),* и плита *(4, 5)* находится с плитой *(1, 2, 3)* в отношении ПОДДЕРЖИВАТЬ (так как объект *(4, 5)* расположен под объектом *(1, 2, 3)).* Итак, на основании собранной информации описание сцены может быть представлено в виде семантической сети, изображенной на рис. 12.34.

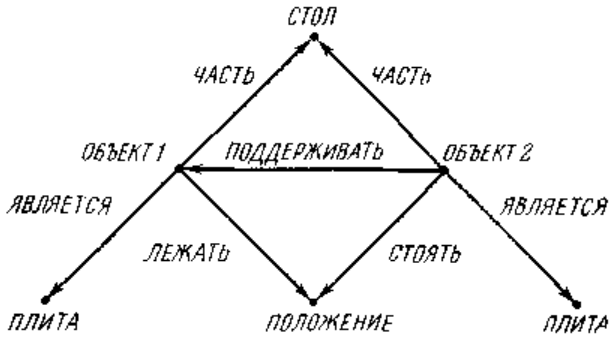


Рис. 12.34. Описание понятия СТОЛ.

Покажем теперь, как понятия (модели), существующие в системе, могут быть использованы для образования новых, более сложных понятий.

Процесс образования понятия заключается в предъявлении машине последовательности сцен, характеризующих как собственно понятие, так и понятия, подобные образовываему, но отличающиеся от него в одной или нескольких принципиальных деталях (контрпримеры).

Мы будем рассматривать образованное понятие как модель, включающую описание обязательных признаков и описание признаков, которые не должны присутствовать в примерах, удовлетворяющих данному понятию.

На рис. 33 приведена последовательность сцен, определяющих образовываемое понятие СТОЛ. Программа анализа изображения при предъявлении первой из сцен, определяющей собственно понятие, составляет описание сцены в терминах известных ей понятий и отношений.

Примерный вид полученного описания приведен на рис. 12.34.

В определении стола существенным отношением является отношение поддержки, так как не существует столов без этого отношения между составляющими его объектами. Следовательно, цель примеров — показать машине существенные признаки понятия. (Позже мы увидим, что некоторые отношения становятся менее существенными, а другие запрещенными.)

Второй пример на рис. 12.33 является контрпримером. Его единственное отличие от первого примера заключается в том, что одна плита не поддерживает другую. Описание этого изображения отличается от описания первого примера отсутствием отношения ПОДДЕРЖИВАТЬ между плитами. Программа, сравнивающая

описания, обнаружит это единственное различие и сделает вывод, что именно отсутствие этого отношения привело к невозможности отнести второй пример к понятию СТОЛ. Таким образом, программа помечает отношение ПОДДЕРЖИВАТЬ как существенное заменой ею в описании понятия на отношение ДОЛЖЕН ПОДДЕРЖИВАТЬ. Заметим, что мы сформировали новое отношение на основании всего одного примера, а не на основании статистических выводов из серии опытов

Остальные контрпримеры (рис. 12.33) выполняют аналогичную роль. Под их воздействием будет получена модель понятия СТОЛ (см. рис. 12.35).

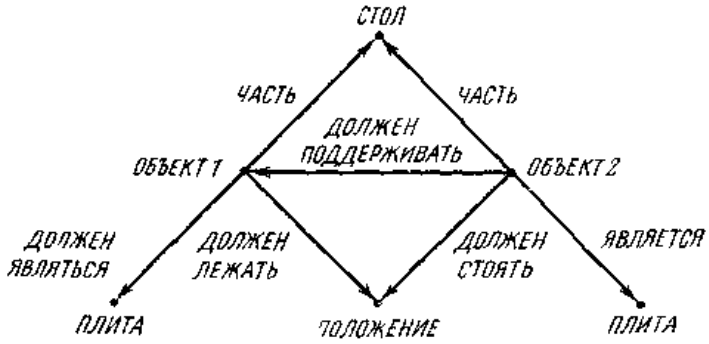


Рис. 12.35. Модель понятия СТОЛ.

Теперь, когда основная идея ясна, рассмотрим более сложную последовательность примеров (рис. 12.36), определяющую понятие АРКА.

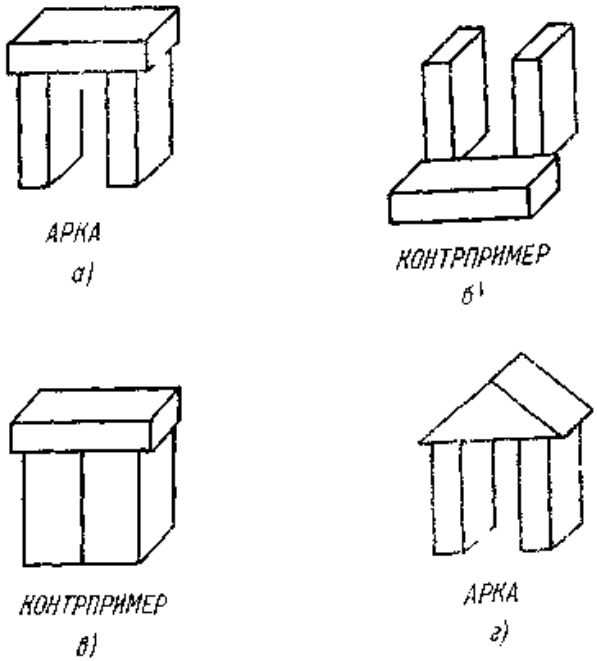


Рис. 12.36. Последовательность примеров, определяющих понятие АРКА.

Первый пример последовательности дает, как и ранее, образец образовывемого понятия, на основании которого машина составляет начальное описание понятия. Следующий пример демонстрирует необходимость отношения ПОДДЕРЖИВАТЬ для понятия АРКА. В отличие от примера на рис. 12.33, *б,* различие касается отсутствия не одного, а двух отношений ПОДДЕРЖИВАТЬ (для каждой из опор арки). Этот пример поднимает сложный вопрос: что делать, если контрпример отличается от описания образованного понятия более, чем в одном отношении или вершине. Для решения этой проблемы необходимо разработать теорию, которая рассортировывает наблюдаемые различия по важности и использует наиболее интересные для формирования гипотез о модификации описания образовывемого понятия. В связи с отсутствием в настоящее время подобной теории перечислим некоторые правила, пригодные для упорядочения различий.

1. Важность различий убывает по мере удаления от начального (верхнего) узла описания, соответствующего нулевому уровню сети Так, различия в отношении между объектами более важны, чем различия в форме объектов, которые в свою очередь более важны, чем различия в затемнении вершин
2. Различия в пределах данного уровня упорядочиваются в соответствии с их типом. Например, различие, состоящее в отсутствии отношения, связывающего две вершины, можно рассматривать как более важное, чем различие в появлении дополнительного отношения.
3. Различия, имеющие одинаковый уровень и тип, могут быть упорядочены на основе некоторых вторичных эвристик. Например, отношение ПОДДЕРЖИВАТЬ зачастую более важно, чем отношение КАСАТЬСЯ или СПРАВА, СЛЕВА.
4. Если два различия имеют одинаковое описание и сущность, то целесообразно считать, что они оба являются причинами отнесения образца к контрпримеру.

Последнее правило ориентировано на обработку образцов, аналогичных контрпримеру, приведенному на рис. 12.36, *б*.

Вернемся к рассмотрению последовательности, определяющей понятие АРКА (рис. 12.36). Очередной контрпример (рис. 12.36, *в)* несет косвенную информацию о необходимости расстояния между двумя опорами арки. Описание данного контрпримера имеет наиболее важное различие с текущим описанием понятия АРКА в наличии дополнительных отношений КАСАТЬСЯ (для каждой из опор). Машина в состоянии сделать заключение, что именно поэтому контр-пример не соответствует понятию АРКА, и выработать в качестве обязательного условия для модели наличие между опорами отношения ДОЛЖНЫ НЕ КАСАТЬСЯ. Это не даст возможность программе идентификации объектов отнести к понятию АРКА объект, имеющий указанное запрещенное отношение.

Итак, мы показали способ, с помощью которого в модель может быть введена как информация, указывающая на необходимость некоторого признака, так и информация, отрицающая наличие некоторого признака.

Рассмотрим теперь, каким образом в модель может быть включена некоторая информация об обобщениях. Поясним суть подхода на примере рис. 12.36, *г,* который указывает на возможность в определении понятия АРКА использовать вместо объекта ПЛИТА объект КЛИН. Простейший способ введения в модель указанного обобщения состоит в добавлении к сети нового понятия, используемого вместо вершины ПЛИТА, соединив вершины ПЛИТА и КЛИН с новым понятием отношением ПОДМНОЖЕСТВО (SUB).

Рассмотрим другой подход к решению этой задачи. На рис. 12.37 приведено описание понятий ПЛИТА и КЛИН и часть общих знаний, имеющихся у робота и связывающих данные понятия.

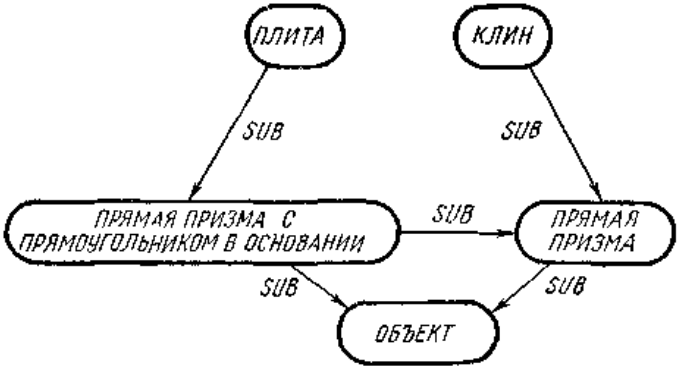


Рис. 12.37. Связь понятий ПЛИТА, КЛИН и ОБЪЕКТ. Все вершины, связаны отношением ПОДМНОЖЕСТВО (SUB).

Из рисунка видно, что первым понятием, включающим множество плит и клиньев, является понятие ПРЯМАЯ ПРИЗМА. На основании этого мы можем заменить в модели вершину ПЛИТА на вершину ПРЯМАЯ ПРИЗМА.

Так как перечисленные выше соображения по образованию модели являются гипотезами, предлагаемыми машиной, то необходим механизм проверки этих гипотез и их корректировки. Ошибки могут быть обнаружены, когда последующие примеры, определяющие понятие, будут противоречить сделанным ранее предположениям. Например, если образец на рис. 12.38, *а* определяет понятие *X*, а рис. 12.38, *б* определяет контрпример, то будет сформировано предположение, что *X* ДОЛЖЕН СТОЯТЬ.

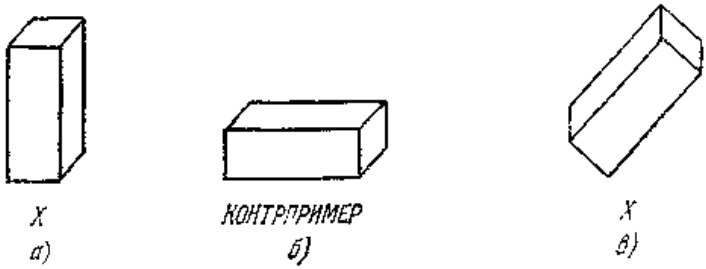


Рис. 12.38. Последовательность примеров, определяющих понятие *X*.

Отметим, что в принципе на основании, например, вторичных эвристик (см. п. 3 правил упорядочения различий) возможно сформировать альтернативное предположение: Х НЕ ДОЛЖЕН ЛЕЖАТЬ. Если мы хотим навязать вторую интерпретацию, то для этого необходимо дать в качестве примера наклонный брусок (рис. 12.38, *в)*. Описание, составленное машиной для наклонного бруска, не имеет отношения СТОЯТЬ и, следовательно, противоречит исходному описанию (рис. 12.38, *а)*. При этих условиях система должна вернуться к предшествующему описанию и рассмотреть альтернативные варианты.

Следует отметить, что в описанном процессе образования модели успех во многом зависит от выбранной последовательности примеров, определяющих понятие. Это не является недостатком предлагаемого процесса, так как и обучение человека в значительной степени зависит от разумного подбора примеров. Более того, именно пренебрежение этим свойством приводило к неудаче многие методы обучения.

Описанный способ образования модели понятия не является универсальным, поэтому при работе со сложными объектами используется интерактивный способ формирования новой модели (понятия). Суть его состоит в следующем. Человек-оператор предъявляет пример нового объекта на экране дисплея и указывает системе его отличительные признаки. Система выделяет из предъявленного примера в первую очередь легко извлекаемые признаки и образовывает на их основе модель нового объекта. Затем система выводит на дисплей изображения, соответствующие новой модели и существовавшим ранее моделям, похожим на нее. Оператор, рассматривая эти изображения, указывает системе на признаки наиболее выжные для различения объектов. Он обучает систему извлекать эти признаки, пометив на экране их местоположение и указав на существующие (или разработав новые) процедуры выделения признаков. После того как указанные признаки извлечены, а модель модифицирована, система сравнивает модифицированную модель с существующими, выделяет похожие, оператор указывает на дальнейшие детали и так до тех пор, пока новая модель понятия не будет отличаться от существующих.

Отметим, что включение в репертуар системы нового объекта может потребовать уточнения некоторых существовавших ранее моделей.

Рассмотрим теперь использование вновь образованной модели понятия в процессе опознавания. Задача опознавания может быть определена довольно разнообразно. Рассмотрим следующие разновидности процесса опознавания:

1) опознавание предъявленной сцены;
2) опознавание всех понятий, представленных на сцене;
3) определение присутствия на сцене примера некоторого понятия.

Первая из указанных задач, заключающаяся в опознавании всей сцены, является простейшей.

Модель понятия соответствует анализируемой сцене, если

а) все отношения, помеченные в модели указателем ДОЛЖЕН, присутствуют в описании сцены;

б) все отношения, помеченные в модели указателем НЕ ДОЛЖЕН, не присутствуют в составленном описании;

в) различия, отличающиеся от указанных в пунктах а) и б), являются устранимыми (см. п.12.2.3).

Если сцена содержит много идентифицируемых рядов, стеков или других групп объектов, то необходимо модифицировать программу опознавания в связи с тем, что существенные отношения могут отсутствовать, так как одни объекты могут закрывать другие. Здесь возможно приписывание объекту некоторых признаков на основании признаков групп, в которую он входит.

Последняя задача, состоящая в поиске на сцене некоторого понятия, является наиболее трудной. Прямолинейный подход состоит в сопоставлении полученного описания сцены и модели опознаваемого понятия. Если при этом некоторые объекты сцены сопоставляются с соответствующими частями модели, то отношения, связывающие эти объекты с посторонними объектами сцены, устраняются и выполняется обычная процедура опознавания. В указанном процессе предстоит решить много задач, связанных с тем, как в процессе сопоставления выбирать на основе контекста правильный участок сцены. После того, как понятия опознаны, мы можем получить описание предъявленной сцены в терминах новых более сложных понятий.

12.2.7. Образовывающие операторы понятий

Теория понятий, позволяющая формализовать и ускорить процесс образования понятий, намного уменьшить вероятность неоднозначного толкования понятия и т. п., зарождается в период развития информационных технологий. Поэтому в настоящее время в теории понятий основное внимание уделяется методам синтеза структуры моделей понятий с использование тех или иных языковых средств. Использование ЭВМ в качестве операторов, которые могут обеспечить реализацию методов формализованного определения и образования

понятий, привело к тому, что широкое распространение получает программная реализация алгоритмов образования понятий. В последнее время все чаще появляются работы, в которых развиваются методы синтеза программно-лингвистической реализации алгоритмов образования понятий, когда одна часть алгоритма образования понятий реализуется программно, а вторая — лингвистически (описательно).

В настоящей работе мы попытались учесть эти тенденции развития теории понятий и ее практического применения при создании и использовании систем формирования фреймов, однако основное внимание сосредоточили на базисных положениях теории понятий и изложили их с учетом результатов, полученных к настоящему времени.

В работе приведено понятие процесса образования понятия, в частности, параллельного процесса. При этом процесс образования понятия рассматривается в качестве исходного условия для задания алгоритма образования понятия.

Образовывающий оператор понятия (ООП) рассматривается как средство, реализующее алгоритм образования понятия, определяющий порядок выполнения отдельных операций или процедур по образованию некоторго понятия — понятия предмета (ПП).

ООП, в период своей работы, в соответствии с алгоритмом образования понятия, осуществляет реализацию последовательных процедур образования понятия, выполняющих процесс образования ПП. При этом последовательность выполняющих ООП процедур зависит от состояния самого ПП и внешнего сигнала *R*, который может быть подан извне, например от другого ООП или от человека.

Совокупность взаимосвязанных ООП и ПП образует систему ООП—ПП (рис. 12.39).

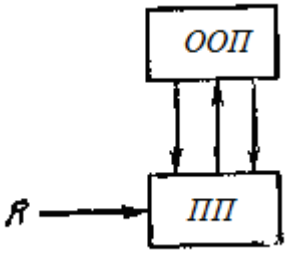


Рис. 12.39

В качестве системы ООП—ПП можно рассматривать обычную ЭВМ, в которой ООП является процессор, а предметом понятия — запоминающее устройство с хранящейся в нем обрабатываемой информацией.

Если есть необходимость более детально рассмотреть функционирование самого процессора ЭВМ в виде системы ООП—ПП, то в качестве ООП рассматривается центральный блок (устройство) управления (ЦБУ), а в качестве ПП—арифметическо-логическое устройство (АЛУ).

Как правило, сложные понятия состоят из отдельных блоков *Б* (рис. 12.40).

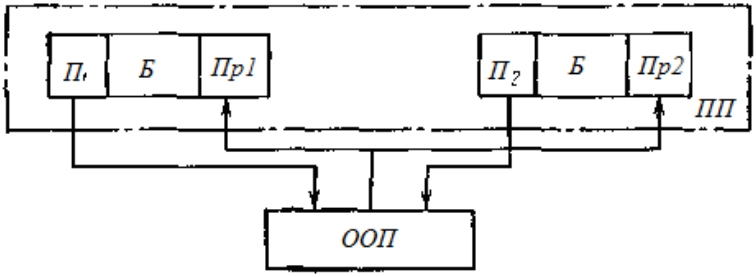


Рис. 12.40

При этом каждый блок сложного понятия обычно содержит один или несколько *признаков предметов*, которые лежат в основе образования понятия, и параметры (сигнализаторы) признаков. Например, если в качестве блока предмета понятия рассматривается некоторый резервуар для жидкости, то в качестве одного отличительного признака $П_1$ предмета может служить привод вентиля для подачи в резервуар жидкости, а в качестве $П_2$ — привод вентиля для слива жидкости из резервуара. Параметрами признаков $Пр_1$ и $Пр_2$ такого блока ПП служат сигнализаторы верхнего и нижнего предельных уровней жидкости в резервуаре.

В свою очередь, ООП может быть *одноблочным* и *многоблочным.*

В одноблочных ООП обычно реализуются достаточно простые алгоритмы образования понятий, а сами ООП включают небольшое число логических элементов, разделение которых по каким-либо признакам на отдельные группы и выделение, таким образом, нескольких блоков нецелесообразно, а в ряде случаев и невозможно. Примерами одноблочных ООП могут служить разнообразные операторы приема последовательности импульсов и преобразования ее в параллельную кодовую комбинацию импульсных или потенциальных сигналов.

При реализации достаточно сложного алгоритма образования понятий, если ООП становится довольно сложным, то его по тем или иным признакам разделяют на отдельные блоки. По функциональному

признаку делят, если в одном блоке многоблочного ООП реализуется одна или несколько тяготеющих друг к другу функций. Например, если для выполнения одной функции необходим достаточно большой объем данных, полученных в процессе выполнения второй функции.

Поэтому блоки многоблочного ООП будем называть *функциональными блоками* (ФБ). В дальнейшем для простоты будем считать, что в одном ФБ реализуется одна функция, которая определяется процедурой реализуемого в ООП алгоритма образования понятия.

Таким образом, в многоблочных ООП каждый из ФБ, выполняя одну функцию, реализует только отдельную часть атгоритма образования понятия (процедуру), которую будем называть *частным алгоритмом образования понятия* или *частным алгоритмом.*

Для того чтобы многоблочный ООП реализовал алгоритм образования понятия полностью, между ФБ должны быть управляющие связи (УС) C_{ij}, определяющие в соответствии с заданным алгоритмом порядок работы ФБ. Можно отметить четыре основных типа построения многоблочных ООП: 1) многоблочные с рассредоточенными управляющими связями (рис. 12.41); 2) многоблочные с концентрированными управляющими связями (рис. 12.42), которые кроме ФБ имеют центральный блок управления (ЦБУ); 3) многоблочные иерархические (децентрализованные) с *n* уровнями блоков управления (БУ), из которых ЦБУ является блоком уровня *n* (рис. 12.43); 4) многоблочные распределенные (рис. 12.44), в которых вместо одного ЦБУ имеется *r* взаимозависимых БУ, обеспечивающих в совокупности те же функции, что и ЦБУ в многоблочных ООП с концентрированными УС.

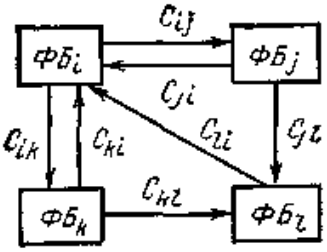


Рис. 12.41

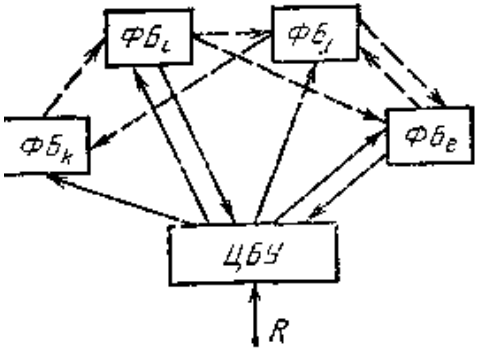


Рис. 12. 42

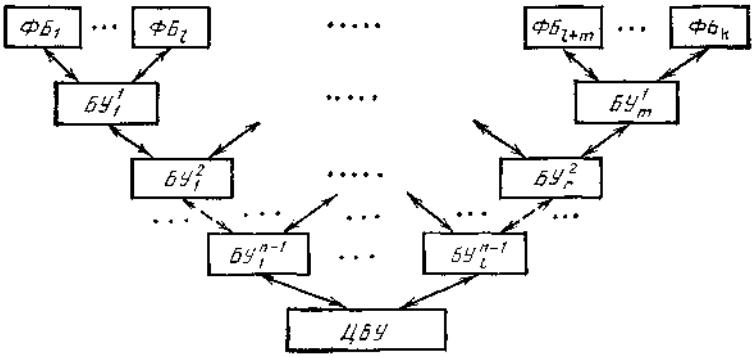


Рис. 12.43

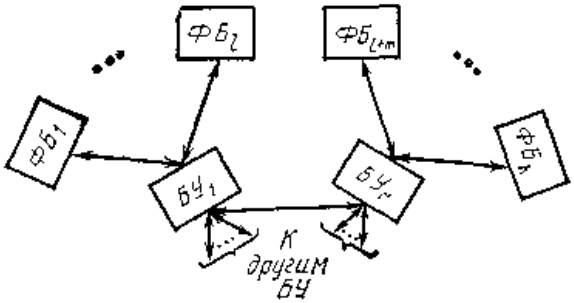


Рис. 12.44

Во всех четырех типах многоблочных ООП для передачи из одного ФБ в другой перерабатываемой информации (данных), необходимой для

обеспечения выполнения алгоритма образования понятия ООП, существуют *функциональные связи* (ФС). Для того чтобы не загромождать рисунки, ФС изображены только на рис. 12.42 пунктирными линиями. Таким образом, по функциональным связям передаются данные, а по управляющим связям — управляющая информация, т. е. команды для выполнения тех или иных процедур в ФБ. При этом следует заметить, что для передачи данных могут использоваться как выделенные функциональные связи, так и совмещенные с управляющими связями, т. е. в ООП создаются *функционально-управляющие связи.* Последнее особенно характерно для иерархических и распределенных ООП.

В одноблочпых ООП изменение алгоритма образования понятия требует изменения схемы ООП. В многоблочных ООП с рассредоточенными управляющими связями изменение алгоритма выполнения ФБ может быть осуществлено путем изменения управляющих связей. Такое изменение УС может быть осуществлено путем подключения их ко входам и выходам специальной кроссировочной гребенки, или, как говорят, *кроссовому коммутатору* (КК), где возможно изменять соединения между входами и выходами КК, т. е. осуществлять перекроссировку. На рис. 12.45 показана схема организации связи *ФБ*$_i$ и *ФБ*$_j$ через *КК,* где осуществлено соединение (кроссировка) выхода *ФБ*$_i$—*С*$_{ij}$, подключенного ко входу *КК,* с выходом *КК,* который соединен со входом *ФБ*$_j$ — *С*"$_{ij}$. Аналогично осуществлена кроссировка в *КК* выхода *ФБ*$_j$,—*С*'$_{ji}$ со входом *ФБ*$_j$-*С*''$_{ji}$:

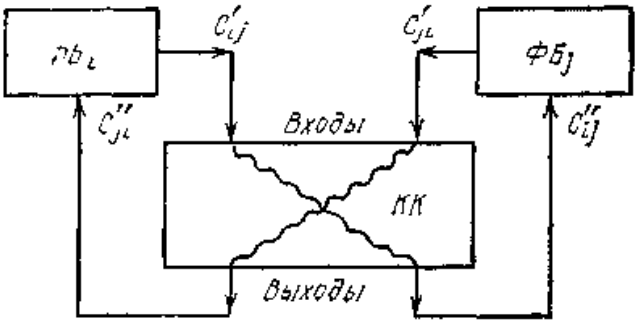


Рис. 12.45

Второй тип построения многоблочных ООП характеризуется отсутствием управляющих связей между отдельными ФБ и наличием одного центрального блока управления, в который от ФБ поступает уведомительная информация (информация об окончании работы ФБ, о полученных в ФБ значениях тех или иных параметров и т. д.), после

обработки которой ЦБУ вырабатывает управляющую информацию к соответствующим ФБ.

Таким образом, в ООП второго многоблочного типа порядок работы ФБ определяется выдаваемой ЦБУ в ФБ управляющей информацией в зависимости от поступающей в ЦБУ из ФБ уведомительной информации и заложенного в ЦБУ алгоритма управления.

Наличие ЦБУ, называемого в случае программной реализации его алгоритма образования понятия *блоком программного управления* (БПУ), позволяет довольно просто, без изменения всей структуры, заменять алгоритм его функционирования путем изменения структуры ЦБУ при аппаратурной реализации программы или программы, хранящейся в *оперативном* или *постоянном запоминающем устройстве* (ОЗУ или ПЗУ).

Многоблочные ООП второго типа, называемые *программными образовывающими операторами понятия,* могут быть использованы в различных системах образования понятий.

Следует заметить, что в том случае, если ЦБУ рассматривается в качестве управляющего блока процессора ЭВМ или электронной управляющей машины (ЭУМ), аппаратурная реализация частного алгоритма образования понятия, соответствующего выполняемой в ЭВМ команды, обычно используется в стационарных ЭВМ и ЭУМ. В мобильных ЭВМ получила наибольшее распространение микропрограммная реализация, сочетающая достоинства аппаратурной и программной реализаций. Программная же реализация частного алгоритма образования понятия в процессоре, рассматриваемого в качестве ЦБУ, применяется в основном лишь в качестве вспомогательного средства к аппаратурной или к микропрограммной реализации, позволяющего в ЭВМ и управляющих устройствах облегчить введение в систему команд специализированных, обычно сложных, команд с возможностью их простого изменения при образовании понятий.

Программный способ управления особенно удобен в том случае, если в процессе эксплуатации автоматической системы образованных понятий необходимо изменять режим уцравления понятиями. Закладывая в ЦБУ некоторый универсальный алгоритм функционирования с возможностью проверки поступающих извне значений параметров управления *R,* можно достаточно просто, особенно при микропрограммной и программной реализациях, осуществлять изменение алгоритма функционирования ООП.

В программном ООП все ФБ можно распределить на две группы. К первой группе относятся ФБ, в которых осуществляется проверка каких-либо условий. Эти ФБ, называываемые *логическими* (ЛФБ), после

завершения работы передают сигнал о значении проверяемого условия в ЦБУ. В качестве ЛФБ могут рассматриваться всевозможные средства, обеспечивающие снятие показаний значений параметров признаков.

В дальнейшем будут рассматриваться лишь такие ЛФБ, которые вырабатывают уведомительную информацию о выполнении (p_i= 1) или невыполнении (p_i = 0) проверяемого условия p_i. Обычно ЛФБ включаются сигналом из ЦБУ. Вместе с тем могут применяться ЛФБ, которые непрерывно контролируют проверяемый ими параметр. Однако всегда будем предполагать, что уведомительная информация от него в ЦБУ может поступать лишь по запросу из ЦБУ. Таким образом, ЦБУ управляющим сигналом включает ЛФБ или запрашивает значение параметра, постоянно проверяемого ЛФБ.

На рис. 12.46 приведена упрощенная схема запроса значения параметра p_i, проверяемого *ЛФБ*$_i$.

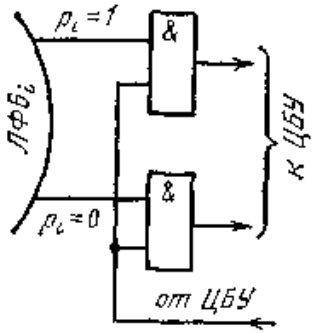


Рис. 12.46

Ко второй группе ФБ — *операторным* ФБ (ОФБ) — относятся все остальные ФБ, осуществляющие управление тем или иным объектом системы. К операторным ФБ относятся всевозможные средства, обеспечивающие выработку сигналов управления на исполнительные средства образования понятий.

В качестве ОФБ может использоваться простое исполнительное средство, открывающее под воздействием сигнала от ЦБУ путь для передачи информации от одного средства (блока) образования понятия к другому. Об окончании выполнения соответствующей операции от операторных ФБ в ЦБУ может передаваться уведомительная информация.

Как логические, так и операторные ФБ могут иметь местное управление (МУ). В таких ФБ может выполняться некоторая

последовательность операций по управлению образованием понятий независимо от ЦБУ до естественного окончания этой последовательности или до тех пор, пока не придет из ЦБУ сигнал выключения. Функциональные блоки с местным управлением называются *активными ФБ* в отличие от *пассивных,* в которых местное управление отсутствует. Программные ООП с активными ФБ являются частным случаем многоблочных иерархических (децентрализованных) ООП, в которых функции ФБ совмещены с блоком управления.

Использование в качестве активных ФБ микропроцессоров позволяет широко применять в системах управления иерархические ООП, приближая органы управления к блокам образования понятий. При этом во многих случаях оказывается целесообразным строить не только двухуровневые, но и многоуровневые иерархические программные ООП на базе компьютеров.

Наличие микропроцессоров создало хорошие условия для образования и широкого распространения распределенного программного ООП. При этом в случае относительно большого расстояния между блоками образования понятий, а следовательно, и между ФБ создается *сеть связи ФБ,* в которой весь алгоритм функционирования ООП распределен между ФБ, а вместо БУ устанавливаются коммутационные компьютеры (или коммутационные микропроцессорные модули), соединенные между собой *моноканалом* или *разветвленной сетью,* образуя *локальные информационно-вычислительные сети образования понятий* или в случае их использования для управления образованными понятиями *локальные управляющие сети понятиями.*

Активные ФБ часто применяют для преобразования дискретных управляющих сигналов, вырабатываемых ЦБУ, в аналоговые сигналы, необходимые для управления ИМ с аналоговым входом, и наоборот, для преобразования аналоговых сигналов в дискретные сигналы в ЦБУ. Активные ФБ обычно обеспечивают также заданную продолжительность включения ИМ или включения ИМ до тех пор, пока датчик не отметит переход блока образования понятия в заданное состояние.

Многие системы образования понятий характеризуются тем, что ряд операций по образованию понятий выполняется не автоматическими средствами, а человеком-разработчиком, на которого возлагаются функции принятия решений в сложных ситуациях. Тогда говорят, что существует *автоматизированная система управления образованием понятий* (АСУ ООП). В данном случае образуется система АСУ ООП-ПП, структура которой изображена на рис. 12.47.

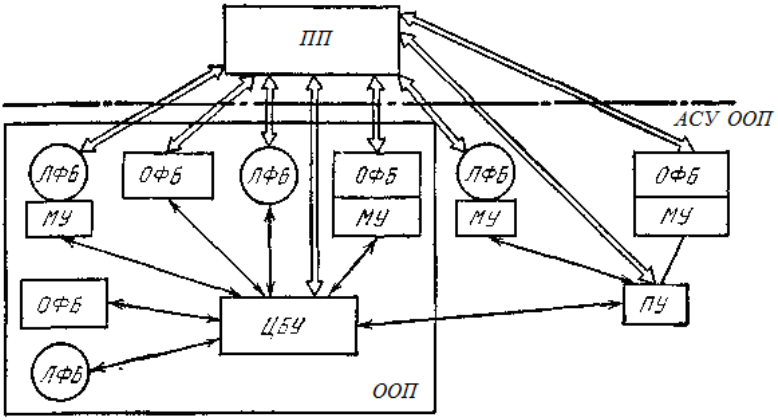


Рис. 12.47

Одинарными линиями указаны пути, по которым передаются сигналы управления внутри АСУ ОПП, а двойными — пути, по которым передаются сигналы взаимодействия (сигналы управления и сигналы о состоянии образовываемого ПП) между АСУ ООП и ПП. Объект управления может быть связан с ЦБУ не только через ФБ, но и непосредственно или через пульт управления (ПУ) оператора.

В ряде случаев активные ФБ, осуществляющие взаимосвязь АСУ ООП с ПП, удобно рассматривать в виде *промежуточных управляющих устройств* (ПУУ), выделенных из ООП (рис. 12.48).

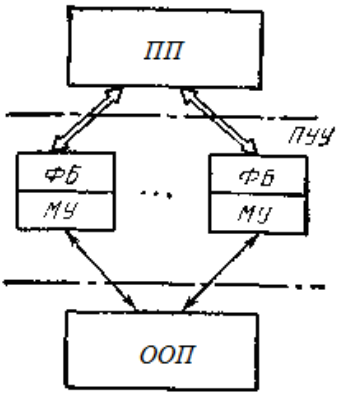


Рис. 12.48

Это особенно удобно в том случае, если ОУ находится на значительном расстоянии от ООП и при этом необходимое число связей между ООП и активными ФБ существенно меньше числа связей

между ФБ и ОУ. В этом случае при наличии в ОУ удаленных друг от друга блоков все ПУУ также целесообразно разделить на соответствующее число блоков, например на два, как показано на рис. 12.49.

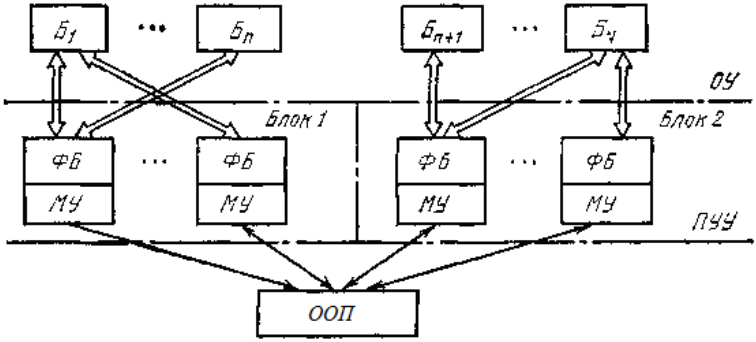


Рис. 12.49

В программном ООП ЦБУ вырабатывает сигнал включения очередного ФБ после того, как закончит работу предыдущий ФБ. В зависимости от способа получения центральным блоком управления сигнала об окончании работы ФБ различают *асинхронный* и *синхронный* режимы работы программного ООП.

При асинхронном режиме сигнал в ЦБУ об окончании работы каждого ФБ поступает непосредственло от ФБ, при этом от ОФБ поступает один сигнал об окончании его работы, а от ЛФБ — два сигнала (сигналы об истинном и ложном значениях проверяемого условия).

Синхронный режим работы программного ООП характеризуется тем, что в ЦБУ сигнал об окончании работы ОФБ и один из сигналов от ЛФБ (рис. 12.50) имитируются *тактовым генератором* (ТГ).

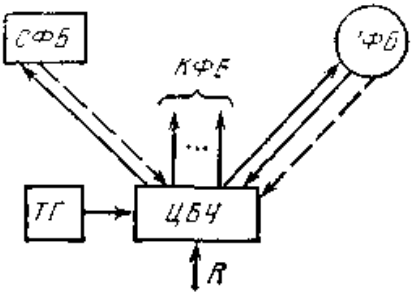


Рис. 12.50

При этом частота тактового генератора выбирается такой, что любой ФБ успевает закончить свою работу до появления очередного импульса от ТГ. В связи с тем что продолжительность работы одного ФБ может существенно отличаться от продолжительности другого, при синхронном режиме быстродействие ООП определяется наиболее медленно действующим ФБ. Для повышения быстродействия ООП частота ТГ может выбираться па основании продолжительности работы большинства ФБ. На время выполнения ФБ, продолжительность работы которого превышает продолжительность такта, тактовый генератор отключается от ЦБУ до получения сигнала от ФБ об окончании его работы или на предусмотренное заранее время.

Максимальное быстродействие при выбранных элементах дает асинхронный режим работы в сочетании с оптимальной программой. Однако при асинхронном .режиме требуется получение ЦБУ сигналов от всех ФБ об окончании их работы, что может привести к усложнению ЦБУ и ФБ.

Как при асинхронном, так и при синхронном режиме использование ЦБУ остается достаточно низким, так как при управлении *медленно действующим функциональным блоком* (МФБ) ЦБУ не может приступить к управлению следующим ФБ до окончания работы МФБ. Устранить этот недостаток позволяет *многопрограммное управление,* при котором может одновременно выполняться несколько процессов. Различные процессы могут протекать как по одной и той же программе, так и но разным. Многопрограммное управление позволяет значительно повысить использование ЦБУ, а в связи с этим повысить его производительность.

Программный ООП можно рассматривать как ЭВМ, специализированную для решения определенного круга задач образования понятий.

Для хранения промежуточных данных, определяемых теми или иными ФБ, сведений о моменте прерывания выполнения программы и т. п. при многопрограммном управлении в ЦБУ необходимо иметь *оперативную память* (ОП), допускающую как считывание, так и запись информации.

В связи с этим программный ООП может бьпь представлен в виде структурной схемы, изображенной на рис. 12.51.

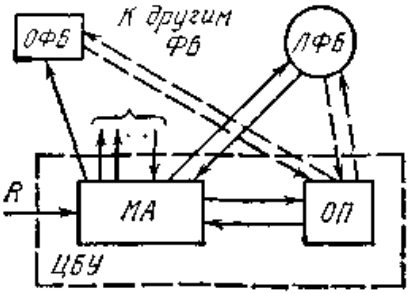


Рис. 12.51

На рисунке сплошными линиями указаны цепи, по которым передается управляющая и уведомительная информация, а пунктирными — цепи передачи вспомогательной информации (например, промежуточных величин) .

В качестве активных ФБ эффективно могут использоваться микропроцессоры. При этом возможности МП позволяют сосредоточить функции ряда ФБ в одном микропроцессорном модуле (МПМ). Тогда структурная схема ООП включает ЦБУ и ряд МПМ (рис. 12.52,*а*). В частном случае функции всех ФБ могут быть реализованы в одном МПМ (рис. 12.52,*б*).

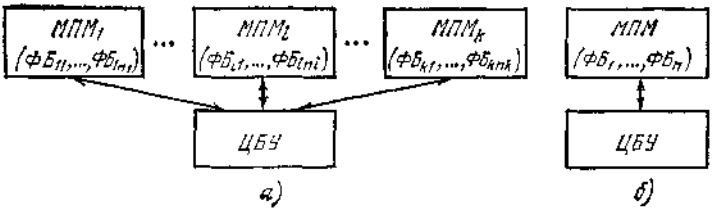


Рис. 12.52

В качестве ЦБУ может также использоваться МПМ. В этом случае вся схема ООП примет децентрализованный вид (рис. 12.53,*а*), в которой в качестве ЦБУ используется $МПМ_0$, или вид распределенной многомикропроцессорной системы, в которой нет ЦБУ и все, например четыре, МПМ (рис. 12.53,*б*) являются равноправными.

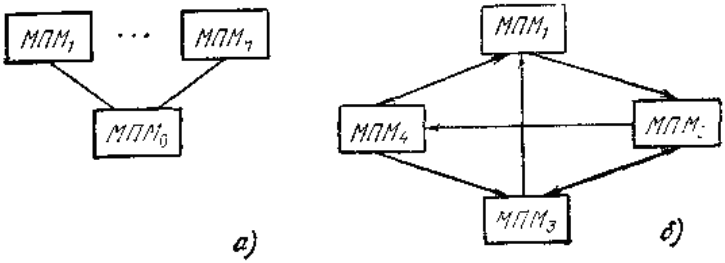


Рис. 12.53

Если блоки ОУ, а следовательно, и МПМ находятся на значительном расстоянии друг от друга, то связь между ними может осуществляться через сеть связи, и тогда образуются сетевая децентрализованная (рис. 12.54,*а*) и сетевая распределенная (рис. 12.54,*б*) многомикропроцессорные системы.

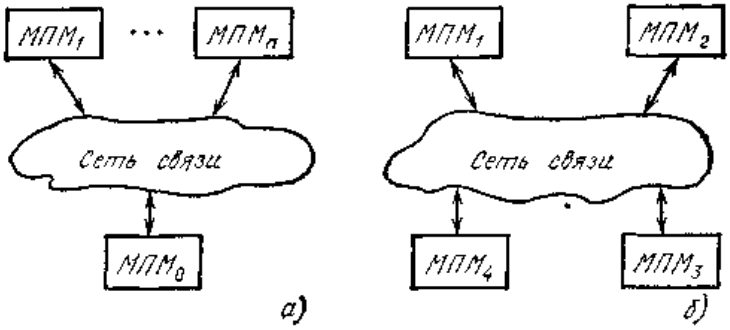


Рис. 12.54

Последнюю будем называть *микропроцессорной сетью* или *сетью микропроцессоров.*

Использование многомикропроцессорных ООП, особенно распределенных, создает значительные возможности по сравнению с многопрограммным управлением по организации параллельной обработки уведомительной информации и реализации в них параллельных процессов образования понятий.

13. Предпосылки создания автоматизированных систем образования фреймов

13.1. Системный характер сложных объектов (истин) и процессов образования фреймов

Процессы атоматизированного образования фреймов и их компонентов относятся к числу сложных. Они характеризуются большим числом элементов (истинами), сложными пространственно-временными связями, зависимостью общих свойств объекта (истины) не только от свойств составляющих его элементов, но и от характера связей между ними.

Более сложными являются процессы и системы автоматизации образования фреймов (САОФ), которые состоят из большого числа взаимосвязанных операций образования фреймов, осуществляющих поиск, анализ, синтез, оценку, оптимизацию и выбор решений на различных стадиях образования фреймов.

Традиционные методы изучения и образования сложных истин (объектов и процессов), при которых основное внимание уделялось качественному и количественному описанию свойств объектов и составляющих их частей, не позволяют строить адекватные действительности модели, отображающие связи объектов с окружающей средой, их функцию и многоуровневую структуру. В то же время указанные характеристики объектов оказывают решающее влияние на вид и структуру алгоритмов процесса образования фреймов.

В связи с этим все большее значение для дальнейшего развития теории и методов образования фреймов приобретает системный подход к процессам образования фреймов. В отличие от традиционных методов образования системный подход исходит из того, что специфика сложных объектов и процессов не исчерпывается свойствами составляющих их элементов, а обусловлена характером связей и отношений между элементами.

Системный подход является методом комплексного изучения сложных объектов и процессов со стороны того, как они устроены, в каких отношениях и связях находятся их части, какова функция частей и объекта в целом, каков характер его взаимодействия с окружающей средой. К понятию «сложный объект» или «система» относятся объекты любой природы, которые можно условно или физически

расчленить на совокупность более простых взаимосвязанных между собой частей. Отношения характеризуют связи между частями и их свойствами, посредством которых части и элементы объединяются в систему. К категории системных относятся только те объекты и процессы, которые состоят из отдельных частей и элементов, но обладают целостным характером функционирования. Системные объекты обладают новыми функциями и свойствами, которых может не быть у составляющих их элементом. Например, собранный из деталей узел характеризуется свойствами и функцией, которых нет у отдельных деталей.

Системные объекты и процессы Q описываются следующими системными характеристиками: связями с окружающей средой H, структурой S, функцией F и набором свойств Z:

$$Q = \{H,S,F,Z\}.$$

Процесс образования фреймов рассматривается как часть обобщенного процесса САОФ, связанная с качественным изменением состояния объектов . В этом определении выделяются дне системные характеристики: целостность процесса и его функция. Процесс образования фреймов характеризуется как относительно обособленная часть обобщенного процесса САОФ, связанная с другими его частями; со стороны функции — как процесс качественного и количественного преобразования объектов из состояния истин в состояние фреймов. Со стороны структуры процесс образования фреймов представляет собой совокупность взаимосвязанных этапов, операций и фрагментов операций. Процессы, обладающие указанными свойствами, могут рассматриваться как системные. Это дает возможность при разработке методов анализа и синтеза процессов образования фреймов опираться на аппарат кибернетики и теории систем.

В системных исследованиях можно выделить три основных направления: структурно-функциональный анализ, структурализм и непосредственно системный подход. Эти направления отличаются тем, что рассматривают различные стороны объекта как системы.

Структурно-функциональный анализ изучает сложноорганизованные объекты с точки зрения выполняемых ими функций по отношению к более сложной системе, и состав которой они входят. Различные свойства объекта синтезируются в целостную картину при помощи совокупности функций. Это позволяет представить объект как организованную систему и перейти к изучению ее структуры и организации. В структурно-функциональном анализе основную нагрузку несет понятие функции, а структура объектов и процессов как бы постулируется.

В структурных исследованиях изучается проблематика целостности, расширяется понятие о связях и их типологии. Принцип целостности нашел конкретное воплощение в выдвижении на первый план понятия структуры как инвариантной характеристики сложного объекта. Понятие структуры связано с осознанием иерархичности строения сложных объектов и процессов. Выдвигается задача выявления и классификации типов связей, формулируются требования комплексного подхода, учитывающего различные виды связей в сложноорганизованных объектах. В связи с этим в структурных исследованиях основной упор делается на изучение структуры, а функциональная сущность ее частей берется как одна из предпосылок.

В системном подходе понятие система является более широким, оно включает в себя в качестве составляющих такие понятия, как структура, функция, состояние, связь, элемент, отношение, управление и др.. Это создает преимущества системного подхода перед традиционными методами исследования. В связи с этим системный подход служит методом комплексного изучения сложных объектов и процессов со стороны того, как они устроены, в каких отношениях и связях находятся их части, какова функция частей и объекта в целом, каков характер взаимодействия с окружающей средой.

Для системного подхода свойственно углубленное внимание к разработке собственного методологического аппарата. Специфической чертой этой методологии является стремление основывать ее на принципе изоморфизма законов в различных областях знаний. Один из основоположников общей теории систем Л. Берталанфи считал, что выявление и анализ законов и соотношений, общих для различных объектов и процессов,— главная ее задача. Отсюда вытекает тезис о междисциплинарном характере системного подхода и о возможности переноса законов и понятий из одной области знаний в другую.

Особое место в системных исследованиях занимает кибернетика. В ней широко используется понятийный аппарат системного подхода, но, несмотря на это, кибернетика обладает своим собственным предметом исследования. Как отмечал М. И. Сетров, кибернетика, являясь наукой об управлении, рассматривает прежде всего одну очень важную сторону функционирования систем — регулятивную, в то время как для системного подхода характерно комплексное изучение объекта, охватывающее различные стороны его строения и функционирования.

К понятию системный объект или процесс относятся объекты и процессы любой природы, которые можно условно или физически расчленить на совокупность более простых взаимосвязанных между собой частей, выступающих как единое целое. Отношения характеризуют связи между частями и их свойствами, посредством которых

части и элементы объединяются в систему. В свою очередь каждая полученная часть может рассматриваться как сложный объект, состоящий из более простых элементов.

В связи с этим к категории системных относятся не все объекты и процессы, а только те, которые состоят из отдельных частей и элементов и обладают целостным характером функционирования. Свойства и функции системных объектов не сводятся непосредственно к сумме свойств и функций составляющих их элементов. Они обладают новыми функциями и свойствами, которых может не быть у отдельных элементов. Например, собранный из отдельных деталей узел представляет собой техническую систему. Он характеризуется новыми свойствами и функциями, которых нет у отдельных деталей. В дальнейшем сложные объекты и процессы образования фреймов будем рассматривать как системы автоматизированного образования фреймов (САОФ).

САОФ определим пятеркой следующих характеристик:

$$\text{САОФ} = \{H, F, S, Z, U\},$$

где H — связи системы с окружающей средой; F — набор выполняемых системой функций; S — структура системы; Z — совокупность функциональных и структурных свойств системы; U — история функционирования и развития системы.

Приведенные характеристики относятся к числу системных и определяют наиболее существенные черты строения и функционирования сложных объектов и процессов. Для того чтобы процессы образования фреймов можно было бы отнести к категории системных, необходимо показать, что они обладают указанными выше системными характеристиками.

Структура процессов образования фреймов в САОФ характеризуется совокупностью моделей и алгоритмов, описывающих информационные, логические и функциональные связи операций образования фреймов, процедур и модулей.

Важной системной характеристикой САОФ является ее история, включающая накопленный и обобщенный опыт процессов образования фреймов, изменения и усовершенствования структуры алгоритмов и программ выбора решений.

Процессы, обладающие совокупностью указанных свойств, могут рассматриваться как системные. Это дает возможность при разработке методов анализа и синтеза процессов образования фреймов опираться на аппарат кибернетики и теории систем. По сравнению с традиционными методами системный подход позволяет на единой методологической основе разрабатывать методы автоматизации образования фреймов, строить модели и формализованные языки для

их описания и осуществлять проектирование САОФ как сложной человеко-машинной системы. Рассмотрим более подробно основные системные характеристики объектов и процессов в САОФ.

13.2. Связи системных объектов и процессов с окружающей средой и их функция

Функциональная целостность и относительная выделенность САОФ предусматривают наличие связей между системой и окружающей средой. К последней будем относить совокупность объектов и систем, которые оказыиают влияние на рассматриваемую систему или записях от характера ее функционирования. САОФ связаны с каждой из систем окружающей среды, как правило, не одним, а несколькими видами связей и отношений. Так, для суперфрейма технологического процесса характерны материальные, энергетические и информационные связи с системами окружающей среды.

Фрейм узла, или сборочной единицы машины, связан с другими фреймами узлов, образующими окружающую среду, конструктивными, кинематическими и размерными связями. Поэтому для правильного понимания взаимодействия системных объектов и процессов с окружающей средой необходим синтетический охват и учет влияния различных видов связи.

В кибернетическом плане суперфрейм технологического процесса изготовления деталей в системе оперативного управления производственным участком представляет собой объект управления (рис. 13.1). На входы технологического процесса изготовления деталей поступают заготовки и управляющая информация. Одна часть этой информации включает плановые задания, определяющие календарные сроки запуска и выпуска деталей, а вторая — технологическую документацию, содержащую алгоритм и программы управления процессом изготовления деталей на различных операциях.

К выходам системы относятся готовые детали и информация о фактическом времени их изготовления и технологических отклонениях. Эта информация поступает в систему оперативного управления производством и в службы технологической подготовки производства. Таким образом, окружающей средой для технологических процессов изготовления деталей будут заготовительные и сборочные цехи, службы технологической подготовки и оперативного управления производством.

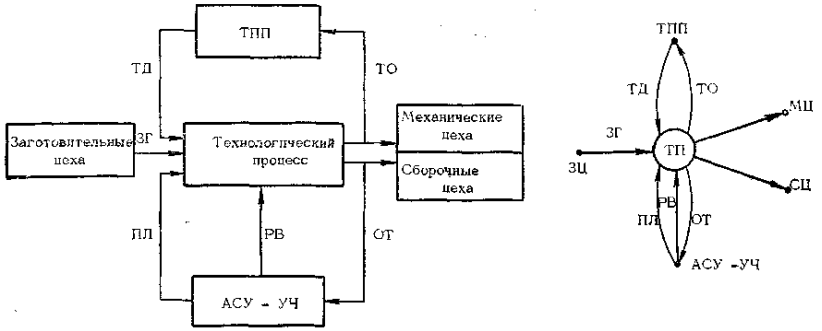


Рис. 13.1. Схема и граф связей технологического процесса с окружающей средой: *ТПП*— технологическая подготовка производства; *ЗЦ, МЦ, СЦ* — соответственно заготовительные, механические и сборочные цеха; *АСУ-УЧ*— автоматизированная система управления участком; *ПЛ, ОТ, РВ* — соответственно план, отклонения от выполнения плана, регулирующие воздействия; *ТД, ТО* — технологическая документация и отклонения

Математической моделью, отражающей множественный характер связей САОФ с каждой из систем окружающей среды, служит мультиграф *H (Q, U).* Множеству его вершин соответствуют рассматриваемая система Q_p и взаимодействующие с ней системы окружающей среды Q_i, а множеству дуг *U* — связи и отношения между Q_p и Q_i. В мультиграфе *H(Q, U)* каждую пару вершин Q_p и Q_{ik} связывает не одна, а несколько дуг в соответствии с числом связей между ними. На рис. 13.1 показан фрейм как модель связей технологического процесса с различными службами и подразделениями предприятия.

Фрейм «Вилка переключения» представляет связь детали вилки переключения (рис. 13.2) с другими деталями узла, которая осуществляется посредством комплекса поверхностей

$$H= \{(П_{11}, П_{21}), (П_{12}, П_{51}), (П_{13}, П_{22}), (П_{14}, П_{31})\},$$

определяющих вид соединений, взаимное положение и степени свободы как расматриваемой детали, так и других деталей, к ней присоединяемых.

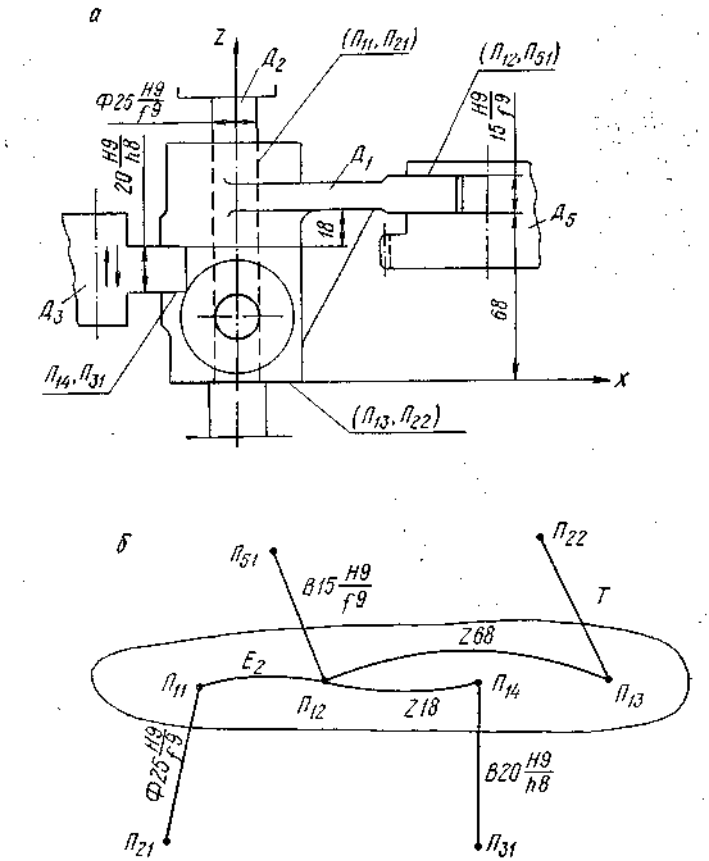


Рис. 13.2. Модель связей детали $Д_1$ с другими деталями узла: *а*—схема механизма переключения; *б* — граф связей детали $Д_1$ с другими деталями: $П_{11}$—$П_{14}$— поверхности соединений детали $Д_1$, $П_{21}$, $П_{31}$ $П_{41}$, $П_{51}$ — поверхности соединений других деталей ($Д_2$, $Д_3$, $Д_4$)

Моделью конструктивных и размерных связей указанных деталей служит граф Н(П, *U*). Его вершинами служат поверхности сопряжений взаимодействующих деталей, а дугами — вид сопряжения и его размеры.

В конструкциях машин и приборов применяются разнообразные типы неподвижных и подвижных соединений: инструментальные конусы, муфты, направляющие, шлицевые, шпоночные и другие соединения. В связи с этим для автоматизации процессов образования фреймов ак-

туальной становится задача систематизации и формализованного описания конструктивных связей и соединений.

Процессы образования фреймов, протекающие в САОФ, также осуществляются на основе обмена информацией с другими системами. На входе исходной информацией для образования фреймов служат чертежи, спецификации изделий и плановые задания, поступающие из конструкторских служб в АСУ технологической подготовки производства. На выходе суперфреймы, представляющие разработанные технологии, передается в АСУП и производственные подразделения предприятия.

Среди различных типов связей системных объектов с окружающей средой особенно выделяются входные и выходные воздействия. Они обеспечивают активное взаимодействие объекта с окружающей средой и во многом определяют его функцию, например преобразование заготовок в готовые детали, передачу движения от двигателя к шпинделям, переработку исходной информации в выходную в САОФ. Другие конструктивные и размерные связи обеспечивают правильное взаимное расположенно и необходимые степени свободы взаимодействующих узлов, т. е. направлены на выполнение требуемой функции и заданных параметров.

В широком смысле функцию определяют как способность системы к целесообразной деятельности в рамках более сложной системы, в состав которой она входит. Целесообразная деятельность характеризуется совокупностью таких реакций системы на изменения, происходящие во внешней среде и внутри системы, которые делают функционирование направленным на достижение поставленной цели; в САОФ, например, в станках с адаптивным управлением эти изменения фиксируются специальными датчиками, передающими информацию о состоянии внешней и внутренней среды в управляющие устройства системы. Последние вырабатывают целесообразные реакции на возникающие в среде изменения, т. е. определяют режимы обработки, которые обеспечивают наибольшую производительность или минимальную себестоимость при достижении требуемой точности обработки детали.

В отличие от живых организмов функция технических систем жестко предопределена ее конструкцией на стадии проектирования или может быть запрограммирована как, например, для станков с программным управлением. Детали машин выполняют различные функции: крепления, фиксации, направления движения, разделения двух сред, объединения деталей в один узел (корпусные детали) и др. Функция каждой детали направлена на выполнение общей функции узла, в ко-

торый она входит. Точно так же функции отдельных узлов направлены на обеспечение функции машины в целом.

Таким образом, функция САОФ Q_T или отдельного ее элемента — это такое их отношение с другими элементами $Q_T\Psi\{Q_1, Q_2, \ldots, Q_n\}$, которое определяет взаимозависимость части и целого, делает функционирование элемента направленным и целесообразным.

Для объектов-преобразователей, связанных с другими объектами посредством входов и выходов, функция определяется преобразованием φ множества входов $X=\{X_i\}$ в множество выходов $Y=\{Y_j\}$ и описывается отображением

$$\varphi:\{ X_i \}\rightarrow \{Y_j \}.$$

Системное определение функции отражает не только количественную, но и качественную сторону зависимости части и целого.

Функция технологического процесса заключается в преобразовании $\varphi_{тп}$ детали из исходного состояния заготовки $C_о$ в конечное $C_к$, определяемое чертежом:

$$\varphi_{тп} : C_o\rightarrow C_к$$

Исходное состояние $C_о$ задается набором параметров, характеризующих форму и размеры заготовки, материал и его механические свойства. Конечное состояние $C_к$ определяет форму, размеры, точность и физико-механические свойства готовой детали. Для процессов сборки $C_о$ характеризуется множеством деталей и сборочных единиц DT_i, $СБ_j$, входящих в состав изделия, а состояние $C_к$ определяет собранное изделие:

$$\varphi_{СБ}:\{ DT_i, СБ_j \}\rightarrow ИЗ.$$

Преобразование φ является сложным. В соответствии с разделением технологического процесса на операции общая функция F расчленяется на совокупность функций отдельных операций: $F\rightarrow\{ \Phi_2, \Phi_2, \ldots\}$. Функция каждой операции заключается в преобразовании детали из одного промежуточного состояния в другое: φ_i: $C_{i-1}\rightarrow C_i$. Преобразование φ_i соответствует виду технологической операции (токарная, фрезерная и т. д.). Промежуточное состояние C_i, характеризует форму, межоперационные размеры детали и их точность, физико-механические свойства поверхностей, полученные в результате выполнения i операции. Состояние C_{i-1} обозначает указанные свойства детали-заготовки, поступающей на операцию.

Для реализации операционной функции Φ требуется выполнить определенное число основных и вспомогательных переходов. В соответствии с заданной структурой операции ее функция расчленяется на совокупность функций отдельных переходов: $\Phi\rightarrow\{ f_1, f_2, \ldots, f_l\}$. Функция основного перехода состоит в преобразовании ψ_j простой или сложной обрабатываемой

поверхности из состояния σ_{j-1} в состояние σ_j, т. е. ψ_j: $\sigma_{j-1} \rightarrow \sigma_j$. Здесь ψ_j — вид перехода (точить, шлифовать и т. д.), а σ_{j-1} , σ_j — состояние обрабатываемой поверхности до и после выполнения перехода.

Таким образом, каждому структурному элементу технологического процесса соответствует своя функция. Расчленение его на операции, переходы, приемы, движения и команды управления станком с ЧПУ приводит к расчленению общей функции процесса на отдельные подфункции.

Механизмы и узлы машин выполняют разнообразные функции: передачи и преобразования движений, направления, опоры, фиксации, зажима и т. д. Сложные объекты, как правило, выполняют не одну, а несколько функций. Например, функция шпиндельного блока заключается в передаче вращения от вала электродвигателя $n_{дв}$ к нескольким шпинделям $n_1, n_2, \ldots, n_k$:

$$\gamma_ш : n_{дв} \rightarrow (n_1, n_2, \ldots, n_k).$$

Функция шпинделя заключается в фиксации $f_ф$, закреплении $f_з$ и сообщении вращения инструменту $f_в$: $F_ш = \{ f_ф, f_з, f_в \}$.

Функция процесса образования фреймов, реализуемого в САОФ системой алгоритмов и программ, характеризуется преобразованием сведений об обрабатываемой детали *DT,* программе выпуска *N* и производственной системе предприятия *ПС* в информационные модели технологических процессов *ТП,* наиболее рациональных для данных производственных условий:

$$v : \{(DT, N)_i, ПС\} \rightarrow \{ТП_i\}.$$

Преобразование *v* является сложным и расчленяется на ряд операций формирования фреймов, таких, как поиск аналогов, синтез решений, имитационное моделирование, оптимизация и др. Описание функций этих операций будет рассмотрено ниже.

Для реализации одной и той же функции могут быть созданы системы САОФ с различной структурой, обладающие разными технико-экономическими характеристиками. Например, обработку детали можно производить с помощью различных по структуре технологических процессов, а заданную функцию узла выполнять различными по конструкции механизмами. Это обстоятельство приводит к многовариантности задач синтеза сложных объектных фреймов и процессных фреймов на основе заданной функции.

13.3. Структура САОФ и процессов образования фреймов

Качественная определенность САОФ обусловлена их структурой, под которой понимается совокупность устойчивых отношений между частями целостного объекта или процессов. Относительная выделенность частей системы и их взаимосвязь — это две противоположности. В связи с этим структуру необходимо рассматривать как единство противоположных сторон: расчлененности и целостности.

Расчлененность отражает одну из общих сторон структуры и характеризуется качественной спецификой частей системы и их количеством. Для каждой системы существует несколько способов расчленения на подсистемы и элементы. Так, процесс образования фреймов можно по-разному расчленить на операции образования фреймов. Изменение качественного состава и количества операций образования фреймов при переходе от одного способа расчленения к другому вызывает изменение структуры САОФ. Это справедливо и для сложных объектов. Металлорежущий станок по функциональному признаку расчленяется на несколько узлов: шпиндельный блок, коробку подач, суппорт, станину и др. При другом способе тот же станок разделяется на механические, электрические, гидравлические и другие узлы.

Таким образом, способ расчленения R системы Q характеризуется множеством ее компонент $R_Q=\{q_1, q_2, ..., q_k\}$ и их качественной спецификой, описываемой набором параметров $Z_q= \{Z_1, Z_2, ...\}$.

Для процессов образования фреймов также может быть не один, а несколько способов расчленения задачи образования фреймов на компоненты различной сложности. Так, при одном способе расчленения в основу принимается вид операций образования фреймов независимо от функционального характера решаемых задач. При этом процесс образования фреймов разделяется на операции образования фреймов: синтез решений, поиск фреймов-аналогов, оптимизация, моделирование, анализ, оценка и др. При другом способе расчленения задача образования фреймов делится на подзадачи по функциональному признаку. Например, общая задача образования фрейма « Проектирование операционнаяй технологии» разделяется на определение формы и межоперационных размеров детали-заготовки, структуры операции, выбор элементов системы СПИД и др.

Расчлененность характеризуется своеобразной ступенчатостью, простирающейся в каждой структуре на различную глубину. Применив

к исходной системе определенный способ расчленения r^1_1, получим множество подсистем первого уровня $\{Q^1_1, Q^1_2, \ldots, Q^1_n\}$, на втором уровне для деления каждой подсистемы Q^1_i может использоваться свой признак: для Q^1_1 — r^2_1, для Q^1_k — r^1_k и т. д. Каждый из признаков r^2_i конкретизирует признак r^1_1. В результате последовательного расчленения получим граф $S_J(Q, R)$ структурного состава САОФ (рис. 13.3).

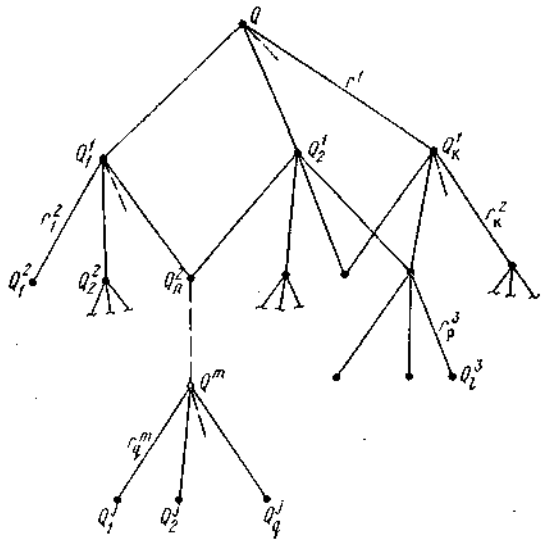


Рис. 13.3. Граф структурного состава САОФ

Вершинами последнего уровня являются базовые структурные элементы объектов и процессов, дальнейшее расчленение которых нецелесообразно с точки зрения характера решаемых задач. Например, при образовании фреймов, в которых представляются станки с ручным управлением технологический процесс расчленяется до уровня переходов, а для станков с цифровым программным управлением процесс обработки дифференцируется до уровня приемов команд управления станком.

Выбор способа расчленения системы зависит от типа решаемых задач. Правильное исходное расчленение объекта или процесса позволит наиболее просто решать задачи анализа и синтеза систем и процессов образовании фреймов, в то время как нерациональное расчленение сильно усложнит эти задачи или сделает невозможным их решение. При каждом способе расчленения получается различное число качественно специфичных частей с разным характером взаимного

расположения. Поэтому для однозначного задания структуры системы необходимо указать способ ее расчленения. Каждому способу соответствует определенный тип взаимосвязей частей системы — своя форма целостности. Целостность — это вторая сторона структуры. Если системность подразумевает наличие связей между компонентами объекта, а также между объектом и окружающей средой, то целостность характеризует большую силу и существенность внутренних связей по сравнению с внешними связями системы с окружающей средой. Именно это обстоятельство создает качественную определенность и выделенность объекта как системы.

Виды и количество связей между частями системы во многом зависят от их природы. Каковы части, их относительное взаимное расположение, таков характер связей и взаимодействия между ними. В сложных объектах, например в механических узлах, существуют кинематические, конструктивные и размерные связи между его частями и соответствующие им три вида структур: S_q, S_k, S_p. В процессе образования фреймов можно выделить функциональные, временные и пространственные связи между различными его структурными элементами. Каждому виду связей соответствует своя структура. Поэтому можно говорить о функциональной, временной и пространственной структурах процесса образования фреймов и его частей $S_ф$, S_B, $S_п$. Таким образом, структура выступает как определенный закон взаимосвязи частей сложных объектов и процессов, как инвариантный аспект системы, в то время как система представляет собой качественно определенную целостность, характеризующуюся множеством взаимосвязанных структур.

Для построения моделей структур фреймов сложных объектов, технологических процессов необходимо выбрать математический аппарат, наиболее просто и адекватно отображающий способы расчленения и различные виды взаимосвязей между их частями и элементами. Наиболее удобной, на наш взгляд, является теория графов и отношений. Рассмотрим модели структур процесса образования фреймов.

Функциональные связи операций процесса образования фреймов характеризуются состоянием объекта, поступающего с предыдущей операции образования фреймов на последующую, и описываются отношением $C_{i-1}\varphi_i C_i$. Совокупность взаимосвязанных отношений, у которых правый элемент C_i одного отношения . является левым элементом другого $C_i\varphi_{i+1}C_{i+1}$, образует граф функциональной структуры процесса образования фреймов $S(\Psi, C)$. Вершинам графа соответствуют состояния объекта C_i, а дугам — обозначения операций

$A = \{\varphi_1, \varphi_2, ..., \varphi_k\}$, посредством которых объект преобразуется из одного состояния в другое.

Функциональная структура определяет частично упорядоченную последовательность преобразований объекта из начального состояния в конечное. Например, формула

$$S_{\Phi}(\Psi, C) = C_0\varphi_1 C_1 \begin{cases} \varphi_2 C_2 \\ \varphi_3 C_3 \varphi_4 C_4 \begin{cases} \varphi_5 C_5 \\ \varphi_6 C_6 \\ \varphi_7 C_7 \end{cases} \end{cases} \varphi_8 C_8$$

обозначает следующий порядок преобразований. Вначале объект из состояния C_o посредством операции φ_1 преобразуется в состояние C_1. Далее возможно выполнение операций φ_2 или φ_3. С точки зрения обеспечения точностных параметров объекта порядок их выполнения безразличен. Такое положение встречается при обработке второстепенных поверхностей. Затем посредством операции φ_4 объект преобразуется в новое промежуточное состояние C_4, и так до тех пор, пока не будет получено требуемое конечное состояние объекта C_8. При этом порядок выполнения операций φ_5, φ_6, φ_7 друг относительно друга строго не регламентируется.

Окончательный порядок выполнения операций задается временной структурой процесса. Временную упорядоченность элементов процесса зададим тремя типами отношений: $a_i \rho a_{i+1}$ или $a_i \cdot a_{i+1}$, $a_i \omega a_{i+1}$, $a_i \tau a_{i+1}$.

Отношение $a_i \cdot a_{i+1}$ обозначает последовательное выполнение a_i и a_{i+1}, отношение $a_i \omega a_{i+1}$ — одновременное, а отношение $a_i \tau a_{i+1}$— сдвиг во времени на величину x начала выполнения a_{i+1} относительно a_i. Первое и третье отношения бинарные, а второе может быть как бинарным, так и n-арным.

Совокупность указанных отношений образует граф или гиперграф $S_B(A, \Omega)$ временной структуры процесса образования фреймов или различных его компонент. Множество вершин графа соответствует операциям, , приемам, а множество дуг — отношениям, отражающим соответственно последовательное, одновременное или со сдвигом фаз выполнение операций. Например, один из вариантов временной структуры процесса обработки детали, соответствующий приведенной выше формуле, описывается следующим выражением:

$$S_B = \varphi_1 \cdot (\varphi_2 \omega \varphi_3) \cdot \varphi_4 (\varphi_5 \omega \varphi_6) \tau_1 \varphi_7 \omega \varphi_8.$$

Из формулы следует, что операции φ_2 и φ_3 выполняются одновременно, φ_7 сдвинута по фазе на τ_1 относительно $(\varphi_5 \omega \varphi_6)$, а φ_8 выполняется одновременно с ней.

Пространственная структура САОФ описывается графом $S_П(B, U)$, множество вершин B которого соответствует элементам системы, а

множество дуг *U* — отношениям, определяющим взаимное расположение элементов в пространстве, их размерные и точностные связи. Так, граф пространственной структуры операции характеризует компоновку элементов станка в рабочей зоне, т. е. взаимное расположение в пространстве шпинделя, стола, приспособления, детали, инструмента, а также размерные связи между ними. Различное взаимное расположение элементов СПИД или частей в сложных объектах оказывает значительное влияние на их структуру. Изменение пространственного расположения силовых голонок относительно обрабатываемой детали и мпогопозпционного стола приводит к разным структурным схемам компоновок агрегатных станков (с центральной колонной, барабанного типа, с силовыми головками, расположенными по периферии стола). Различными по структуре будут и детали, состоящие из одних и тех же элементов, например цилиндров, расположенных относительно друг друга по оси, под прямым углом, параллельно друг другу. В первом случае — это детали класса «тела вращения», во втором — крестовины и угольники, в третьем — коленчатые валы.

Процессы образования фреймов характеризуются иерархической, информационной и логико-функциональной структурой. Первый вид структуры отражает отношения вхождения одних задач в другие, второй — информационные связи между задачами и операциями образования фреймов. С помощью логико-функциональной структуры описывается логика взаимодействия задач и операций образования фреймов, т. е. алгоритм процесса образования фреймов на различных уровнях расчленения.

При многоуровневом расчленении наблюдается прямая связь с периодическим переходом друг в друга расчлененности и целостности. Возьмем, например, образование фрейма для операцию, выполняемой на многопозиционном агрегатном станке. С точки зрения планово-организационной она выступает как целое. Проникая в ее внутреннюю структуру, мы сразу сталкиваемся с отрицанием целостности — расчлененностью на отдельные позиции. При переходе на новый уровень расчленения позиционные переходы выступают как целое и в свою очередь подразделяются на более мелкие части — инструментальные переходы. Эта расчлененность, как показано выше, небезгранична. На следующих уровнях она заканчивается конкретной формой целостности — простым переходом.

Таким образом, расчленение — это отрицание простой целостности системы и переход к новой ее форме на качественно более глубоком уровне расчлененности.

В результате проведения системно-структурного анализа формируется представление о сложной расчлененно-целостной структуре объектов и процессов образования фреймов.

13.4. Функционально-структурные свойства объектов и процессов образования фреймов

Качественные и количественные характеристики системных объектов и процессов образования фреймов задаются совокупностью функционально-структурных свойств. Они характеризуют те стороны системы, которые связаны с ее функционированием и структурой. На различных уровнях расчленения сложных объектов и процессов образования фреймов указанные свойства описываются своим множеством параметров

$$Z^k = \{Z_1, Z_2, \bullet\bullet\bullet, Z_i\text{-}, \ldots, Z_n\},$$

где Z_i — параметр, описывающий определенное функциональное свойство объекта k уровня расчленения.

Так, отдельная операция определяется типоразмером станка *СТ,* установочно-зажимного приспособления *ПР,* типом наладки *НЛ,* составом и количеством переходов *m,* операционным и подготовительно-заключительным временем $T_{оп}$, $T_{пз}$, себестоимостью операции $C_{оп}$ и разрядом работы *Р:*

$$Z_{оп} = \{СТ, ПР, НЛ, m, T_{оп}, T_{пз}, C_{оп}, P).$$

Совокупность функционально-структурных свойств образует техническую характеристику САОФ.

Функционально-структурные свойства программных комплексов САОФ задаются набором параметров и выражений, определяющих объем программы, машинное время решения задачи, точность вычисления искомых параметров. Таким образом, с помощью множества Z описываются характеристики вершин графа иерархической структуры сложного объекта или процесса на всех уровнях расчленения.

Функциональные свойства САОФ, как правило, не сводятся к сумме свойств составляющих их частей. Благодаря структурным связям частей в системе последние обладают принципиально новыми системными свойствами, которых может и не быть у отдельных ее частей. Так, функциональные свойства изделия иные, чем свойства составляющих его узлов и деталей.

Системность и синтетический характер функциональных свойств проявляются в том, что последние зависят не только от свойств

составляющих систему частей, но и от структуры системы в целом, т. е. от характера взаимосвязей ее частей, описанных графом S(B, *U)*. Например, точностные параметры состояния детали после выполнения операции в соответствии с принципом совмещения конструкторских баз с технологическими определяются не только точностью выполнения размеров в отдельных переходах, но и структурой операции, т. е. порядком выполнения переходов.

Характеристики программ образования фреймов (время вычислений, точность решения) также не аддитивны и зависят не только от количества операторов проектной задачи и значений указанных параметров для каждого оператора, но и от характера связей между элементами, т. е. от целостной структуры задачи образования фреймов.

Рассмотренные выше системные характеристики отражают состояние системы в определенный период времени. В процессе эксплуатации система претерпевает определенные изменения. Одни изменения ухудшают значения функциональных свойств (понижается точность, надежность), другие модернизация, адаптация, обучение) приводят к восстановлению работоспособности, совершенствованию и развитию системы. Совокупность длительных, в ряде случаев необратимых изменений системы в процессе эксплуатации составляет ее историю.

Для процессов образования фреймов, история характеризуется упорядоченной совокупностью усовершенствований, осуществляемых за счет внедрения новых, более прогрессивных методов образования фреймов. В результате этих изменений САОФ в процессе эксплуатации проходит ряд состояний: исходное — $V_и$, состояние после определенного периода эксплуатации — V_b, , модернизации — $V_М$ и др. Указанные состояния задаются набором параметров, описывающих достоверность, надежность, а также дополнительные функциональные возможности, появившиеся после усовершенствования или модернизации.

Историю САОФ можно задать в виде последовательности изменений преобразований системы из исходного состояния $V_и$ в текущее.

Знание истории эксплуатации САОФ и текущего его состояния дает возможность составить объективные алгоритмы выбора рациональной модели фрейма в зависимости от требуемых условий.

Еще большее значение история функционирования имеет для САОФ, в которой прошлый опыт образования фреймов часто принимается за основу при выборе решений. В САОФ накопленный опыт образования фреймов в первую очередь включает массивы ранее образованных фреймов, а также массивы типовых и групповых фреймов.

История САОФ связана с накоплением и обобщением опыта образования фреймов, усовершенствованием программ образования

фреймов, обновлением информационных массивов и улучшением их организации. Совокупность программ, информационных массивов и применяемых технических средств характеризует состояние САОФ. История описывается упорядоченной совокупностью преобразований указанных компонент системы и возникающих при этом состояний САОФ :

$$И_{\text{САОФ}} = C_и \eta_1 C_1 \eta_2 C_2, \ldots, \eta_k C_k.$$

Преобразование η_i, включает в себя действия по накоплению и обобщению опыта образования фреймов, обучению и адаптации программ, модернизации и замены элементов технического комплекса системы. Указанные действия производятся периодически обслуживающим персоналом, программами обучения и адаптации. В результате реализации указанных мероприятий повышается качество функционирования системы автоматизации образования фреймов, а история САОФ становится существенным признаком ее полезности.

На основе изложенного общая математическая модель САОФ и процессов образования фреймов может быть задана двумя группами соотношений. Первая определяет состав системных характеристик и описание объекта как целого и его компонент на различных уровнях расчленения:

$$Q = \begin{cases} \langle H, F, S_J(S_ф, S_п, \ldots) Z, U \rangle, \\ \{\langle F, (S_ф, S_п, \ldots) Z \rangle_i^1\}, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \{\langle F, (S_ф, S_п, \ldots) Z \rangle_i^k\}, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \{\langle F, Z \rangle_q^m\}. \end{cases}$$

В этой модели выделено три типа объектов: системный объект как целое, объекты промежуточных уровней расчленения и базовые. Указанные объекты отличаются составом системных характеристик, необходимых для их описания. В первой строке модели задаются характеристиками системы в целом, т. е. рассматриваются ее связь с окружающей средой, функция, свойства, структурный состав системы, история функционирования, а также функциональная и другие виды структур. Компонентами этих структур являются объекты, непосредственно образующие систему.

Следующие соотношения описывают функцию, свойства и структуру объектов на промежуточных уровнях расчленения. Для них связи с окружающей средой не задаются, так как они вошли в описание

структуры более сложных объектов предыдущего уровня. История функционирования, как правило, относится к системе в целом и поэтому при описании отдельных компонент не указывается. Последнее соотношение задает функцию и функционально-структурные свойства базовых объектов и процессов на последнем уровне расчленения, когда знание структуры объектов для решения поставленных задач не требуется.

Соотношения второго типа описывают модели каждой из системных характеристик для объектов различных уровней расчленения:

$$H = (Q,\ U),$$

$$F = \begin{cases} Q\omega\{Q_1,\ Q_2,\ \ldots,\ Q_n\}, \\ \varphi : \{X_i\} \to \{Y_j\}, \end{cases}$$

$$S = \{S_1(B,\ U),\ S_2(B,\ L),\ \ldots\},$$

$$Z = \{Z_1,\ Z_2,\ \ldots,\ Z_k\},$$

$$U = V_1\eta_1 V_2\eta_2,\ \ldots,\ \eta_k V_k.$$

Системные характеристики объекта или процесса как целого характеризуются наибольшей степенью интеграции. Чем глубже уровень расчленения, тем детальнее представлены структура, функция и свойства компонент системы. На последнем уровне расчленения достигается наибольшая степень детализации системных характеристик.

Для решения конкретных задач в системе образования фреймов не всегда нужны все системные характеристики. Поэтому на основе общей системной модели могут быть построены различные частные модели, содержащие описания только некоторых характеристик с требуемой степенью детализации.

13.5. Методические основы построения теории автоматизированного образования фреймов

В связи с автоматизацией образования фреймов возникла острая необходимость осмысливания накопленного опыта и построения теории автоматизированного образования фреймов, раскрывающей основные закономерности процесса образования фреймов и служащей научным фундаментом создаваемых САОФ.

Теорию автоматизированного образования фреймов будем строить не столько «снизу» за счет индуктивного обобщения полученных наукой и практикой результатов, сколько «сверху» по отношению к ним, т. е. путем дедуктивного построения на основе сформулированных фунда-

ментальных исходных посылок и принципов целой системы более конкретных утверждений, раскрывающих структуру и содержание х операций синтеза, оптимизации и выбора решений автоматизированного образования фреймов.

При таком построении теории автоматизированного образования фреймов исходные посылки и принципы должны отражать не простые истины, доказательство которых не требуется, а фундаментальные закономерности процессов автоматизированного образования фреймов и принципы их системной организации. Они получены на основе обобщения опытных данных, систематизации результатов теоретических и практических работ в системных исследованиях. Так, например, на основе сформулированного принципа технологической совместимости получены утверждения и соответствующие им алгоритмы выбора допустимых вариантов методов обработки, технологических баз, типоразмеров станков, установочно-зажимных приспособлений и инструментов, которые должны использоваться при автоматизированном образовании фреймов в области технологии машиностроения.

На основе исходных посылок дедуктивно, т. е. сверху вниз, развертывается целая система более конкретных утверждений и следствий, являющихся базой построения алгоритмов и программ автоматизированного образования фреймов. Построенная таким образом система утверждений имеет иерархическое строение. Верхний, нулевой, уровень системы образует множество исходных посылок $ИП = \{ ИП_{1k}, ИП_2, ..., ИП_k\}$. На их основе формируется и доказывается ряд промежуточных утверждений первого уровня

$$УТ^1 = \{ УТ_1, УТ_2, ..., УТ_r\}.$$

В дальнейшем при использовании исходных посылок и утверждений первого уровня $УТ^1$ строятся более конкретные утверждения второго уровня. Этот процесс продолжается до тех пор, пока не будут получены утверждения, определяющие функцию, структуру и параметры элементов объекта или процесса автоматизированного образования фреймов. В формальном виде дедуктивная система построения целостной модели процесса автоматизированного образования фреймов описывается совокупностью соотношений

$$ИП^0 = \{ИП_1, ИП_2, \ldots, ИП_k\},$$

$$Г^1 : ИП^0 \rightarrow УТ^1,$$

$$Г^2 : \{ИП^0, УТ^1\} \rightarrow УТ^2,$$

$$\cdots\cdots\cdots\cdots$$

$$Г^n : \{УТ^q, \ldots, УТ^{n-1}\} \rightarrow УТ^n,$$

где $Г^1, Г^2, ..., Г^n$ — процедуры формулирования и доказательства более конкретных утверждений следующего i уровня на основе исходных посылок $ИП^0$ и полученных ранее более общих утверждений i—1 уровня. Переход от совокупности исходных посылок к утверждениям и от одних утверждений к другим, более конкретным, осуществляется как формально, так и путем содержательного синтеза знаний, полученных в системных исследованиях. Утверждения последнего уровня представляют собой математические модели построения допустимых техническими ограничениями вариантов решений и служат основой для построения алгоритмов и программ автоматизированного образования фреймов:

$$E : УТ^n \rightarrow \{АЛ_i\},$$

$$TP : \{АЛ_i\} \rightarrow \{ПР_i\},$$

где E — совокупность процедур построения на основе утверждений $УТ^n$ комплекса алгоритмов $АЛ$ решения задач автоматизированного образования фреймов; TP — совокупность процедур трансляции алгоритмов в программы $ПР$.

В результате модель теории автоматизированного образования фреймов может быть представлена в виде графа $W(П, U)$. Вершинам верхнего уровня (рис. 13.4) соответствуют исходные посылки и принципы, а вершинам следующих уронней — утверждения и следствия, которые получены на основе исходных посылок и утверждений предыдущих уровней.

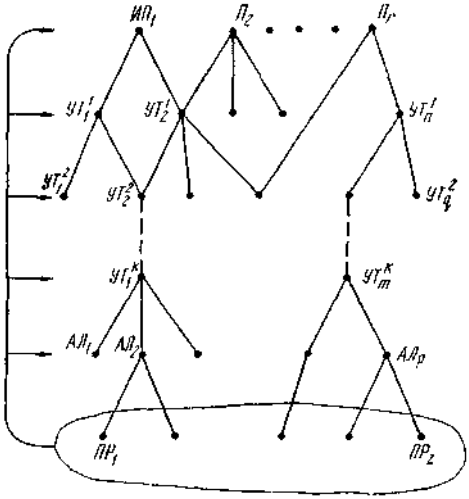


Рис. 13.4. Структура теории автоматизированного образования фреймов: $ИП_1$, $ИП_2$—исходные посылки и принципы; $УТ^k$ — утверждения k уровня; *АЛ, ПР* — алгоритмы и программы автоматизированного образования фреймов

На последних уровнях на основе одного утверждения может быть построено несколько различных реализующих его алгоритмов и программ, т. е. обеспечивается функциональная избыточность программного комплекса САОФ.

Полученная таким образом система исходных посылок и утверждений должна отражать основные закономерности процессов автоматизированного образования фреймов в выбранной проблемной области и, как правило, не зависеть от условий конкретной области, т. е. являться инвариантной основой САОФ, в то время как алгоритмические и программные уровни подвергаются настройке при переходе от одной предметной области к другой и составляют изменяемую, адаптируемую часть САОФ.

При хорошо разработанной теории автоматизированного образования фреймов утверждения последнего уровня определяют не только единственный вариант оптимального решения задачи в конкретных условиях, но и наиболее рациональный (с минимальным числом переборов) путь, к нему ведущий. В тех случаях, когда теория некоторых задач недостаточно разработана, утверждения последнего уровня определяют уже не один, а множество допустимых вариантов решений. Выбор оптимального варианта может осуществляться с помощью математических методов оптимизации, если задача формализуема, или построения диалоговых процедур и обращения ЭВМ к разработчику с просьбой решить задачу, если последняя не поддается формализации по тем или иным причинам.

Таким образом, построенная теория автоматизированного образования фреймов определяет роль и место математических методов оптимизации и диалоговых процедур выбора решений. Характерной особенностью рассмотренной модели теории автоматизированного образования фреймов является ее целостность, обусловленная взаимосвязанностью исходных посылок и утверждений различных уровней. В результате практической проверке подвергается вся система исходных посылок и утверждений как единое целое, а не только отдельные ее части. Практическая проверка теории производится по результатам автоматизированного образования фреймов, которые выдает построенный на основе этой теории программный комплекс САОФ.

Рассмотрим некоторые принципы и утверждения системно-структурного анализа сложных объектов и технологических процессов.

Принцип совместимости. Совокупность объектов может быть объединена в систему, если они обладают свойством совместимости по наиболее существенным видам связей и отношений, т. е. такой общностью по выполняемым функциям структурным и функциональным свойствам, благодаря которым обеспечивается их совместное функционирование как единого целого в соответствии с заданными техническими требованиями. Так, фреза и шпиндель станка совместимы, если форма и размеры хвостовика фрезы соответствуют форме и размерам инструментального конуса шпинделя. Операции технологического процесса совместимы, если состояние обрабатываемой детали на выходе одной операции будет исходным для других. Это значит, что некоторые поверхности, обработанные в предшествующей операции, могут быть приняты в качестве базовых на последующих операциях, а форма и размеры обрабатываемых поверхностей одних операций таковы, что обеспечивают необходимые припуски и допуски для обработки детали на следующих операциях.

В реальных системах и процессах входы и выходы одних объектов могут отличаться конструктивным исполнением, расположением в пространстве, величинами размеров и другими свойствами. Все это приводит к тому, что невозможно обеспечить непосредственную совместимость подсистем или их элементов. Сформулируем утверждение, определяющее условия объединения элементов в фреймовую систему при отсутствии их непосредственной совместимости.

Утверждение 1. Объединение в фреймовую систему элементов, несовместимых по одному или нескольким видам связей, достигается путем введения специальных звеньев-посредников, выполняющих функции совместимости по несогласованным видам связей между взаимодействующими элементами. Звенья-посредники в устройствах и в процессах встречаются регулярно. Так, например, инструментальная оправка является звеном-посредником между фрезой и шпинделем станка в связи с тем, что посадочные размеры выбранной фрезы не совпадают с аналогичными размерами шпинделя. Операция подготовки технологических баз, например фрезерно-центровальная, служит операцией-посредником между заготовительными и операциями механической обработки вала. Более подробно вопросы совместимости фреймовых систем и функции звеньев-посредников будут рассмотрены в следующих разделах.

Совместимость взаимодействующих фреймовых систем может осуществляться различными способами, каждый из которых характеризуется различной величиной затрат на его реализацию.

Утверждение 2. Оптимальным среди множества возможных $B = \{b_{iг}\}$ будет такой способ совместимости фреймовой системы с окружающей средой или структурных элементов между собой, который обеспечивает заданные технические требования на взаимосвязь и взаимодействие указанных объектов $\delta_д \lesseqgtr \delta_{iдоп}$ при минимальных суммарных затратах на совместимость по всем видам связей e_j:

$$b^* = \min_{b^* \in B} \sum_{j=1}^{n} e_j,$$

$$\delta_q \lesseqgtr \delta_{i\ доп}.$$

Так, с точки зрения этого утверждения оптимальным по затратам на технологическую совместимость фреймов будет такой вариант фрейма технологического процесса, который при обеспечении заданной производительности и точности обработки определяет минимальные затраты на специальные приспособления, вспомогательный инструмент, операции подготовки установочных баз и другие элементы, выполняющие функции совместимости.

Введем понятие системной оптимальности, которая в отличие от традиционных представлений связывает системные характеристики объекта с затратами на их осуществление. В связи с этим данное утверждение определяет только одну сторону системной оптимальности — это минимальные затраты на совместимость фреймовой системы с окружающей средой и структурных компонент в составе фреймовой системы, т. е. отражает структурные особенности фреймовой системы.

Вторая, не менее важная сторона системной оптимальности— это минимальные затраты $E_{(F,\ Z)}$ на реализацию заданной функции фреймовой системы F и совокупности заданных параметров Z.

Третья сторона системной оптимальности — это затраты $E_{И}$, связанные с совершенствованием и модернизацией фреймовой системы за период эксплуатации. На основе изложенного сформулируем утверждение об оптимальности фреймовой системы в целом.

Утверждение 3. Оптимальным Q* среди множества допустимых техническими ограничениями $\delta_д \lesseqgtr \delta_{iдоп}$ вариантов образовываемых фреймов объектов или процессов будет такой, который обеспечивает минимальные суммарные затраты на совместимость фреймовой системы с окружающей средой, выполнение заданных функций и затраты на совершенствование и модернизацию за период эксплуатации:

$$Q^* = \min_{Q^* \in Q} (E_н + E_{F,S} + E_и).$$

Системные критерии оптимальности в явном виде дифференцированно учитывают затраты на реализацию системных характеристик фрейма. В связи с этим системные критерии лучше отвечают требованиям построения рациональных информационных моделей фреймов-объектов и фреймов-процессов на стадии их образования.

Принцип совместимости и вытекающие из него утверждения 1 — 3 являются фундаментальными и определяют многие важные закономерности образования фреймов.

14. Многоуровневый итерациооный метод образования фреймов

14.1. Многоуровневая декомпозиция процессов образования фреймов

Большое разнообразие форм и размеровобъектов, средств и методов их создания, приводят к тому, что процессы образования фреймов представляет собой сложную, многовариантную и трудпоформализуемую задачу. Исходными данными для ее решения служат задание функции процесса образования фреймов $\varphi : C_0 \rightarrow C_к$. В описании функции известной является информационная модель фрейма-объекта $C_K = \langle F^q, S^q, Z^q \rangle$, определяющая ее системные характеристики на всех q уровнях условного расчленения.

Ограничениями, определяющими допустимые варианты процесса образования фреймов, выступают применяемые методы описания поверхностей и объектов

$$m_l \in \{m_1, m_2, \ldots, m_{n1}\},$$

состав используемых устройств и их техническая характеристика

$$CT_к \in \{CT_1, CT_2, \ldots, CT_{n2}\},$$

набор универсальных, типовы и групповых компьютерных программ

$$ПР_d \in \{ПР_1, ПР_2, \ldots, ПР_{n3}\},$$

$$ИН_r \in \{ИН_1, ИН_2, \ldots, ИН_{n4}\},$$

множество основной и вспомогательной информации необходимой для образования фреймов

$$MT_p \in \{MT_1,\ MT_2,\ \ldots,\ MT_{n5}\}.$$

Задача образования фреймов состоит в том, чтобы при заданных ограничениях определить системные характеристики процесса образования фреймов и его элементов, обеспечивающие формирование заданных форм, точностных параметров и физико-механических свойств объекта с наименьшей себестоимостью:

$$E_{\text{т}} = E_{\text{м}} + \sum_{i=1}^{n} E_i \to \min,$$

где $E_{\text{м}}$ —затраты, необходимые для образования фреймов; $E_{\text{т}}$ — себестоимость отдельных операций образования фреймов ; n — число операций в процессе образования фреймов.

Это значит, что необходимо найти функциональную структуру процесса образования фреймов в целом $S_{\text{ф}}^{\text{т}} = C_{\text{о}}\text{-}A_1C_1A_2C_2, \ldots, A_nC_n$; функциональную, временную и пространственную структуры и характеристики каждой операции образования фреймов: $S_{\text{ф}}(C, A), S_B(A, \Omega), S_n(B, L), Z$; управляющие программы $S_{\text{у}}(\sigma, \Theta)$.

В общей постановке проектирование процессов образования фреймов относится к числу сложных индетерминированных задач, характерные особенности которых следующие:

1) недостаточность имеющихся к началу проектирования исходных данных и технических ограничений для получения решений требуемой степени детализации, например, операционной технологии образования фреймов или управляющих программ;

2) слабая изученность закономерностей процессов образования фреймов; пока еще не выявлены аналитические и логические зависимости, связывающие системные характеристики описываемых устройств и процессов, со структурой и параметрами процесса образования фреймов;

3) необходимость учета изменений информационной окружающей среды. Эти изменения окружающей среды невозможно с достаточной полностью предвидеть и учесть в алгоритмах на стадии их создания.

Одним из наиболее общих способов преодоления начальной неопределенности задачи образования фреймов является разработанный нами многоуровневый итерационный метод. Сущность метода раскрывается совокупностью принципов и утверждений, определяющих характер и структуру процессов образования фреймов.

Принцип многоуровневой декомпозиции процессов образования фреймов. Проектирование $П_{\text{т}}$ дискретных процессов образования фреймов и сложных объектов расчленяется на несколько взаимосвязанных уровней (стадий) 1,2,…,k, характеризующихся

последовательно возрастающей от уровня к уровню степенью детализации j_n проектных решений:

$$П_т \rightarrow \langle П^1_{j1},\ П^2_{j2},\ \ldots,\ П^k_{jn} \rangle .$$

Этот принцип отражает опыт решения сложных задач образования фреймов, когда в связи с недостаточностью априорной информации решения первого уровня носят наиболее общий, абстрактный характер и на последующих уровнях разворачиваются до требуемой степени детализации. Виды и количество уровней зависят от сложности образовываемых фреймов объектов или процессов, а также от требуемой степени детализации информационных моделей фреймов .

Утверждение 1. Проектирование процессов образования фреймов расчленяется на четыре уровня: принципиальная схема процесса образования фреймов — *ПС,* маршрут образования фреймов — *ТМ,* операционная технология образования фреймов — *ОП* и управляющие программы— *УП.*

Принципиальная схема процесса образования фреймов определяет состав и последовательность этапов $Э_u$ образования фреймов, тип объекта (процесса) C_o на который образовывается фрейм, состояние объекта (процесса) $\bar{C}_i$ после выполнения каждого этапа:

$$ПС = \bar{C}_0 Э_1 \bar{C}_1 Э_2 \bar{C}_2,\ \ldots,\ Э_n \bar{C}_n .$$

Маршрут образования фреймов характеризует состав и последовательность операции образования фреймов A_q в каждом этапе, состояние объекта (процесса) $\hat{C}_q$ после выполнения каждой операции образования фреймов и основные характеристики системы образования фреймов (типы технических средств — *СТ,* схема установки — *СУ,* состав подоперций операции образования фреймов a_j**):**

$$ТМ_{Э_t} = \hat{C}_1 A_1 \hat{C}_2 A_2, \ldots,\ A_m \hat{C}_m,$$

$$A_q = \{СТ,\ СУ,\ \{a_j\}\}.$$

Операционная технология образования фреймов включает в себя определение структуры и характеристик операции, дальнейшую детализацию элементов системы образования фреймов:

$$S_A = \{S_ф(C,\ A),\ S_в(A,\ \Omega),\ S_ц(B,\ L)\},$$

Управляющие программы характеризуются совокупностью команд управления техническими средствами образования фреймов при выполнении каждого операционного перехода $S_j(\sigma, \Theta)$.

Для первого уровня свойственны наибольшая степень абстракции и определение только принципиальных особенностей структуры и

функции процесса образования фреймов. От уровня к уровню степень детализации принятых решений возрастает. На последнем уровне она доводится до полного образования фреймав и команд управления техническими средствами.

Доказательство этого утверждения базируется на принципе декомпозиции системы на компоненты по наиболее слабым информационным связям. В принятой схеме декомпозиции связь менаду задачами внутри уровня более сильная, чем двух соседних уровней между собой.

Утверждение 2. Многоуровневый процесс образования фреймов развивается сверху вниз, т. е. от синтеза общих принципиальных моделей на первом уровне к решениям требуемой степени детализации на следующих уровнях. При этом решения, полученные на предыдущем k—1 уровне, используются в качестве дополнительных исходных данных для образования фреймов на k уровне:

$$\nu^k : \{ИД,\ R_i^{k-1}\} \rightarrow \{R_{ij}^k\}. \qquad (1)$$

Так, например, сведения о принципиальной схеме процесса образования фреймов, полученные на первом уровне, служат для синтеза маршрута образования фреймов на втором уровне. Разработка операционной технологии образования фреймов на третьем уровне производится на основе сведений о технологическом маршруте образования фреймов, а для синтеза управляющих программ используются сведения об операционной технологии. Развернув выражение (1) по уровням образования фреймов, получим

$$П_1 = ИД\nu^1 ПС\nu^2 ТМ\nu^3 ОП\nu^4 УП. \qquad (2)$$

В результате многоуровневой декомпозиции и связанной с ней возможности использования промежуточных результатов образования фреймов в качестве исходных данных на следующем уровне системные характеристики процесса образования фреймов на каждом уровне описываются более простыми моделями и алгоритмами.

Общая системная модель многоуровневого процесса образования фреймов включает две группы соотношений. Первая описывает системные характеристики процесса образования фреймов в целом, а также функцию, структуру, накопленный опыт $H_т$ и свойства проектных процессов на всех четырех уровнях детализации:

$$Q = \begin{cases} \langle H,\ v^1 : ИД \to ПС;\quad S^1(\Theta,\ У);\ И^1_{пс};\ Z^1_{пс} \rangle, \\ \langle v^2 : (ИД,\ ПС) \to ТМ;\quad S^2(\Theta,\ У);\ И^2_{тм};\ Z^2_{тм} \rangle, \\ \langle v^3 : (ИД,\ ТМ) \to ОП;\quad S^3(\Theta,\ У);\ И^3_{оп};\ Z^3_{оп} \rangle, \\ \langle v^4 : (ИД,\ ОП) \to УП;\quad S^4(\Theta,\ У);\ И^4_{уп};\ Z^4_{уп} \rangle. \end{cases}$$

Соотношения второй группы определяют взаимосвязь системных характеристик различных уровней. Общая функция процесса образования фреймов описывается как упорядоченная последовательность преобразования исходных данных и промежуточных результатов, полученных на отдельных уровнях (см. формулу (2)).

Структура процесса образования фреймов представляет собой объединение графов структур, описывающих прямые и обратные связи процессов образования фреймов разных уровней:

$$S(\Theta,\ У) = \bigcup_{k=1}^{k=4} S^k(\Theta,\ У).$$

Информационные массивы, отражающие накопленный опыт образования фреймов, представляют собой теоретико-множественное объединение массивов $И^к$ различных уровней

$$И = И^1_{пс} \cup И^2_{тм} \cup И^3_{оп} \cup И^4_{уп} = \bigcup_{k=1}^{k=4} И^k.$$

Таким образом, в результате многоуровневой декомпозиции задача образования фреймов сводится к определению наиболее рациональных системных характеристик процесса образования фреймов на начальном, промежуточном и заключительном уровнях образования фреймов, обеспечивающих заданные технические условия образования фреймов.

Возможно несколько моделей многоуровневого процесса образования фреймов. Основу первой составляют последовательное от уровня к уровню многовариантное построение допустимых вариантов процесса образования фреймов и отбор наиболее рациональных из них на последнем уровне. Процесс образования фреймов на каждом уровне представляет собой многовариантную процедуру. На основе одного варианта R_i^{k-1} k—1 уровня формируется множество $\{R^k_{iq}\}$ более детальных вариантов k уровня. В результате реализации процессов образования фреймов на всех уровнях образуется дерево допустимых вариантов образованных фреймов, отвечающих заданным техническим ограничениям (рис.14.1).

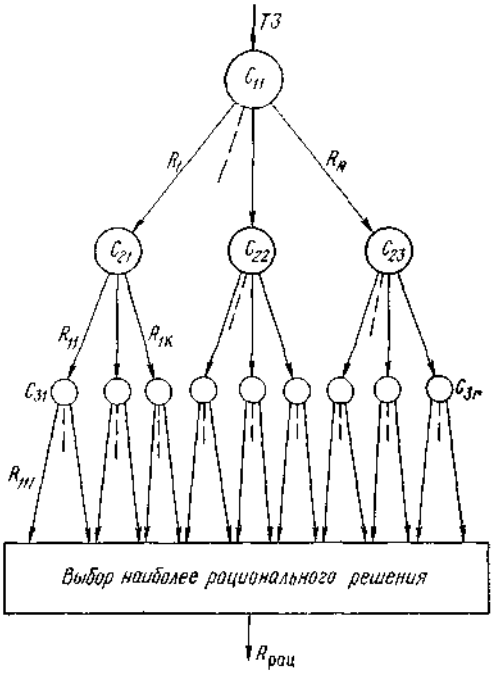


Рис.14.1. Модель многоуровневого процесса образования фреймов с выбором наиболее рационального решения на последнем уровне: *ТЗ* — техническое задание; C_{ij} — операции синтеза решений образования фреймов; R_{qk} —варианты образованных фреймов

Вершинам дерева соответствуют операции синтеза решений образования фреймов, а дугам— полученные варианты этих решений. Дуги дерева последнего уровня характеризуют варианты образованных фреймов заданной степени детализации.

В целом указанную модель процесса образования фреймов можно представить в виде совокупности операций синтеза *v* решений образования фреймов на различных уровнях и операций отбора *W* наиболее рационального варианта по заданному критерию на последнем уровне

$$Q_1 = (ИД v^1 \{R_i^1\} v^2 \{R_{i,j}^2\} v^3 \{R_{i,j,q}^3\} \ldots) W R_q.$$

Рассмотренная модель процесса образования фреймов характеризуется низкой эффективностью. Чтобы выбрать один рациональный вариант, необходимо до конца образовать очень большое число допустимых ограничениями вариантов.

Вторая модель дополняет первую и базируется на применении промежуточного отбора $W_{к\text{-}1}$ на каждом уровне наиболее рациональных вариантов процесса образования фреймов.

Сформулируем утверждение о характере критериев промежуточного отбора вариантов.

Утверждение 3. На всех уровнях, кроме последнего, ввиду недостаточной детализации решений образования фреймов критерии отбора вариантов E_j носят обобщенный, эвристический характер. Они последовательно уточняются при переходе от уровня к уровню, достигая необходимой точности (эффективности) на последнем уровне образования фреймов:

$$E_i^1 < E_j^2 < E_l^3 < E_r^4 .$$

Так, на первом уровне невозможно формировать критерий, позволяющий выбрать один оптимальный вариант принципиальной схемы фрейма. Это связано с тем, что представление об образовываемом фрейме носит сугубо принципиальный характер и на следующих уровнях, как правило, уточняется.

Критерии отбора вариантов маршрута образования фреймов точнее, чем на предыдущем уровне, так как детализация решений образования фреймов значительно выше. Еще более точными будут критерии на уровнях операционной технологии образования фреймов и синтеза управляющих программ.

Эвристические критерии получены на основе опыта решения аналогичных задач и направлены на достижение требуемых результатов.

Из изложенного выше вытекает следующее утверждение о характере промежуточного отбора вариантов при многоуровневом процессе образования фреймов.

Утверждение 4. На начальном и промежуточных уровнях процесса образования фреймов в связи с эвристическим характером критериев из множества синтезированных вариантов $\{ R_i^{k-1} \}$ отбирается не один, а несколько (два-три) наиболее рациональных решений $\{R_j^{k-1}\}$. Окончательный вариант процесса образования фреймов, соответствующий экстремальным значениям точного критерия, определяется только на последнем уровне

$$W^{k-1} : \{R_i^{k-1}\} \rightarrow \{R_j^{k-1}\} \quad (i = 1 \div n, \quad j = 1 \div m, \quad m \ll n),$$

где W^{k-1}— операция отбора вариантов на промежуточных уровнях.

Процесс образования фреймов на k уровне, основанный на отборе неокончательных решений, описывается следующим выражением:

$$Q = \overline{R^{k-1} v \{R_j^k\} W \{R_{jc}^k, R_{jd}^k \ldots\}} P_{(k)}.$$

Эвристические критерии на промежуточных уровнях подобраны так, что малоэффективные варианты, исключенные из рассмотрения, как правило, не дают эффективных вариантов на заключительных уровнях образования фреймов. Вероятность того, что в результате промежуточного отбора будет потеряно самое лучшее решение, тем меньше, чем больше значения критерия качества исключаемого варианта превышают установленный для отбора порог.

На промежуточных уровнях образования фреймов каждый исходный вариант разделяется на несколько более подробных, часть из которых отсеивается блоком оценки. Многоуровневый синтез в сочетании с многоступенчатым отбором и фильтрацией позволяет резко сократить число анализируемых вариантов процесса образования фреймов и значительно повысить эффективность решения задач. На рассматриваемом уровне наряду с детализацией решений образования фреймов производятся корректировка и уточнение решений, принятых на предыдущих уровнях. Вследствие этого между уровнями процесса образования фреймов возникают обратные связи. Например, (рис. 14.2), выбранный на втором уровне тип станка или схема установки детали в ряде случаев может уточняться при проектировании операционной технологии на третьем уровне.

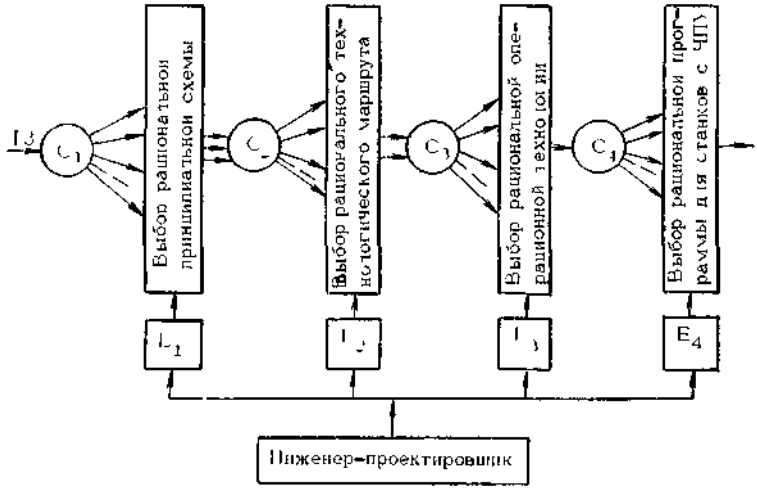


Рис. 14.6. Модель многоуровневого процесса проектирования с пороговыми отборами рациональных решений на каждом уровне

Третья модель является дальнейшим развитием предшествующих и основана на накоплении и обобщении опыта образования фреймов и совершенствовании на этой основе структуры и параметров алгоритмов синтеза и критериев промежуточного самоотбора. Эффективность второй модели образования фреймов можно повысить, если на каждом уровне ввести контур самообучения, включающий операции «накопление» *U* и «обобщение» *T* опыта образования фреймов (рис. 14.3). С помощью операции «накопление» отбираются и заносятся в оперативную память системы только оригинальные решения и процессы, которые до этого отсутствовали в базе данных.

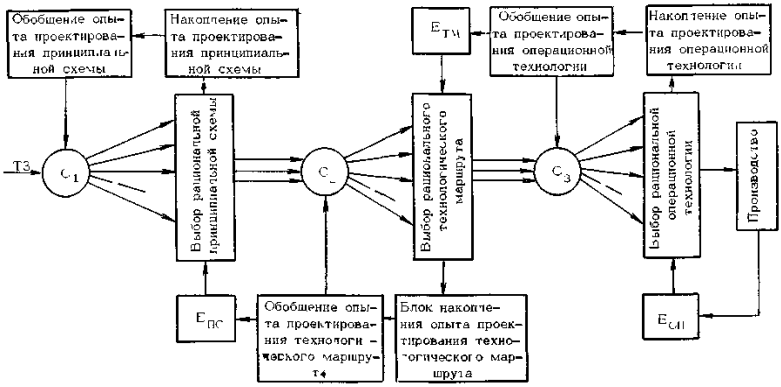


Рис. 14.3. Модель автоматизированной системы проектирования с накоплением и обобщением опыта проектирования на каждой стадии

Производится учет частоты использования типовых, групповых алгоритмов и процессов-аналогов.

Операцией «обобщение» накопленного опыта из числа ранее образованных фреймов формируются типовые решения образования фреймов, типовые и групповые алгоритмы. Улучшается значение эвристических критериев самоотбора, совершенствуются структура и параметры алгоритмов синтеза, оптимизации и анализа. Обобщение накопленного опыта производится в режиме человеко-машинного проектирования с оперативным отображением процессов-аналогов на терминалы устройств отображения.

В результате обучения и самообучения алгоритмы синтеза решений образования фреймов и эвристические критерии промежуточного самоотбора становятся более эффективными. Вместо генерирования большого числа возможных вариантов целенаправленно, с учетом положительного прошлого опыта синтезируется небольшое количество наиболее перспективных вариантов образованных фреймов. За счет улучшения значений эвристических критериев в процессе

самообучения на каждой промежуточной стадии отбирается для дальнейшего образования фреймов меньшее, чем прежде, число наиболее рациональных вариантов.

Таким образом, контур самообучения, работающий на основе использования опытаразработчика, позволяет повысить качество решений образования фреймов и резко сократить затраты машинного времени.

В результате целенаправленного синтеза и промежуточного отбора на каждом уровне генерируются не все возможные варианты, а только наиболее перспективные. Они могут иметь недостатки, которые выявляются с помощью операций анализа и оценки, а затем устраняются алгоритмами оптимизации.

Аналогичное положение наблюдается при автоматизации проектирования путем преобразования процессов-аналогов образования фреймов. В результате приходим к необходимости организации итерационной модели процесса образования фреймов, основной чертой которой является последовательное улучшение исходного варианта до требуемой степени совершенства.

Сформулируем утверждение, определяющее характер процесса образования фреймов на каждом уровне.

Утверждение 2.5. Прцесс образования фреймов на каждом уровне расчленяется на совокупность операций образования фреймов, итерационно взаимосвязанных между собой и осуществляющих формирование множества вариантов образования фреймов, их анализ, оптимизацию и отбор.

Общая схема итерационного алгоритма образования фреймов приведена на рис. 14.4.

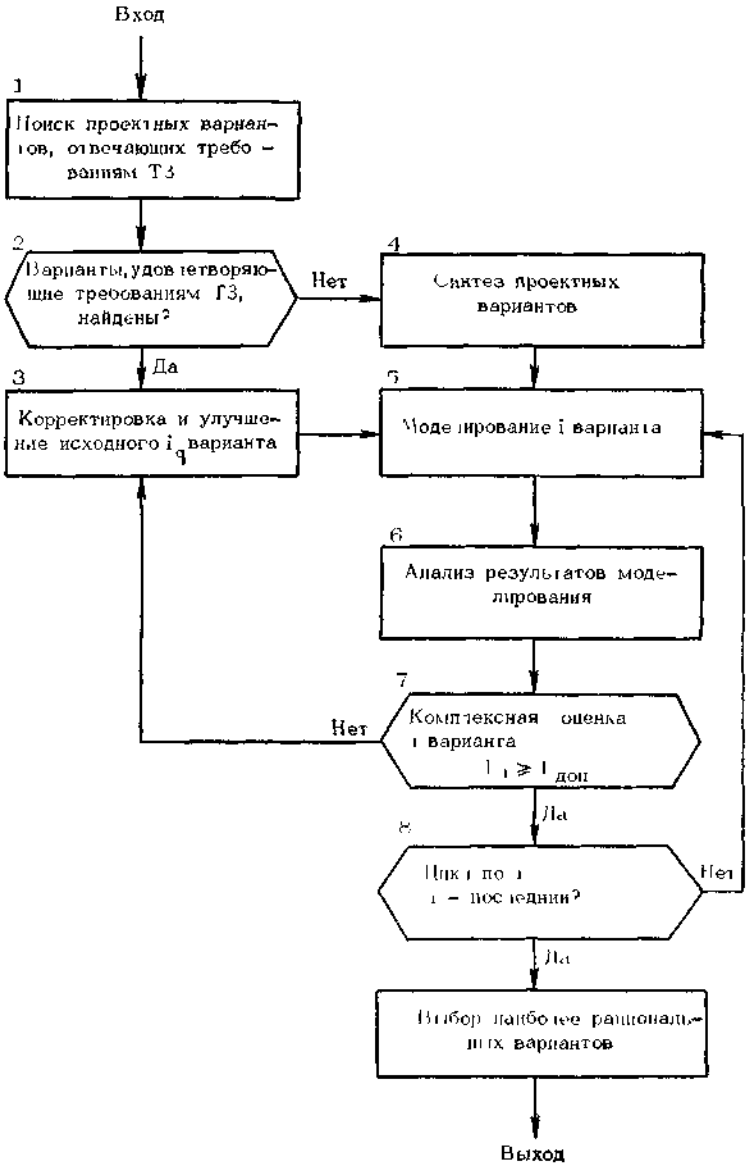


Рис. 14.4. Итерационный алгоритм процесса образования фреймов на каждом уровне

Основными структурными элементами являются такие операции образования фреймов, как поиск решений-аналогов, синтез различных вариантов, имитационное моделирование процесса образования фреймов или отдельного его элемента, анализ и оценка результатов моделирования, оптимизация и отбор наиболее рациональных вариантов.

Процесс образования фреймов начинается с операции поиска процессов-аналогов образования фреймов в массиве накопленного опыта информационного банка данных. Если такие процессы найдены, то логическим блоком *2* управление передается блоку образования фреймов по аналогии, если не найдены, то управление передается блоку синтеза. В этом блоке целенаправленным способом синтезируется некоторое количество вариантов процесса образования фреймов, удовлетворяющих заданным техническим требованиям и ограничениям.

Для того чтобы оценить тот или иной вариант, необходимо построить информационную модель процесса образования фреймов. Блок имитационного моделирования позволяет опробовать любой вариант процесса образования фреймов в модельном эксперименте и тем самым прогнозировать характер функционирования образованого фрейма.

В блоке анализа полученные при моделировании характеристики исследуются на соответствия их с заданными техническими условиями, выясняются причины возникновения тех или иных отклонений. Анализ производится по всем технико-экономическим показателям. В результате вычисляются значения локальных критериев и их взаимосвязь. В блоке оценки на основе выявленных локальных критериев определяется интегральный критерий качества того или иного варианта. Оценивается степень совершенства рассматриваемого варианта по сравнению с другими, определяется необходимость улучшения тех или иных его показателей. В блоке оптимизации производится выбор направления улучшения варианта образованного фрейма в соответствии с моделью, характеризующей взаимосвязь локальных критериев. В результате проведенных преобразований исходного варианта возникает новый улучшенный вариант. Сведения о нем снова поступают в блок моделирования, анализа и оценки. За каждый итерационный цикл процесса образования фреймов качество исходного варианта улучшается. Процесс последовательного улучшения оканчивается, когда вариант по всем основным показателям удовлетворяет заданным требованиям и дальнейшее его совершенствование не приводит к существенному улучшению интегрального критерия.

С помощью блока *8* осуществляется циклическое повторение операций моделирования, анализа, оценки и оптимизации для всех вариантов, полученных в блоках поиска или синтеза. В результате для операций «выбор» подготавливается множество целесообразных вариантов, из которых затем отбираются наиболее рациональные. На промежуточных уровнях для дальнейшего процесса образования фреймов выбирается несколько наиболее рациональных вариантов, а на последнем уровне — один окончательный вариант.

В приведенной модели можно выделить две ветви процесса образования фреймов. Первая включает в себя блоки 1*, 2, 3, 5, 6, 7, 8, 9* и служит для процесса образования фреймов на основе преобразования процессов-аналогов, в том числе типовых и групповых. Во вторую ветвь входят блоки *4 — 9,* они и определяют набор операций образования фреймов, используемый при образовании фреймов методом целенаправленного синтеза. Обе ветви имеют общие блоки, связанные с операциями моделирования, анализа, оценки и выбора рациональных вариантов. Различаются эти ветви операциями генерирования исходных вариантов. В первом случае — это поиск решений-аналогов и их преобразование, а во втором — целенаправленный синтез решений образования фреймов. Остальные операции, связанные с улучшением и оптимизацией исходного варианта, имеют одинаковое назначение. Рассмотрим структуру и содержание основных проектных операций образования фреймов.

14.2. Примеры бразования фреймов технологических операций

14.2.1. Фрейм операции «Поиск решений аналогов»

В различных отраслях промышленности накоплен большой опыт проектирования технологических процессов и оснастки для изготовления разнообразных объектов. В руководящих материалах, типовых и групповых технологических процессах этот опыт в определенной степени систематизируется и становится достоянием технологов. Однако большая часть сведений о прошлых разработках, связанная с созданием технологии изготовления сложных (нетиповых) объектов, при повторном проектировании используется недостаточно. Основная причина такого положения — несовершенная система накопления, хранения и поиска технологической документации. Это приводит к тому, что разыскать в архиве спроектированные ранее

технологические процессы и оснастку на подобные объекты значительно труднее, чем разработать их заново. Для того чтобы в АСТП сведения о прошлых разработках оперативно использовались инженерами-технологами и программами проектирования, необходимо автоматизировать процедуры накопления, хранения и поиска технологической информации и создать банк фреймов технологический процессов и операций.

С помощью операции «поиск» на любом уровне проектирования по заданной совокупности признаков в информационном массиве системы отыскиваются соответствующие им объекты-аналоги и фреймы технологических процессов их изготовления. В основе этой операции лежит принцип технологического подобия обрабатываемыхобъектов, технологических процессов и их элементов. Он заключается в следующем. Сложные объекты и процессы B_1 и B_2 считаются технологически подобными $B_1 \backsim_\tau B_2$, если их различия в функции, структуре и свойствах не превышают некоторых допустимых величин

$$F_{B_1} - F_{B_2} \leqslant \Delta F,$$
$$S_1(X, U) - S_2(X, U) \leqslant \Delta S(X, U),$$
$$Z_{B_1} - Z_{B_2} \leqslant \Delta Z,$$

при которых искомый технологический процесс или его элементы могут быть получены путем преобразований структуры и характеристик процессов-аналогов. Диапазон изменения величин ΔF, $\Delta S\ (X, U)$ и ΔZ устанавливается заранее технологом или специальными алгоритмами в зависимости от выбранных целей проектирования. Рассмотрим более подробно технологическое подобие обрабатываемых деталей и элементов технологического процесса.

Детали $Д_1$ и $Д_2$ или их межоперационные состояния будем считать технологически подобными, если различия $\Delta S\ (П, U)$ в графах $S_1(П, U)$, $S_2(П, U)$, описывающих форму, размерные и точностные связи элементов детали и значения параметров (габаритные размеры — *Р,* материал — *МТ,* физико-механические свойства — *ФМ* и величины партии запуска — *N),* таковы, что не оказывают существенного влияния на структуру и параметры технологического процесса их изготовления:

$$S_1(П,\ U) - S_2(П,\ U) \leqslant \Delta S(П,\ U),$$
$$P_1 - P_2 \leqslant \Delta P,$$
$$MT_1 - MT_2 \leqslant \Delta MT, \qquad (3)$$
$$ФM_1 - ФM_2 \leqslant \Delta ФM,$$
$$N_1 - N_2 \leqslant \Delta N.$$

В ряде случаев для того чтобы установить технологическое подобие обрабатываемых деталей, следует сравнивать выполняемые ими функции, так как они оказывают влияние на выбор методов обработки и структуру технологического процесса.

Для выявления технологического подобия функций двух структурных элементов процесса $F_1 \backsim_{\tau} F_2$ необходимо, чтобы в выражениях φ_i: $C_{i-1} \to C_i$ и φ_j: $C_{j-1} \to C_j$, выполнялись следующие условия: виды преобразований (типы операций или переходов) должны быть одинаковыми $\varphi_i = \varphi_j$; характеристики, описывающие состояния деталей на предшествующей и выполняемых операциях, должны отличаться не более некоторых наперед заданных величин, т. е.

$$C_{i-1} - C_{j-1} \leqslant \Delta C_1,$$
$$C_i - C_j \leqslant \Delta C_2.$$

Последние условия будут достигнуты, если выполняются соотношения (3).

Технологическое подобие структур двух элементов процесса обеспечивается, если графы функциональной, временной и пространственных структур этих элементов отличаются не более некоторых допустимых величин

$$S^1_{ф}(C,\ A) - S^2_{ф}(C,\ A) \leqslant \Delta S_{ф}(C,\ A),$$
$$S^1_{в}(A,\ \Omega) - S^2_{в}(A,\ \Omega) \leqslant \Delta S_{в}(A,\ \Omega).$$

Элементы двух систем СПИД (станок, приспособление, инструмент) технологически подобны $СТ_1 \backsim СТ_2$, $ПР_1 \backsim ПР_2$, $ИН_1 \backsim ИН_2$, если различия в выполняемых функциях и значениях параметров технических характеристик не превышают заранее установленных величин.

Таким образом, при поиске деталей-прототипов, процессов-аналогов, оборудования и элементов оснастки наряду с описанием их системных характеристик *F, S, Z* необходимо указывать допустимые отклонения в функции ΔF, структуре ΔS и свойствах ΔZ.

В зависимости от заданных целей поиск объектов и процессов-прототипов может осуществляться на основе технологического подобия одной или нескольких системных характеристик:

1) набора признаков Z. Объекты $B_1 \backsim_Z B_2$, если $Z_1 - Z_2 \leq \Delta Z$;

2) выполняемых функций *F*. Объекты $B_1 \backsim_F B_2$, если $F\text{-}F \leq \Delta F$;

3) структуры объектов и процессов *S(X, U)*. Объекты $B_1 \backsim_S B_2$, если $S_1(X, U) - S_2(X, U) \leqslant \Delta S(X, U)$;

4) выполняемых функций и набора признаков. Объекты $B_1 \backsim_{F,Z} B_2$, если $F_1 - F_2 \leqslant \Delta F$ и $Z_1 - Z_2 \leqslant \Delta Z$;

5) структуры и набора признаков. Объекты $B_1 \backsim_{S,Z} B_2$, если $S_1(X, U) - S_2(X, U) \leqslant \Delta S(X, U)$, $Z_1 - Z_2 \leqslant \Delta Z$;

6) функции и структуры. Объекты если $B_1 \backsim_{F,S} B_2$, $F_1 - F_2 \leqslant \Delta F$, $S_1(X, U) - S_2(X, U) \leqslant \Delta S(X, U)$;

7) функции, структуры и набора признаков, т. е. по полному системному описанию объекта.

Указанные наборы системных характеристик и допустимых отклонений образуют поисковые образы *H* объектов и процессов. В связи с этим применяемые методы поиска только по набору признаков недостаточны для нахождения в массиве накопленного опыта сложных объектов и процессов-аналогов. Поиск представляет собой преобразование *Г(H)* исходного массива *U* с целью выделения объектов $\{U_k\}$, технологически подобных по заданным системным характеристикам поисковому образу *H:*

$$\Gamma(H) : U \rightarrow \{U_k\}.$$

В общем случае операция «поиск» включает в себя следующие процедуры: формирование поискового образа *H*; определение различий между функцией, структурой и характеристиками выделенного объекта и поискового образа ΔF, ΔS, ΔZ; оценку и принятие решения о технологическом подобии объекта поисковому образу $U_k \backsim H$.

Операция «поиск» иерархическая. Она построена так, что в процессе поиска исходный массив резко сужается. Вначале осуществляется поиск по главным признакам, например по общности выполняемых функций. В результате из массива *U* выделяется подмассив объектов U_F, функции которых технологически подобны поисковому образу. В полученном массиве определяется совокупность объектов $U_{F,S}$ с технологически подобной структурой, а в ней — объекты, подобные по набору характеристик $U_{F,S,Z}$.

Фрейм операции «поиск» имеет самостоятельное значение. В результате ее выполнения может быть найдено решение, удовлетворяющее запросу, или она используется как процедура, входящая в состав других фреймов операций, например в фрейм операции «Синтез проектных вариантов».

14.2.2. Фрейм операции «Преобразование процессов-аналогов»

После того как найдены детали и процессы-аналоги $(D,ТП)_{ан}$, проектирование заключается в том, чтобы на основе системных моделей и заданного конечного состояния конкретной детали $С_к$ определить рациональную структуру и параметры процесса ее обработки

$$W : \{C_k,\ (D,\ ТП)_{ан}\} \rightarrow ТП_k,$$

где W — операция преобразования процессов-аналогов. Преобразование осуществляется методами исключения или дополнения структурных элементов в процессы-аналоги на основе выявления различий между обрабатываемой деталью и состояниями детали-аналога. Метод исключения структурных элементов основан на том, что из графа структуры процесса-аналога $S^а_ф$ (С, *АЛ*) исключаются некоторые пути или дуги $\{C_{q-1}A_qC_q\}$, соответствующие операциям или переходам обработки отсутствующих у конкретной детали поверхностей или поверхностей более высокой точности. Структура конкретного процесса образуется в результате операции разности графов

$$S^к_ф(C_1,\ A_1) = S^а_ф(C,\ A) \setminus \{C_{q-1}A_qC_q\},$$

где $C_1 = C \setminus \{C_q\};\ A_1 = A \setminus \{A_q\}.$

В результате применения этой операции структура конкретного процесса получается более простой, чем аналога.

Метод дополнения структурных элементов базируется на присоединении к графу структуры процесса-аналога $S^а_ф(C,\ A)$ множества дуг $\{C_{r-1}A_rC_r\}$, соответствующих вновь вводимым операциям и переходам по обработке поверхностей конкретной детали, которые отсутствуют в деталях-аналогах или имеют более низкую точность. Структура конкретного процесса получается более сложной и образуется в результате операции объединения графов

$$S^к_ф(C_1,\ A_1) = S^а_ф(C,\ A) \cup \{C_{r-1}A_rC_r\},$$

где $C_1 = C \cup \{C_r\};\quad A_1 = A \cup \{A_r\}.$

В ряде случаев возникают задачи проектирования, когда преобразование процессов-аналогов производится тем и другим методами

$$S^к_ф(C_1,\ A_1) = S^а_ф(C,\ A) \setminus \{C_{q-1}A_qC_q\} \cup \{C_{r-1}A_rC_r\}.$$

Преобразование процесса-аналога методом исключения структурных элементов осуществляется установлением технологического подобия состояний детали-аналога со структурой и параметрами конкретной детали. Для этого в графе функциональной структуры технологического процесса-аналога выделяются висячие вершины, соответствующие конечным состояниям групп обрабатываемых поверхностей. Если $C_j^{а} \infty C_j^{к}$, то операция A_j включается в маршрут обработки конкретной детали, в противном случае она исключается из маршрута-аналога.

Технологическое подобие $C_j^{а}$ и $C_j^{к}$ состояний детали-аналога и обрабатываемой детали устанавливается на основе сравнения множества видов обрабатываемых поверхностей $M^{к}$, $M^{а}$ и точности их размеров:

$$M^{к} \subset M^{а},$$

$$T_i^{к} - T_i^{а} \leqslant \Delta T,$$

$$Ш_i^{а} - Ш_i^{к} \leqslant Ш,$$

$$\delta_i^{к} - \delta_i^{а} \leqslant \delta.$$

Если точностные параметры обрабатываемой детали превышают установленный порог ΔT, $\Delta Ш$, $\Delta\delta$, то рассматриваемая операция исключается, так как требуемая точность обработки может быть получена на предшествующей операции, т. е. в состоянии C_{j-1}.

Пороговые значения ΔT, $\Delta Ш$, $\Delta\delta$ устанавливают интервалы классов точности шероховатости по двум соседним операциям процесса. Например, если заключительная операция выполняется по второму классу точности, а предшествующая — по четвертому, то ΔT будет лежать в интервале второго-четвертого классов точности.

Преобразование процесса-аналога методом дополнения структурных элементов (операций, переходов) заключается в определении вида и количества этих элементов и рациональном их расположении среди операций процесса-аналога. При этом могут быть следующие варианты структурных дополнений. Если технологические возможности оборудования и оснастки операций процесса-аналога обеспечивают обработку дополнительных поверхностей, то они включаются в их состав

$$A_k = \{a_i\} \cup \{a_r\},$$

где $\{a_r\}$ — множество переходов по обработке дополнительных поверхностей; $\{a_i\}$ — множество переходов, определяющее технологические возможности операции-аналога; A_k — состав

переходов в операции после ее дополнения переходами по обработке дополнительных поверхностей.

При невыполнении этого условия обработка дополнительных поверхностей выделяется в самостоятельные операции.

Если точностные параметры обрабатываемых поверхностей конкретной детали выше, чем у идентичных поверхностей детали-аналога, то возможны два варианта. Если технологические возможности операции процесса-аналога позволяют произвести обработку более точных поверхностей конкретной детали, то их обработка включается в одну из операций процесса-аналога. В противном случае в проектируемый технологический процесс вводятся операции обработки дополнительных поверхностей. Решение указанных задач для деталей сложной формы трудноформализуемо и поэтому должно производиться в диалоговом режиме.

14.2.3. Фрейм операции «Синтез технологических процессов»

На различных уровнях проектирования исходными данными в задачах синтеза служит описание функции технологического процесса или отдельных его элементов (этапов, операций, переходов) φ_i^k: $C_{i-1}^k \rightarrow C_i^k$. В этом выражении известны вид преобразования φ и параметры состояния C_i^k, которые необходимо получить. Задача заключается в том, чтобы на основе этих данных определить допустимые техническими ограничениями варианты структуры процесса обработки $S_ф$ *(C, A),* параметры предшествующего состояния C_{i-1}^k и функции F_q^{k-1} составляющих процесс обработки структурных элементов *k*—1 уровня.

Операция «синтез» базируется на декомпозиции заданной сложной функции на отдельные подфункции, а эти последние — на еще более простые подфункции, и так до тех пор, пока не будут получены функции базовых структурных элементов требуемого уровня детализации

$$\mathscr{P}^k : F_r^k \rightarrow \{F_{q1}^{k-1},\ F_{q2}^{k-1} \ldots F_m^{k-1}\},$$

где $\mathscr{P}^k$ — процедура декомпозиции функции на *k* уровне проектирования; F^k_r — функция *r* элемента, подлежащая расчленению; $\{F_m^{k-1}\}$ — множество подфункций *k*—1 уровня, на которые расчленена F^k_r .

В результате декомпозиции строится граф функций проектируемого структурного компонента процесса *T(F, R),* вершинам которого

соответствуют функции отдельных элементов F_q, а дугам — отношения включения одних функций в другие.

Вторым звеном синтеза является выбор структурных элементов процесса, посредством которых заданная функция может быть реализована. Ранее было показано, что одна и та же функция может быть осуществлена с помощью различных структурных элементов. Поставив в соответствие каждой функции F_j^{k-1} один или несколько элементов процесса, обеспечивающих ее реализацию, получим различные варианты структурного состава процесса обработки

$$F_j^{k-1} \rightarrow \{A_{q1}^{k-1},\ A_{q2}^{k-1},\ \ldots,\ A_{ql}^{k-1}\}.$$

Третьим звеном синтеза служит процедура формирования пространственно-временных связей между отдельными структурными элементами и получения графа возможных вариантов технологического процесса. Она характеризуется преобразованием массива элементов процесса $\{A_{ql}^{k-1}\}$ в графы возможных вариантов функциональной пространственной и временной структур технологического процесса на k уровне:

$$У_{ф} : \{A_{q,l}^{k-1}\} \rightarrow S_{ф}(C,\ A),$$

$$У_{в} : S_{ф}(C,\ A) \rightarrow S_{в}(A,\ \Omega),$$

$$У_{п} : \{S_{ф}(C,\ A),\ S_{в}(A,\ \Omega)\} \rightarrow S_{п}(B,\ L).$$

Декомпозиция общей функции процесса, как известно, может осуществляться различными способами. При этом получается разный состав структурных элементов и связи между ними, т. е. различные варианты структуры процесса. Для того чтобы сделать синтез целенаправленным и эффективным, декомпозицию общей функции, выбор структурных элементов и установление связей между ними необходимо производить, руководствуясь принципом минимальных затрат на технологическую совместимость. Реализация его требует максимального соответствия между компонентами процесса. Следуя этому принципу, варианты маршрута синтезируются так, чтобы обеспечить совместимость выбранных методов обработки и характеристик детали с технологическими возможностями имеющегося оборудования (совместимость по оборудованию); конструкторских баз с технологическими при выборе схем базирования и последовательности операций.

На основе полученных вариантов структуры технологического процесса производится расчет параметров предшествующего состояния C_{i-1}^k обрабатываемой детали. Степень детализации и точность параметров C_{i-1}^k состояния на различных уровнях

проектирования разные. На первом уровне определяются виды заготовок и точность обработки детали на разных этапах технологического процесса.

На следующих уровнях проектирования точность параметров C_{i-1}^{k} состояния детали возрастает, достигая наибольшей при проектировании операционной технологии и управляющих программ для станков с ЧПУ.

Таким образом, фрейм операции синтеза на каждом уровне проектирования состоит из процедур выбора метода декомпозиции функции технологического процесса или отдельных его частей на более простые подфункции; декомпозиции общей функции на отдельные подфункции в соответствии с выбранным методом; выбора структурных элементов процесса, обеспечивающих реализацию каждой из этих подфункций; формирования пространственно-временных связей между отдельными структурными элементами и получения графа возможных вариантов технологического процесса; расчета параметров предшествующего C_{i-1} состояния обрабатываемой детали.

14.2.4. Фрейм операции «Имитационное моделирование»

Фрейм операции моделирования включает в себя следующие процедуры: построение имитационных моделей процесса обработки, описывающих явления, которые необходимо учитывать при проектировании на различных уровнях; определение законов изменения входных воздействий (параметров деталей-заготовок), характеристик систем СПИД и технических ограничений; прогнозирование достижимой точности обработки, надежности ее осуществления, производительности и эффективности технологического процесса.

Прогнозирование осуществляется имитацией на математических моделях процесса обработки при вариациях параметров окружающей среды и характеристик систем СПИД. Точность прогноза зависит от того, насколько модель процесса и выявленные законы изменений характеристик окружающей среды и системы СПИД адекватны реальным явлениям, происходящим в процессе производства.

На уровне проектирования принципиальных схем ввиду незначительной детализации разрабатываемого технологического процесса возможности математического моделирования ограничены. Оно сводится к эвристическому прогнозу того, как влияют выбранные

этапы и методы обработки на возможность эффективного достижения требуемой точности размеров поверхностей и их взаимного расположения. Для деталей сложной конфигурации в связи с недостаточностью сведений о технологическом процессе операция моделирования на первом уровне не поддается формализации и должна выполняться технологом.

На уровне проектирования маршрута обработки математической моделью служит граф функциональной структуры технологического процесса. Он характеризует взаимосвязь выбранных базовых и обрабатываемых поверхностей и порядок смены базовых поверхностей. Моделирование включает в себя расчет погрешностей базирования и их влияние на размеры, выдерживаемые в операциях. Вычисляются ориентировочное время и себестоимость обработки детали.

Математической моделью операции на третьем уровне проектирования служат графы пространственно-временной структуры операции и модель процесса резания. По графу пространственной структуры операции определяются погрешности размеров в технологической размерной цепи, а по графу временной структуры — время и себестоимость выполнения операции. Модель резания представляет собой систему соотношений, связывающую режимы обработки n, s, t с параметрами системы СПИД. С помощью этой модели находятся допустимые системой СПИД и заданной точностью межоперационных размеров режимы резания и динамические погрешности, возникающие в процессе обработки.

При необходимости разработки управляющих программ для станков с ЧПУ модель процесса резания переносится в четвертый уровень проектирования. В рассмотренных операционных моделях некоторые величины из-за отсутствия точных значений могут задаваться их вероятностями. Однако в целом точность операционных моделей значительно выше, чем моделей маршрутной технологии.

Таким образом, при переходе от одногб уровня проектирования к другому степень детализации и точности моделей процесса возрастает от эвристических на первом уровне до достаточно точных математических моделей, отражающих статические и динамические характеристики процесса на заключительных уровнях проектирования.

Структурные модели процесса и его элементов при решении сложных задач, связанных с расчетом погрешностей при различных схемах базирования, компоновкой инструментальной наладки и расчетом траекторий обработки сложных участков детали на станках с ЧПУ, целесообразно выводить на экраны устройств отображения или вычерчивать с помощью чертежных автоматов. Графическое

моделирование таких задач дает возможность технологу выявить недостатки в структуре проектируемого варианта процесса и устранить их.

14.2.5. Фрейм операции «Анализ проектных вариантов»

Полученные при моделировании характеристики технологического процесса исследуются на их соответствие заданным техническим условиям. Определяются причины возникновения тех или иных отклонений. В результате анализа обнаруживаются слабые места и невыполнимые условия в рассматриваемом варианте технологического процесса. Анализ производится по всем основным технико-экономическим показателям процесса: точности обработки, производительности, себестоимости; выявляются значения этих показателей и характер их взаимосвязи и влияние друг на друга. Точность анализа возрастает при переходе от начальных уровней проектирования к заключительным.

На первом уровне проектирования определяется степень соответствия выбранных методов и этапов обработки требуемой точности и производительности. Полученные при моделировании значения этих характеристик процесса носят сугубо ориентировочный характер. Поэтому и результаты анализа помогают вскрыть только крупные отклонения и их причины. На уровне проектирования операционной технологии производится анализ выбранной системы СПИД, пространственной и временной структур операции, режимов обработки, времени выполнения операции и ее себестоимости. Определяется влияние отдельных составляющих погрешности и принятого порядка выполнения переходов на результирующую точность и производительность обработки. При анализе математической модели расчета режимов резания выявляются технические ограничения, которые лимитируют режимы резания и др.

В связи с большой степенью детализации проектных решений точность анализа на этом уровне проектирования высока. В операции «анализ» можно выделить три проектные процедуры: сравнение полученных при моделировании значений показателей процесса обработки с допустимыми и определение отклонений; выявление причин отклонений от требуемой точности и производительности; установление характера взаимосвязей отдельных показателей и построение модели, определяющей взаимное влияние показателей друг на друга.

Поиск причин возникновения отклонений от требуемой точности, производительности и построение модели взаимного влияния показателей друг на друга будем называть технологической диагностикой. Правильная постановка технологического диагноза действующему или спроектированному варианту процесса обработки является основой для выработки направлений по его совершенствованию.

Часть задач технологической диагностики может быть формализована и выполнена специальными алгоритмами. В то же время на различных уровнях проектирования существуют задачи, формализация которых в настоящее время затруднена. В связи с этим сложные задачи анализа и технологической диагностики должны решаться в режиме человеко-машинного проектирования.

14.2.6. Фрейм операции «Оценка вариантов»

На всех уровнях проектирования необходимо принимать решения по выбору наиболее рациональных вариантов технологических процессов или отдельных его элементов. На начальных уровнях недостаточной детализации проектируемого процесса критерии ввиду отбора носят эвристический характер.

Так, на первом уровне проектирования оценка вариантов принципиальных схем обработки детали и отбор наиболее рациональных основаны на весьма приближенных эвристических критериях. На втором уровне критерии отбора более точные, чем на первом. Они позволяют выбрать рациональные схемы базирования деталей и технологический маршрут обработки. На третьем уровне применяются еще более точные критерии, которые ближе всего связаны с технологической себестоимостью и обеспечивают выбор вариантов операционной технологии, близких к технологически оптимальным. Критерии четвертого уровня позволяют выбрать оптимальные траектории движения режущего инструмента и детали при обработке на станках с ЧПУ.

На различных уровнях проектирования необходимо принимать такие решения, которые были бы наилучшими по всей совокупности выявленных в блоке анализа значений локальных критериев, связанных с себестоимостью, производительностью, надежностью обеспечения заданной точности и др. Оптимальные значения локальных кршернси обычно не совпадают, кроме того, они имеют различные единицы измерения.

В связи с этим оценка проектного варианта производится по обобщенному (интегральному) критерию качества, объединяющему все локальные критерии. При этом компромиссное решение может быть неоптимальным ни по одному локальному критерию, но наилучшим по их совокупности. Интегральный критерий следует сформировать так, чтобы можно было сравнивать показатели с различным технологическим смыслом и размерностью.

В соответствии с теорией аддитивной полезности локальные критерии в первом приближении считаются независимыми, а обобщенный критерий W может быть получен суммированием характеристик, отражающих полезность отдельных критериев w:

$$W = \sum_{i=1}^{q} \lambda_i f_i (w_i),$$

где λ_i — весовые коэффициенты локальных критериев, с помощью которых задается степень их важности λ_i = 0-1; $f_i(w_i)$ —функция полезности для i локального критерия.

Функция полезности характеризует оценку P_w в баллах (по десятибалльной шкале) того или иного критерия качества в зависимости от изменений его значений. Значения λ_s и $f_s(w_s)$ задаются технологом на стадии разработки системы и затем могут корректироваться. Предполагается, что инженер-технолог отчетливо представляет цели проектирования и может однозначно задать значения весовых коэффициентов λ_s по всем критериям w_s. На основе анализа опыта проектирования выявляется вид функции полезности $f_s(w_s)$ для каждого критерия. Графически ее можно выразить в координатах P_w, W.

В задачах технологического проектирования наиболее распространены следующие виды оптимизации: по одному критерию W_{sk}, для остальных λ_s = 0; по одному какому-нибудь критерию при условии, что другие показатели имеют значения не хуже заданных, т. е.

$W_{sk} \rightarrow \min$ при $W_{s1} \leqslant W_{s1\,доп}, \ldots, W_{sn} > W_{sn\,доп}$, по обобщенному критерию с учетом всех локальных.

Первый случай применяется, например, при определении оптимальных параметров режимов обработки, когда структура операции и системы СПИД уже выбрана. Тогда технологическая себестоимость или производительность являются основными критериями, по которым производится выбор режимов резания. Второй случай часто встречается на более ранних стадиях проектирования, когда среди нескольких критериев можно выделить главный и установить пороговые значения для остальных показателей. В качестве оптимального принимается вариант, имеющий наибольшее значение

главного показателя при условии, что остальные будут иметь значения не хуже пороговых.

Операция «оценка вариантов» включает в себя процедуры определения значений локальных критериев W_i и весовых коэффициентов λ_i вида функции полезности $f_i(w_i)$ для каждого критерия; нахождения относительных оценок критериев P_{ws} в баллах и вычисления интегрального критерия

$$W = \sum_{i=1}^{q} \lambda_i P_{wi}.$$

При проектированиитехнологических процессов обработки сложных оригинальных деталей формализация оценки вариантов затруднена. В таких случаях должен применяться режим диалога, обеспечивающий участие технолога в окончательной оценке и выборе вариантов проектных решений.

14.2.7. Фрейм операции «Корректировка и улучшение» проектного варианта

Задача корректировки и улучшения исходного варианта процесса обработки возникает тогда, когда в результате анализа и комплексной оценки установлено, что этот вариант не полностью удовлетворяет поставленным требованиям и необходимо улучшить его структуру и значения отдельных параметров. На основе выявленных в блоке анализа причин возникновения этих отклонений определяются решения, с помощью которых можно улучшить характеристики проектируемого технологического процесса или его элементов. Системный подход здесь состоит в том, чтобы для каждого вида недостатков установить возможные причины, их вызывающие. Сложные причины необходимо разбить на более простые, каждой из них поставить в соответствие технологические решения по их устранению, а затем выбрать рациональные по корректировке и оптимизации сходного варианта.

На различных уровнях проектирования применяются разные методы корректировки и оптимизации исходного варианта фрейма операции. Так, на первом уровне корректировка осуществляется измерением состава этапов, методов обработки и вида заготовки, переносом методов чистовой обработки из одного этапа в другой. На втором уровне корректируются состав операций, их последовательность, установочные базы и характеристики системы СПИД. При образовании фреймов операционных технологий на третьем уровне

возможна корректировка порядка выполнения установок и переходов межоперационных размеров и допусков, режимов резания и др. Например, в фреймах анализа и оценки могут быть установлены пути сокращения числа переустановок детали, если частично отступить от принципа совмещения баз. В фрейме корректировки определяется новый вариант базирования, производится перерасчет размерных цепей и допусков, изменяется последовательность переходов. В фреймах моделирования, анализа и оценки определяются возможность достижения новых допусков принятыми методами и дополнительные затраты, связанные с их ужесточением. Затем они сравниваются с экономией от сокращения времени на переустановку детали и определяется целесообразность изменения исходного варианта базирования.

Для деталей сложной формы процесс корректировки первоначально синтезированных решений трудно формализовать, поэтому должен применяться режим диалога. Технолог может задавать варианты корректировки, а ЭВМ по программам осуществлять анализ и оценку точности, производительность каждого варианта. Результаты расчетов будут выведены на экран устройства отображения для того, чтобы технолог мог принять окончательное решение.

14.3. Диалоговые процедуры многоуровневого итерационного метода образования фреймов

14.3.1. Рациональное разделение функций между разработчиком фреймов и ЭВМ

На различных уровнях образования фреймов степень формализации задач может быть разной. Проектирование процессов образования фреймов на сложные объекты требует непосредственного участия разработчика фреймов при решении трудноформализуемых творческих задач, связанных с синтезом, моделированием, оценкой и выбором решений.

В связи с этим возникает необходимость создания диалоговых систем, обеспечивающих оперативное участие разработчика фреймов в процессе разработки фреймов и управления процессом. В таких системах на основе разделения функций между разработчиком фреймов и ЭВМ трудноформализуемые задачи решаются разработчиком фреймов, а остальные — программным комплексом. Система должна иметь технические средства, обеспечивающие

общение разработчика фреймов с ЭВМ на формализованном языке образования фреймов, близком к естественному.

Для создания систем диалогового образования фреймов необходимо решить следующие основные задачи: в рамках выбранного метода автоматизированного образования фреймов произвести рациональное разделение функций между разработчиком фреймов и программно-техническим комплексом системы; разработать методику диалогового образования фреймов, определяющую процесс взаимодействия разработчика фреймов с ЭВМ и программами решения задач образования фреймов; создать формализованный язык образования фреймов (ФЯОФ) обеспечивающий связь разработчика фреймов с ЭВМ в диалоговом режиме; оснастить рабочее место разработчика фреймов специальными техническими средствами, позволяющими сделать процесс общения естественным и удобным.

Рациональное разделение функций в системе базируется на детальном изучении свойств и возможностей разработчика фреймов, программных и технических средств, применяемых при автоматизации образования фреймов. При этом было бы неправильно исходить из крайних точек зрения, что технические средства могут полностью заменить разработчика фреймов, или в противоположность этому считать, что чисто человеческие функции мышления не могут быть переданы ЭВМ. В действительности как человек, так и ЭВМ имеют свои преимущества и недосгагки и способны взаимно дополнять друг друга, обеспечивая резкое повышение производительности труда и качество решений образования фреймов . Разработчик фреймов лучше ЭВМ решает творческие задачи восприятия и анализа геометрической информации об объектах сложной формы, выбора наиболее рациональных решений в услониях неопределенности и недостаточности сведений о явлениях. В то же время работа разработчика фреймов характеризуется рядом существенных недостатков: небольшой скоростью вычислений и решений несложных, часто повторяющихся задач, возможными ошибками, зависимостью его деятельности от внешних условий, физического и психического состояния, утомляемостью, низкими скоростью и качеством оформления документации.

Вычислительные машины и другие автоматические устройства, уступая человеку в гибкости и широте возможностей, превосходят его в производительности выполнения расчетов и анализа больших массивов справочной информации, выборе решений по известному алгоритму. Эти возможности непрерывно совершенствуются и принципиально трудно указать границы их развития.

Исходя из приведенного анализа возможностей разработчика фреймов и технических средств системы, можно сформулировать общие положения о разделении функций между этими двумя звеньями. ЭВМ необходимо передать максимальное число стандартных, часто повторяющихся информационно-вычислительных задач, на которые в могут быть разработаны программы их решения. За разработчиком фреймов целесообразно закрепить решение таких задач, на которые в трудно или невозможно разработать достаточно четкие алгоритмы.

На первом уровне образования фрейма к таким задачам относятся, например, выбор вида и способа получения заготовок для сложных деталей, определение ряда этапов технологического процесса (черновая обработка, старение и др.). На втором уровне образования фрейма, например, при построении маршрута обработки деталей сложной формы разработчик выбирает установочные базы и порядок их смены; производит оценку и выбор наиболее рационального плана обработки детали в целом. При разработке операционной технологии в функции оператора-технолога входит выбор схемы совмещения переходов и структуры инструментальной наладки. Для остальных задач проектирования маршрутной и операционной технологии, как показывает опыт, могут быть разработаны эффективные алгоритмы и программы.

В результате изучения и анализа решаемых на каждом уровне задач и проектных операций, с помощью которых они реализуются, составляется таблица распределения функций (табл. 1).

Таблица 1

Распределение функций между технологом и ЭВМ при построении маршрута обработки деталей

Состав задачи	Проектные операции		
	синтез	моделирование	выбор варианта
Построение маршрута обработки поверхностей	Ф	—	Ф
Формирование этапов обработки детали	Д	Т	Ф
Определение состава переходов в операциях	Ф	—	Ф
Построение схемы базирования	Д	Ф	Д
Последовательность операций	Ф	—	Ф

По одному ее входу записывается дерево структурного состава решаемых задач, а по другому — проектные операции. На пересечении строки столбцов указывается тип проектной операции *(Ф* — формализуемая, *Т* — решаемая технологом, *Д* — операция, выполняемая в диалоговом режиме). Такие таблицы, составленные для каждого уровня образования фреймов, служат основой для разработки программного комплекса САОФ и выбора рационального режима взаимодействия разработчика фреймов с ЭВМ.

14.3.2. Взаимодействие разработчика фреймов с ЭВМ при различных формах диалога

Методика диалогового образования фреймов характеризует общую схему и способы взаимодействия разработчика фреймов с программами разработки фреймов при различных формах диалога. В зависимости от того, какой элемент системы (разработчик фреймов или ЭВМ) выполняет функции оперативного управления процессом образования фреймов, будем различатьт три формы диалога: под управлением ЭВМ, разработчика фреймов и смешанную.

Первая форма диалога характеризуется наличием управляющего алгоритма, координирующего действия разработчика фреймов и программ образования фреймов, и применяется для решения задач, у которых функции разработчика фреймов и программ четко дифференцированы, определен строгий порядок взаимодействия разработчика фреймов с ЭВМ. Например, для деталей сложной формы выбор вида заготовки должен производиться технологом, а расчет припусков, напусков и размеров заготовки — программами. Связующим звеном между ЭВМ и разработчика фреймов служат фреймы запроса, содержащие формулировку творческой задачи и обращение к разработчику фреймов с просьбой решить ее. Разработка фреймов запросов осуществляется на стадии алгоритмизации задач образования фреймов. В общем виде фрейм запроса задается отображением

$$P{:}УЭ(ИД,\ ОБ) \rightarrow R,$$

где *Р* — творческая процедура построения или выбора решения; *УЗ* — формулировка условия задачи; *R* — выбранное решение.

В условия задачи входят исходные данные *ИД,* имеющиеся к моменту ее решения, технические ограничения *ТО* и обращение к разработчику фреймов *ОБ* с просьбой решить задачу. По характеру задач, содержащихся в фреймах запроса, последние разделяются на три типа: логические фреймы, фреймы выбора решений из заранее заданного множества альтернатив и фреймы с задачами построения решений, вид

и количество которых на стадии алгоритмизации установить невозможно.

Логические фреймы запроса ФЗ-1, приведенные ниже, содержат исходные данные и формулировку вопроса *ФВ,* на который разработчику фреймов необходимо ответить «да», «нет» (1,0):

$$P: ИД,\ ФВ \rightarrow 1 \wedge 0.$$

Задачи второго типа — это выбор решений из заранее заданного на стадии алгоритмизации множества альтернатив:

$$P: УЗ,\ ОБ,\ \{R_1,\ R_2,\ \ldots,\ R_k\} \rightarrow R_i.$$

Разработчик фреймов с помощью функциональной клавиатуры указывает номер решения, который отображается на экране в поле ответов.

Примеры фреймов запросов первых двух типов:

ФЗ-1

Деталь — АМ 4015 1318
Размер партии — 28
«Старение» детали
Необходимо?
Ответ: 1

ФЗ-2

Деталь — АМ 4015 1317
Размер партии — 50
Выберите вид заготовки?
1. Пруток
2. Труба
3. Поковка
Ответ: 2

Третий тип — это задачи вида *В : УЗ→R,* для которых множество ответов на стадии алгоритмизации указать невозможно. От разработчика фреймов в этом случае требуется решить задачу и с помощью функциональной клавиатуры ввести ответ в ЭВМ. При этом выбранное решение отображается на экране дисплея. Примером может служить модуль выбора технологических баз (ФЗ-5). Разработчик фреймов, прочитав условие задачи, выбирает базовые поверхности и сведения о них вводит в ЭВМ.

ФЗ-5

Деталь — АМ 4015 1317
Этап — чистовой
Операция — фрезерная
Обрабатываемые поверхности — *П17, П25*
Выберите базовые поверхности?
Опорная база — *П3*
Направляющая — *П8*
Упорная — *П5*

ФД-15

Найти приспособление
Деталь — АМ 4015 1317
Операция — фрезерная
Станок — 6Н12
Обрабатываемые поверхности — *П17, П25*
Базовые поверхности:
Опорная база — *П3*
Направляющая — *П8*
Упорная — *П5*
Ответ: *ПР171, ПР522, ПР313*

Диалог под управлением разработчика фреймов применяется, когда требуется корректировка промежуточных результатов машинного образования фреймов, а также при решении новых задач на основе имеющихся программных модулей или задании управляющих воздействий, изменяющих направление процесса образования фреймов и др. Общение разработчика фреймов с ЭВМ при этой форме диалога осуществляется фреймами директив. В отличие от фреймов запросов, содержащих формулировку творческой задачи, которую должен решить разработчик фреймов, фрейм директива характеризует формализованную задачу, решаемую ЭВМ. Директива формулируется разработчиком фреймов, а выполняется ЭВМ по имеющимся программам. При вмешательстве в ход образования фреймов разработчик фреймов должен правильно сформулировать директиву и ввести ее в ЭВМ. Затем она транслируется на внутренний язык системы, находится реализующая ее программа и производится расчет. Результаты выполнения директивы отображаются на экран. Фрейм директивы так же, как и фрейм запроса, задается отображением

$$ДР{:}B,\ УЗ \rightarrow R_{\text{В}},$$

ДР — название директивы (определить, найти, печатать, чертить и т. д.); *В* — определяемый объект или объект, над которым производятся манипуляции (тексты, чертежи, графики); $R_{\text{В}}$— результат выполнения директивы, т. е. значения определяемого объекта *В* или полученный текст, чертежи.

По функциональному назначению можно выделить директивы ввода информации в ЭВМ; решения технологических и геометрических задач; редактирования текстовой и графической информации; вывода результатов проектирования на устройства отображения, печати, чертежные автоматы и др. Директивы каждого класса отличаются типом определяемых объектов и условий задачи. В технологических, геометрических и поисковых директивах определяемыми объектами могут быть операции, переходы, схемы базирования, модели оборудования и оснастки, параметры режимов резания, а также элементы траектории движения инструмента и детали на станках с ЧПУ. Условия задачи включают в себя исходные данные и технические ограничения *ТО,* а для многовариантных задач и критерий оптимальности *W*

$$ДР{:}B,\ \{ИД,\ ТО,\ W\} \rightarrow R_B.$$

Фрейм директивы ФД-15 поиска приспособления приведена выше. Разработчик фреймов формирует исходные данные в виде совокупности наименований характеристик операции и их значений. За критерий оптимальности принимается минимальное время установки $t_{\text{уст}} \rightarrow \min$. Ответ ЭВМ представляет собой набор номеров

приспособлений, у которых t близко к t_{mln}, т. е. $t_{уст} \leq kt_{mln}$, где k -коэффициент близости, задаваемый разработчиком фреймов.

В директиве определения траектории фрезы ответ будет представлен в виде чертежа траектории и таблицы координат опорных точек. В директиве печати операционной карты в качестве исходных данных указываются ее форма и обозначение устройства печати. Общая схема взаимодействия разработчика фреймов (технолога) с ЭВМ в режиме диалога изображена на рис. 14.5.

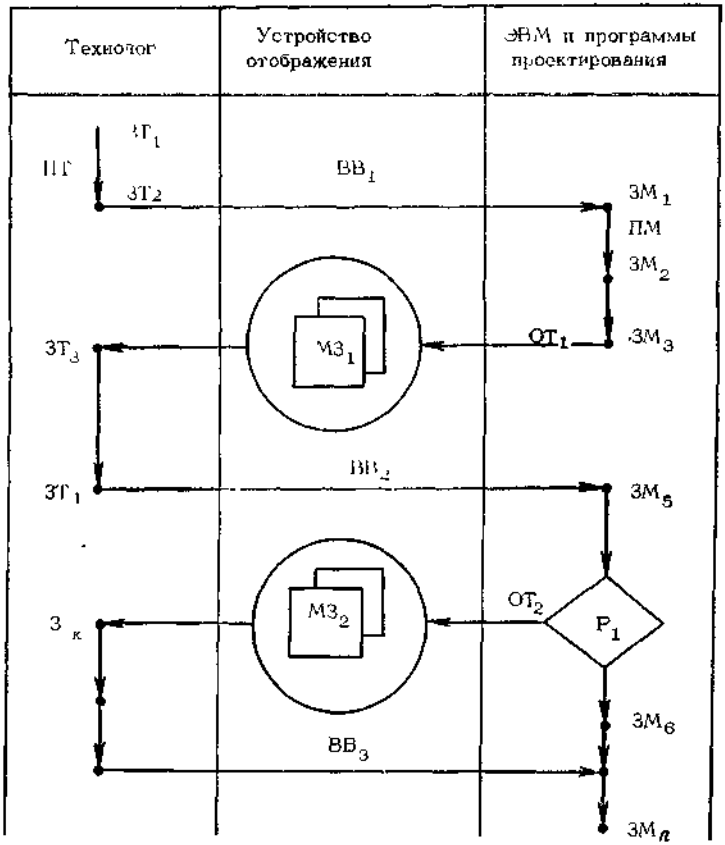


Рис. 9. Схема образования фрейма в диалоговом режиме: *ЗТ, ЗМ* — результаты решения задач соответственно технологом и ЭВМ; *ПТ, ПМ* — процедуры решения задач, выполняемые соответственно технологом и ЭВМ; *ВВ, ОТ*— ввод данных в ЭВМ и отображение на экран модуля запроса *МЗ; P_1* — логический блок анализа сложности

задачи и передачи управления для простых задач программам, а для сложных — технологу

Все задачи технологического проектирования разделены на три группы: формализованные (выполняемые ЭВМ по заранее разработанным программам), творческие (решаемые технологом) и оперативные (процедуры диалога, осуществляемые с помощью дисплея). Горизонтальные стрелки обозначают передачу управления от технолога к ЭВМ и наоборот. В диалоговых алгоритмах, помимо общепринятых обозначений вычислительных и логических операторов, необходимо ввести графические обозначения модулей запросов, модулей директив, операторов отображения информации на экраны и действий, выполняемых технологом.

Процесс проектирования начинается с того, что технолог, сформировав директиву, вводит в ЭВМ информацию о детали. После этого автоматически по ранее разработанным программам решается определенная группа задач $З_1$, $З_2$. Когда появляется необходимость решения незапрограммированной задачи, на дисплей оператором отображения *ОТ* выводится *МЗ* с формулировкой этой задачи. Технолог решает ее и с помощью клавиатуры дисплея вводит ответ в память ЭВМ. На основе полученной информации ЭВМ продолжает дальнейшее проектирование.

В САОФ достигаемая степень автоматизации образования фреймов зависит от сложности задачи. Например, задачи выбора установочных баз, схемы инструментальной наладки для простых деталей могут быть формализованы, а для сложных деталей должны решаться технологом. Анализ сложности конфигурации детали осуществляется логическим оператором P_1, в результате чего для решения одной и той же задачи управление передается программам либо технологу.

В некоторых ветвях процесса образования фреймов разработчик фреймов может сформулировать директивы для ЭВМ по решению интересующих его задач. Диалог продолжается до тех пор, пока не будет разработан фрейм технологического процесса заданной степени детализации, т. е. маршрут или операционная технология. Таким образом, в системе автоматизированного образования фреймов используются все формы диалога.

14.3.3. Формализованный язык образования фреймов (ФЯОФ)

ФЯОФ предназначен для создания информационных фреймов запросов и директив и осуществления на их основе обмена сообщениями между разработчиком фреймов и ЭВМ. Для выполнения указанных функций ФЯОФ имеет словарь терминов, синтаксис построения текстов фреймов запросов, директив и ответов разработчика фреймов, транслятор для их преобразования с естественного языка во внутренние структуры данных и управляющую программу построения в диалоговом режиме фреймов общения.

Словарь терминов включает в себя списки слов и выражений, используемых для описания наименований директив, условий задачи, множества решений. В словаре имеются термины и выражения, общие для проблемы образовываемых фреймов, и списки слов, характерных для каждого уровня. Например, словарь для диалогового проектирования принципиальной схемы процесса содержит названия методов и видов кузнечно-заготовительной, механической и термической обработки, названия видов заготовок и др. В словаре второго уровня, используемого для проектирования технологических маршрутов, имеются названия операций, оборудования, схем установок, базовых поверхностей и т. д.

Синтаксис ФЯОФ определяет правила построения текстов фреймов общения, заданных отображениями

$$B : \langle ИД,\ ОБ,\ ТО \rangle \rightarrow R_{\text{к}},$$
$$ДР : \langle B,\ ИД,\ ТО,\ W \rangle \rightarrow R_{\text{в}}. \qquad (4)$$

Исходные данные в фреймах общения задаются в виде совокупности пар, состоящих из наименований характеристик *НХ* и их значений *ЗН:*

$$ИД = \{(НХ,\ ЗН)_i\}, \quad i = 1, 2, \ldots, \sim n.$$

Обращение ЭВМ к разработчику фреймов в фреймах запросов принято в повелительном наклонении, например, «выберите вид заготовки» или в форме вопроса «старение детали необходимо?» Технические ограничения описывают множество возможных альтернатив, из которых необходимо выбрать рациональное решение $TO=\{R_1\ R_2, \ldots, R_k\}$. Для некоторых задач на стадии алгоритмизации их сформулировать не удается. Разработчик фреймов в процессе решения конкретной задачи выявляет ограничения, а затем на основе интегральных эвристических оценок выбирает решение.

Ответы разработчика фреймов должны быть простыми: «да», «нет», номер решения, выбранный из заданного множества, номера базовых

поверхностей, название методов обработки, операций и т. д. Последовательность разделов в формулировках задач соответствует приведенному в выражениях (4). В целом формулировка задачи должна быть четкой, понятной разработчику фреймов и требующей от него простых и однозначных ответов.

Транслятор ФЯОФ осуществляет синтаксический и семантический контроль текстов фреймов директив и ответов разработчика фреймов, а затем производит их преобразование во внутренний язык системы. Например, если разработчиком фреймов выбран вид или метод получения заготовки, которых нет в словаре, то об этом сигнализируется фразой: «Этот вид (метод получения) заготовки программ неизвестен». При этом на экране отображается множество допустимых видов и методов получения заготовок, из которых необходимо произвести выбор. В тех случаях, когда ответ разработчика фреймов построен не по правилам ФЯОФ, на экране отображается выражение: «Уточните ответ» и приводится правильная его структура.

Выполнение директив может производиться несколькими способами: обращением к программному фрейму, реализующему задачу, сформулированную в директиве; выбором ветви вычислительного процесса (последовательности фреймов) в зависимости от условий задачи; генерированием вычислительного процесса на основе библиотеки программных модулей и фреймовой модели предметной области, охватывающих решение различных задач определенного типа, например выбор станков, технологических баз, приспособлений и др.

С помощью директив первого типа решаются простые задачи, например выбор элемента СПИД, упорядочение переходов но заданному признаку, расчет элемента в режимах резания и др. Директивы второго и третьего типов позволяют задавать ЭВМ решение более сложных задач, например спроектировать инструментальную наладку к револьверному станку. В результате выполнения директивы будут выбраны режущий инструмент, державки, определены размерная настройка инструмента, режимы резания, машинное и вспомогательное время.

Реализация сложных директив может осуществляться в диалоговом режиме под управлением ЭВМ и включать в себя как вычислительныефреймы, так и фреймы запросы. Возможность формирования директив различной сложности обогащает диалог и придает процессам образования фреймов большую гибкость и естественность.

Фрейм директива разрабатывается разработчиком фреймов. Для этого ему необходимо хорошо знать синтаксис и словарь ФЯОФ. Программы

транслятора, производя синтаксический и семантический контроль, выявляют возможные ошибки, которые разработчик фреймов должен устранить. В результате процесс построения директивы носит итерационный характер. Увеличивается время построения, а следовательно, и общее время решения задачи.

Наличие синтаксиса и словаря терминов позволяет создать программу, управляющую процессом формирования директивы и обеспечивающую ее правильность. Основу этой программы составляют фреймы запросы ЭВМ к разработчику фреймов по каждому разделу директивы (рис. 14.6).

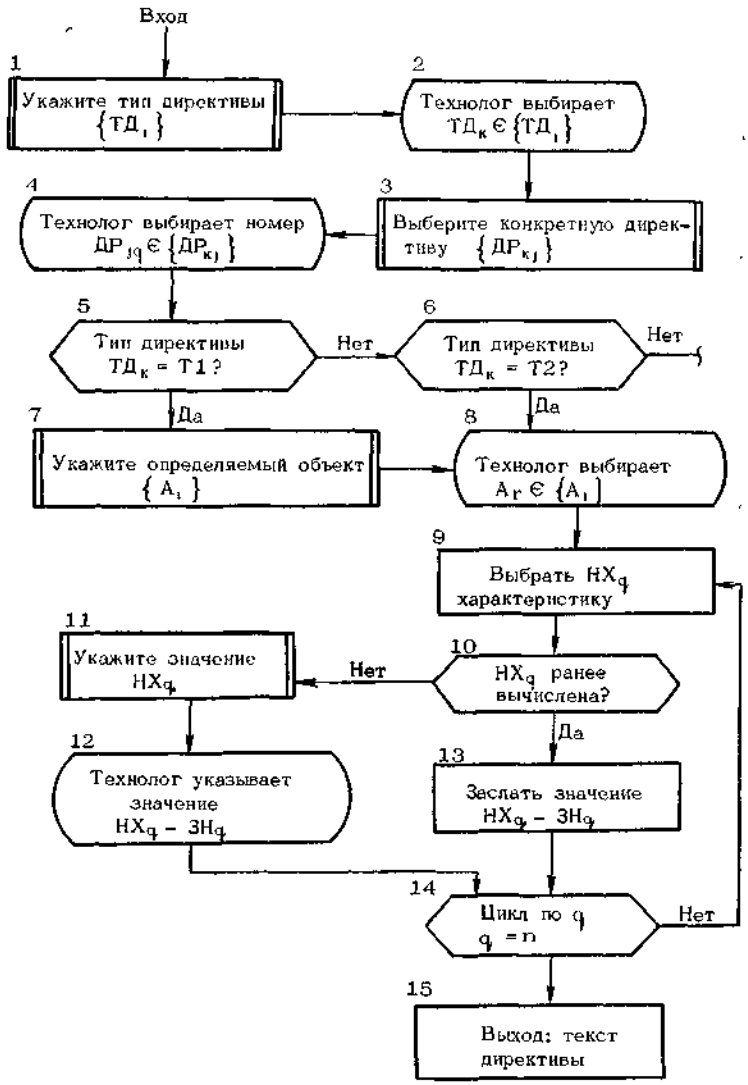


Рис. 14.6. Пример алгоритм диалогового построения текста директивы

Блоками *1, 2* на экране отображаются возможные типы директив, и технолог выбирает одну из них. Аналогичным образом блоками *3, 4* производится выбор конкретной директивы в пределах выбранного

типа. Логические блоки *5, 6* осуществляют выбор ветви формирования текста директивы в зависимости от ее типа (поиск, определение объектов, печать, черчение и др.). Для каждого типа директивы и вида определяемого объекта *A* заранее (на стадии алгоритмизации) устанавливается набор исходных данных и технических ограничений. Это дает возможность построить алгоритм формирования постановки задачи.

Оператором *9* выбирается очередная характеристика HX_q. Если ее значение ранее вычислено, то оператор *13* записывает на экране название и значение характеристики. В противном случае значение характеристики заполняется технологом (блоки *11, 12).* Оператором цикла *14* указанные действия повторяются для всей совокупности исходных данных. В результате процесс построения директивы приобретает целенаправленный характер, уменьшается количество ошибок, упрощаются алгоритмы синтаксического и семантического контроля директивы.

Возможны различные модификации алгоритма. Одна из них заключается в том, что вместо последовательного формирования состава и значений характеристик на экран отображается весь их набор, а разработчик фреймов должен присвоить им соответствующие значения. Первая модификация алгоритма применяется тогда, когда имеется логическая зависимость вида последующей характеристики от значения предыдущей, вторая — когда такая зависимость отсутствует, а задача характеризуется набором независимых друг от друга исходных данных и технических ограничений.

В заключение отметим, что в системах диалогового образования фреймов творческие способности разработчика фреймов дополняются большой памятью и быстродействием ЭВМ. В связи с этим те задачи, которые не могут быть решены разработчиком фреймов в условиях недостатка времени и программным способом из-за трудностей формализации, успешно решаются в диалоговых системах. Поэтому значительно расширяются возможности автоматизированного образования фреймов технологических процессов обработки деталей сложной формы, повышаются производительность труда и качество проектных решений.

15. Математическое моделирование при автоматизированном образовании фреймов

15.1. Иерахическая система математического моделирования

Содержание, качество и эффективность автоматизированного образования фреймов в значительной мере определяются свойствами математического обеспечения САОФ, включающего методы, математические модели и алгоритмы решения задач образования фреймов. Свойства математического обеспечения тесно взаимосвязаны со свойствами информационного и программного обеспечения. Основными принципами создания математического, информационного и программного обеспечения САОФ являются блочная структура построения и модульный принцип программирования. Эти принципы соблюдаются при разработке математического, информационного и программного обеспечения на основе единой системы математического моделирования, обеспечивающей построение взаимосвязанных моделей, соответствующих разнородным объектам, разному уровню знаний и различной полноте представления данных об этих объектах. К таким системам математического моделирования относится ИСТРА — иерархическая система математического моделирования объектных фреймов на различных уровнях абстрагирования.

В системе ИСТРА любой объект — конструкция изделия или оснастки, технологический процесс или процесс образования фреймов — моделируется одинаковыми способами. Рассматриваемый на любом уровне абстрагирования объект *A* с математической точки зрения имеет один и тот же прообраз **A**, адекватный реальному объекту; при этом *A* содержит лишь некоторую часть данных об **A**. В моделях различаются элементы двух типов — собственно объект *A* и его элементы, и контуры этого объекта и его элементов. Понятие контура является экспликацией таких понятий, как свойство, признак, характеристика, параметр и т. п.

Абстрагирование при моделировании объектных фреймов осуществляется по двум направлениям — по глубине структурирования и по степени абстрагирования элементов и контуров объекта, а также отношений между ними. По глубине структурирования сложный обьект *A* рассматривается либо как

неструктурированный объект *A,* представлющий собой единое целое, либо как система взаимосвязанных элементов одного уровня, либо как многоуровневая иерархическая система. По степени абстрагирования объект моделируется на уровнях структурных (методами теории множеств и теории графов), логических (методами математической логики) и количественных свойств и отношений. На каждом из этих основных уровней возможны описания объекта с различной степенью полноты и обобщения, соответствующие разным уровням абстрагирования структурных, логических и количественных свойств и отношений. Теоретической базой для создания системы ИСТРА служат математические исследования взаимосвязи между теоретико-множественным и логическим представлением объектов, логическим и алгебраическим представлением и т. д. Переход от одного к другому уровню описания осуществляется регламентированными способами с указанием границ и условий перехода одних величин в другие. Переход от количественной величины a_i^N к логической величине a_i^L определяется отношением вида

$$a_i^L = \begin{cases} 1, \text{ если } a_i^N R_i^N \Phi_i^N; \\ 0 \text{ — в противном случае.} \end{cases}$$

Здесь Φ_i^N — величина или функция, определяющая предельные значения a_i^N, и R_i^N —количественное отношение вида $\{ =, \neq, >, <, \leq, \geq\}$; возможны и менее четкие отношения вида{≈, «равно с точностью до целого числа» и т. п.}.

Обратный переход от a_i^L к a_i^N определяется отношением вида

$$a_i^L = 1 \rightarrow a_i^N R_i^N \Phi_i^N.$$

Переход от количественной величины a_i^N к элементу a_i^S множества *A* определяется отношением вида

$$a_i^S = \begin{cases} r_i^S \in A, \text{ если } a_i^N R_i^N \Phi_i^N; \\ a_i^S \not\ni A \text{ — в противном случае,} \end{cases}$$

а обратный переход — отношением вида

$$a_i^S \in A \rightarrow a_i^N R_i^N \Phi_i^N.$$

Переход от логической величины a_i^N к элементу $a_i^S \in A$ и обратно определяется отношениями вида

$$a_i^L = 1 \leftrightarrow a_i^S \in A;$$

$$a_i^L = 0 \leftrightarrow a_i^S \not\subset A.$$

Примером описания взаимосвязи структурных, логических и количественных величин является представление контура F_i объекта в виде логической переменной

$$F_i = \begin{cases} 1, \text{ если } \forall m_j \in M_i\,(\omega_j \subseteq \Delta_j); \\ 0, \text{ если } \exists m_j \in M_i(\omega_j \setminus \Delta_j \neq \varnothing), \end{cases}$$

где M_i — множество параметров контура F_i; ω_j — поле рассеяния погрешностей параметра m_j; Δ_j- —поле допуска на погрешности параметра m_j.

При переходе на более высокий уровень абстрагирования осуществляется свертка данных о моделируемом объекте, а при переходе к более детальному уровню описания — развертка этих данных. Система ИСТРА обеспечивает возможность представления в одной математической модели разнородных, в семантическом смысле, объектов при переходе к более абстрактным уровням описания, так как некоторые свойства и отношения на более абстрактном уровне оказываются изоморфными. Эта особенность системы ИСТРА и позволяет создавать по блочному принципу системы математического и информационного обеспечения САОФ, органически взаимосвязанные друг с другом и с модульной системой программного обеспечения (рис. 15.1).

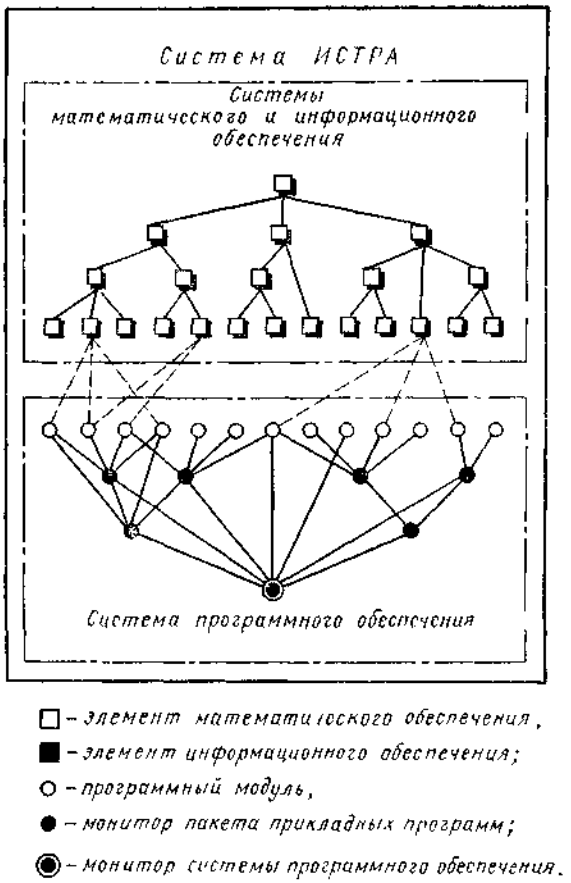


Рис. 15.1. Схема структуры математического, информационного и программного обеспечения САОФ в системе ИСТРА

На структурном уровне моделируются состав элементов объекта в виде неупорядоченного множества $A = \{a_1, a_2, ..., a_n\}$ или упорядоченного множества $A = (a_1, a_2, ..., a_n)$, состав контуров $F(A) = (F_1, F_2, ..., F_m)$ объекта и составы $F(a_i) \subseteq F(A)$ контуров элементов этого объекта, а также структурные отношения между элементами и контурами. К структурным относятся бинарные отношения между элементами и контурами объекта: отношения иерархической подчиненности, отношения порядка, смежности, сопряженности, функциональной связи и т. п. Структурные отношения образуют множество R^s, включающее подмножества $R^s(A)$ отношений между элементами A,

подмножества R^s *(F (A))* и R^s *(F* (a_i)) отношений между контурами и подмножества R^s *(A, F (A))* отношений между элементами и контурами. Например, бинарными отношениями $R^s{}_k \in R^s$ *(F* (a_i)) между контурами F_i, F_j—геометрическими поверхностями детали будут следующие их свойства:

поверхности F_i, F_j, смежны;

поверхности F_i, F_j соединены размером $l_{i,j}$;

поверхность F_i является измерительной базой для F_j;

поверхность F_i является черновой поверхностью для получения F_j при механической обработке и т. п.

Бинарные отношения между контурами элемента $a_1 \in A$ представляются как подмножества $[F\ (a_i) \times F\ (a_i)]$ декартова произведения $F\ (a_i) \times F\ (a_i)$ или в виде булевой матрицы

$$\|c_{i\,(j)}\|_{F(a_i)} = [F\,(a_i) \times F\,(a_i)] = \begin{array}{c} \begin{array}{cccc} F_1 & F_2 & \ldots & F_m \end{array} \\ \begin{bmatrix} c_{1\,(1)} & c_{1\,(2)} & \ldots & c_{1\,(m)} \\ c_{2\,(1)} & c_{2\,(2)} & \ldots & c_{2\,(m)} \\ \ldots & \ldots & \ldots & \ldots \\ c_{m\,(1)} & c_{m\,(2)} & \ldots & c_{m\,(m)} \end{bmatrix} \end{array} \begin{array}{c} F_1 \\ F_2 \\ \ldots \\ F_m, \end{array} \tag{15.1}$$

где $c_{i(j)} = 1$, если бинарное отношение между F_i, F_j существует и $c_{i(j)} = 0$ в противном случае. Матрицу (15.1) можно рассматривать как матрицу смежности вершин графа $G = (F\ (a_i), C)$, дуги $c_{i(j)} = 1$ которого эквивалентны элементам $c_{i(j)} = 1$ матрицы (15.1). Например, на рис. 15.2 показаны графы, соответствующие матрицам бинарных отношений смежности и размерных связей между поверхностями детали.

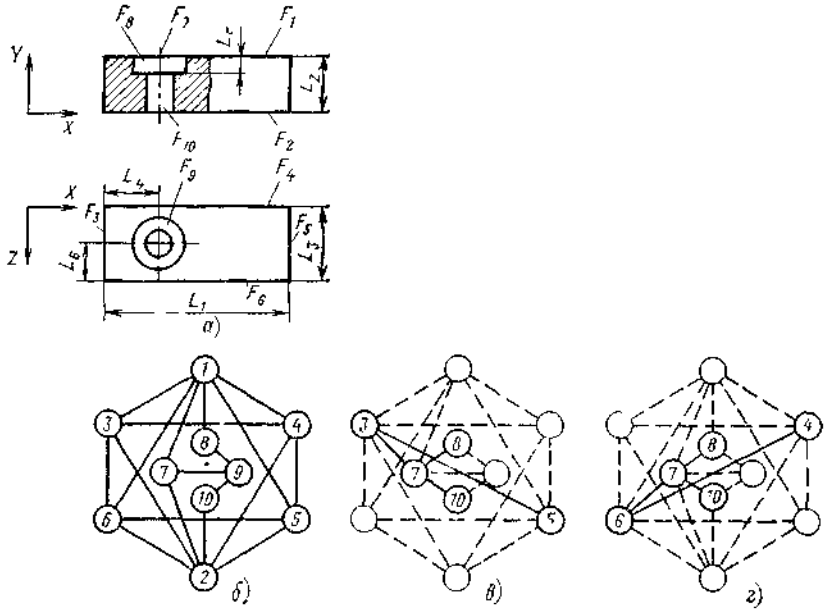


Рис. 15.2. Примеры описания бинарных отношений между поверхностями детали.
а — деталь, б — граф смежности поверхностей детали, *в* — граф размеров, параллельных *OX, г* — граф размеров, параллельных *OZ*

Бинарные отношения $R^S_k \in R^s$ *(F (A))* между контурами сложного объекта описываются аналогичным образом в виде булевой матрицы

$$\| c_{i\,(j)} \|_{F\,(A)} = [F\,(A) \times F\,(A)] \qquad (15.2)$$

или в виде графа *G = (F (A), C).*

Бинарные отношения $R^S_k \in R^s$ *(A, F (A))* описывают составы контуров элементов $a_i \in A$: составы контуров деталей в группе деталей, изготавливаемых по одному групповому технологическому процессу, в совокупности деталей, связанных с определенными видами обработки, и т. п. Эти отношения описываются в виде булевых матриц

$$\| c_{i\,(j)} \|_{A,\,F\,(A)} = [A \times F\,(A)], \qquad (15.3)$$

где $c_{i(j)} = 1$, если F_j входит в состав $F(a_i)$ контуров элемента a_i. Так, для группы деталей (рис. 15.3) матрица контуров (15.3) имеет вид

$$[A \times F(A)] = \begin{array}{c} \begin{array}{ccccccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 & F_8 & F_9 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{array} \qquad (15.4)$$

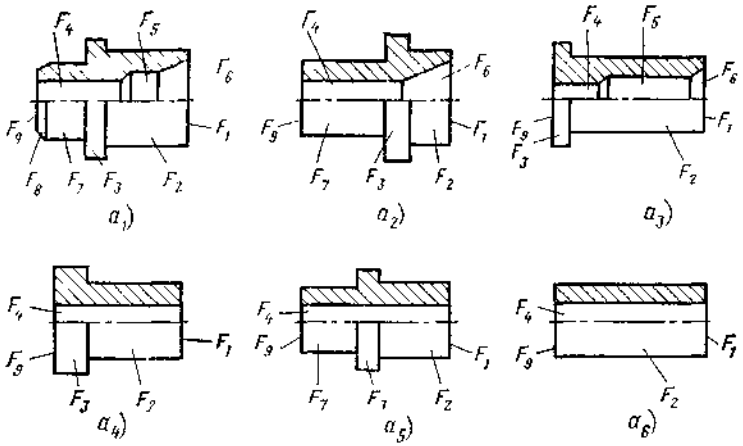


Рис. 15.3. Группа деталей.

Бинарные отношения $R^S_k \in R^s(A)$ между элементами A представляются как подмножества декартова произведения $A \times A$ или в виде булевой матрицы

$$\| c_{i(j)} \|_A = [A \times A], \qquad (15..4)$$

где $c_{i(j)} = 1$, если бинарное отношение между $a_i, a_j \in A$ существует, и $c_{i(j)} = 0$ в противном случае. Матрица (15.4) рассматривается как матрица смежности вершин графа $G = (A, C)$, дуги $c_{i(j)} = 1$ которого эквивалентны элементам $c_{i(j)} = 1$ матрицы $\|c_{i(j)}\|_A$. Каждая i-я строка (столбец) матрицы (15.4) рассматривается как представление бинарного отношения $R^s_k(a_i)$ между a_i и элементами подмножества

$A_i \subseteq A$, $A_i = (a_{i1}, a_{i2}, \ldots, a_{im})$, для которых элементы строки (столбца) матрицы равны единице. Если количество множеств более двух, то бинарные отношения между их элементами описываются в виде блочной матрицы

$$\|c_{i(j)}\|_A = \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & \ldots & A_k \end{array} \\ \left[\begin{array}{c|c|c|c} [A_1 \times A_1] & [A_1 \times A_2] & \ldots & [A_1 \times A_k] \\ \hline [A_2 \times A_1] & [A_2 \times A_2] & \ldots & [A_2 \times A_k] \\ \hline \ldots & \ldots & \ldots & \ldots \\ \hline [A_k \times A_1] & [A_k \times A_2] & \ldots & [A_k \times A_k] \end{array}\right] \end{array} \begin{array}{c} \\ A_1 \\ A_2 \\ \ldots \\ A_k \end{array} \qquad (15.5)$$

Иерархические отношения описываются графами-деревьями иерархической структуры. Например, на рис. 15.4 показаны структуры геометрических контуров монолитного и сборного вариантов конструкции вала с конической шестерней.

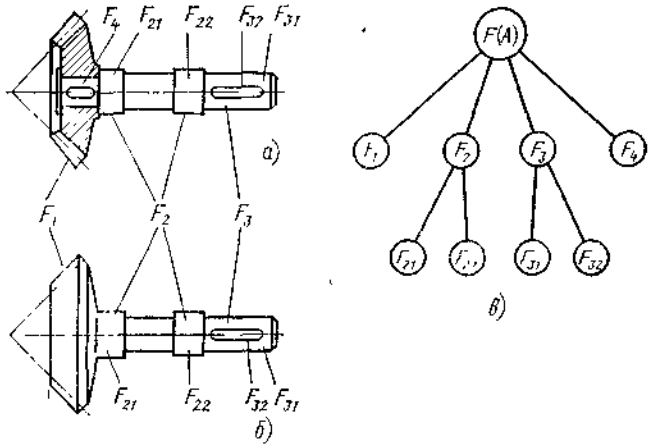


Рис. 15.4. Варианты конструкции вала с шестерней:
а — сборный вал с шестерней; *б* — монолитный вал с шестерней; *в* — иерархическая структура геометрических контуров вала с шестерней

Как видно из рисунка, монолитный и сборный варианты конструкции могут быть описаны подмножествами одного набора контуров, включающего описание: всех свойств при различном конструктивном исполнении этих объектов.

Множество *C* элементов булевой матрицы или дуг графа рассматривается как объект с составом контуров *F (C)*. Между элементами C и контурами *F (C)* также возможны бинарные отношения, которые группируются в подмножества $R^S(C)$, $R^S(c_i)$, $R^S(F(C))$, $R^S(F_j(C))$, $R^S(C, F(C))$, $R^S(F(c_i))$, $R^S(F_j(c_i))$, входящие в R^s.

На логическом уровне каждому множеству, булевой матрице или графу соответствуют наборы логических отношений $R_k^L \in R^L$ между входящими в них элементами, представленными в виде логических переменных. Множествам *A* и *F (A)* соответствуют: логические отношения вида $R_k^L(A)$

$$A = R_k^L(a_1, a_2, \ldots a_n) \qquad (15.6)$$

между объектом *A* и его элементами;

отношения вида $R_k^L(a_i)$

$$a_i = R_k^L(a_{i_1}, a_{i_2}, \ldots, a_{i_{n-1}}) \qquad (15.7)$$

между $a_i \in A$ и другими элементами *A;*

отношения вида $R_k^L(F(A))$ и $R_k^L(F_j(A))$ между контурами объекта

$$F(A) = R_k^L(F_1, F_2, \ldots, F_m); \qquad (15.8)$$

$$F_j(A) = R_k^L(F_{j_1}, F_{j_2}, \ldots, F_{j_{m-1}}); \qquad (15.9)$$

отношения вида $R_k^L(F(a_i))$ и $R_k^L(F_j(a_i))$

$$F(a_i) = R_k^L(F_1(a_i), F_2(a_i), \ldots, F_m(a_i)); \qquad (15.10)$$

$$F_j(a_i) = R_k^L(F_{j_1}(a_i), F_j(a_i), \ldots, F_{j_{m-1}}(a_i)) \qquad (15.11)$$

между контурами элемента $a_i \in A$.

Матрицам бинарных отношений соответствуют логические отношения между множествами элементов строк и столбцов этих матриц, образующие подмножества $R^L(C)$, $R^L(c_i)$, $R^L(F(C))$, $R^L(F_j(C))$, $R^L(F(c_i))$ и $R^L(F_j(c_i))$. Каждой строке или столбцу булевой матрицы соответствует определенное подмножество элементов множества, входящего в декартово произведение; этим подмножествам соответствуют логические отношения, приводимые к виду (15.6)—(15.11). Для столбцов матрицы (15.3) эти отношения имеют специфический вид

$$F_j(A) = R_k^L(F_j(a_1),\ F_j(a_2),\ \ldots,\ F_j(a_n)). \quad (15.12)$$

На количественном уровне каждому элементу множества, булевой матрицы или логической переменной ставится в соответствие алгебраическая или другая количественная переменная, либо — при конкретных вычислениях — числовая величина, а логические отношения переходят в количественные отношения — уравнения, неравенства и т. п. Логические отношения из R^L $(F\ (A))$ и R^L $(F_j\ (A))$ переходят в количественные отношения вида

$$F(A) = R_k^N(F_1,\ F_2,\ \ldots,\ F_m);$$

$$F_j(A) = R_k^N(F_{j_1},\ F_{j_2},\ \ldots,\ F_{j_{m-1}}),$$

а из R^L $(F\ (a_i))$ и R^L $(F_j\ (a_i))$ — в отношения вида

$$F(a_i) = R_k^N(F_1,\ F_2,\ \ldots,\ F_m);$$

$$F_j(a_i) = R_k^N(F_{j_1},\ F_{j_2},\ \ldots,\ F_{j_{m-1}}).$$

Все другие количественные отношения между элементами и контурами объекта аналогичны этим четырем видам отношений. Взаимосвязь структурных, логических и количественных свойств и отношений при моделировании объекта в системе ИСТРА показана на рис. 15.5.

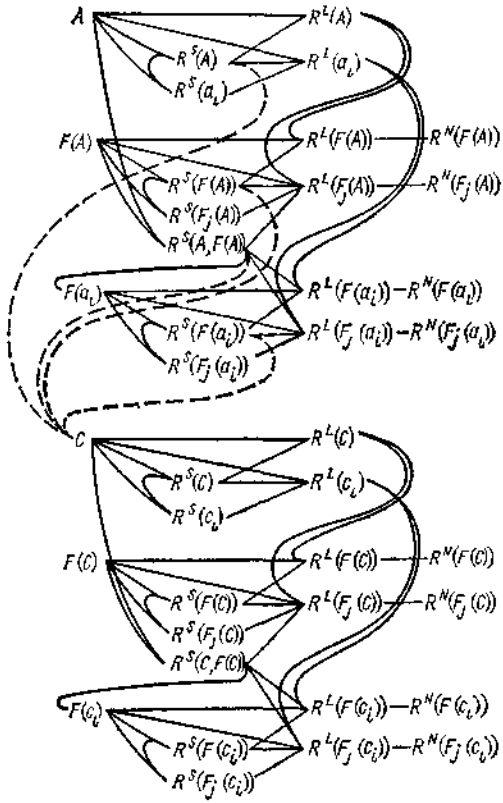


Рис. 15.5. Взаимосвязь структурных, логических и количественных свойств и отношений в системе ИСТРА

Взаимосвязь описаний объектов на уровнях структурных, логических и количественных свойств и отношений принимает конкретные формы в условиях определенного объекта, опреде ленной задачи образования фрейма и т. д. Рассмотрим, например, задачу образования фрейма по вычислению скорости резания при обработке деталей на металлорежущих станках. Известно, что множество факторов, влияющих на скорость резания при различных видах обработки, можно представить в виде

$$M_v = \{c_v,\ k_v,\ T_н,\ m,\ t_{усл},\ s,\ d,\ B,\ x_v,\ y_v,\ z_v,\ r_v\}, \quad (15.13)$$

где c_v — коэффициент, характеризующий нормативные условия обработки; k_v—общий поправочный коэффициент на скорость резания,

учитывающий измененные условия обработки против нормативных; d —диаметр обработки или диаметр инструмента, мм; $T_и$ —стойкость инструмента, мин; m — показатель относительной стойкости инструмента; $t_{усл}$ — условная глубина резания, мм; s — подача, мм/об (мм/зуб или мм/дв. ход, или мм/мин); r —число зубьев режущего инструмента; B —ширина фрезерования или шлифования, мм; x_v, y_v, z_v, r_v — показатели степени.

Анализ причинно-следственных связей между факторами (15.13) показывает, что логические отношения между ними и скоростью резания v имеют вид

$$v = c_v \wedge k_v \wedge T_и \wedge m \wedge$$
$$\wedge [(t_{усл} \wedge x_v) \vee (s \wedge y_v) \vee (d \wedge z_v) \vee (B \wedge r_v)], \quad (15.14)$$

причем c_v, k_v, $T_и$ и m всегда истинно, а истинностные значения других переменных зависят от вида обработки. Например, при продольном наружном точении истинно $t_{усл}$, x_v, s и y_v, а при сверлении — s, y_v, d и z_v и т. д.

Поскольку обобщенная формула количественных отношений между факторами (15.13) имеет вид

$$v = \frac{c_v k_v d^{z_v}}{T_и^{m} t_{усл}^{x_v} s^{y_v} B^{r_v}}, \quad (15.15)$$

то формулы количественных отношений между факторами, с учетом их истинностных логических значений, имеют вид:

при продольном наружном точении

$$v = \frac{c_v k_v}{T_и^{m} t^{x_v} s^{y_v}}; \quad (15.16)$$

при сверлении

$$v = \frac{c_v k_v d^{z_v}}{T_и^{m} s^{y_v}} \quad (15.17)$$

и т. д. Таким образом, формулы (15.13)—(15.17) представляют модели расчета скорости резания на различных уровнях абстрагирования.

15.2. Образование типовых структурно-логических фреймов моделей объектов

При автоматизированном образования объектных фреймов вначале формируется порождающая среда, в которой будет осуществляться синтез этогофрейма. Порождающая среда включает данные о предметной области, к которой относится объект, о существующих и разрабатываемых структурах объекта, известные или прогнозируемые закономерности и связи между элементами и свойствами объекта и внешней среды. На базе этих данных создается математическая модель, охватывающая те данные порождающей среды, которые могут быть формализованы и представлены средствами системы ИСТРА. Одна модель $S_i(A)$ порождающей среды содержит в себе данные о множестве $A = \{A_1, A_2, ..., A_N\}$ объектов. Построение такой модели основано на свойствах эквивалентности элементов $a_i, a_j \in A$, входящих в различные объекты. Отношения эквивалентности устанавливаются на различных уровнях абстрагирования.

На структурном уровне, если учитываются только составы контуров $a_i, a_j \in A$, элемент a_i эквивалентен a_j (обозначается $a_i \equiv a_j$ при условии

$$F(a_i) = F(a_j);$$

если учитываются и бинарные отношения между контурами, от $a_i \equiv a_j$ при условии

$$\{F(a_i) = F(a_j), \quad R_k^S(F(a_i)) \equiv R_k^S(F(a_j))\}, \quad (15.18)$$

т. е. когда эквивалентны матрицы заданного бинарного отношения R_k^S. Например, детали (рис. 15.6) эквивалентны по составу обрабатываемых поверхностей, но не эквивалентны по условию (15.18), так как матрицы смежности обрабатываемых поверхностей не эквивалентны.

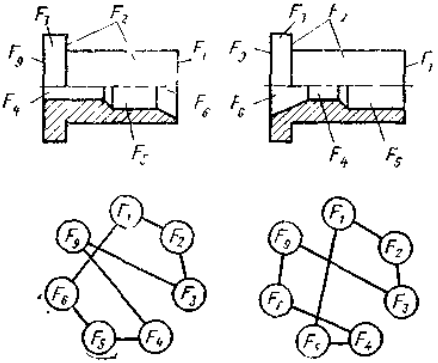


Рис. 15.6. Эквивалентность деталей на структурном уровне

Если учитываются бинарные отношения между самими элементами *A*, то $a_i \equiv a_j$ при условии

$$R_k^S(a_i) = R_k^S(a_j). \tag{15.19}$$

На логическом уровне, если учитываются только отношения (15.7), $a_i \equiv a_j$ при условии

$$R_k^L(a_i) \equiv R_k^L(a_j). \tag{15.20}$$

Если учитываются логические отношения (15.10) между контурами элементов, то $a_i \equiv a_j$ при условии

$$R_k^L(F(a_i)) \equiv R_k^L(F(a_j)), \tag{15.21}$$

а если учитывается только отношение (15.11) между F_i и другими контурами элемента, то $a_i \equiv a_j$ при условии

$$R_k^L(F_l(a_i)) \equiv R_k^L(F_l(a_j)). \tag{15.22}$$

На количественном уровне элементы a_i и a_j эквивалентны при условии

$$R_k^N(F(a_i)) = R_k^N(F(a_j)), \tag{15.23}$$

а если учитываются количественные отношения только для некоторого контура F_i, то $a_i \equiv a_j$ при условии

$$R_k^N(F_l(a_i)) = R_k^N(F_l(a_j)). \tag{15.24}$$

Если $a_i \equiv a_j$ и численные значения количественных величин в R_k^N одинаковы, то $a_i = a_j$.

Условия эквивалентности элементов объекта вида (15.18)— (15.24) могут быть заданы как по отдельным структурным, логическим или количественным отношениям, так и по любым наборам R^s, R^L и R^N. Эквивалентность элементом объекта *A* может быть установлена и на основании анализа свойств дуг в графе $G = (A, C)$.

Структура модели порождающей среды и логические отношения между элементами и контурами модели и объектов могут иметь любой вид. Однако для повышения эффективности автоматизированного образования фреймов целесообразно применять типовые фреймы моделей определенного вида в зависимости от характера решаемых задач и требуемого уровня унификации решений. Все фреймы моделей порождающей среды разделяются на сочетательные и упорядочивающие. Сочетательные модели $S^0_i(A)$ применяются в тех случаях, когда определяется только состав элементов объекта, а упорядочивающие модели $S_i(A)$ — когда определяются и состав, и структурные отношения между элементами объекта. Каждая модель содержит состав А элементов, входящих в объекты A_k, матрицу контуров вида (15.3), структурные, логические и количественные

отношения между элементами и контурами A, определяющие состав и все свойства объектов A_K.

Свойства самих моделей S_i, влияющие на содержание процесса образования фреймов, характеризуются набором контуров $F(S) = \{F_G, F_n, F_\lambda, F_a\}$. Контур $F_G = 1$, если все графы $G_k = (A_k, C_k)$ объектов A_k, проектируемых по данной модели, — суть простые элементарные пути или цепи, и $F_G = 0$ в противном случае; $F_n = 1$, если число элементов во всех вариантах A_k одинаково; $F_\lambda = 1$, если отношение порядка между любыми $a_i, a_j \in A_k$ во всех вариантах A_k одинаково; $F_a = 1$, если состав элементов во всех вариантах A_k одинаков.

Классы сочетательных моделей $S^0_i(A)$ зависят от состава контуров $F(S^0) = \{F_n, F_a\}$ и определяются строками булевой матрицы

$$[S^0 \times F(S^0)] = \begin{array}{c} \begin{array}{cc} F_n & F_a \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} S^0_1 \\ S^0_2 \\ S^0_3 \end{array} \qquad (15.25)$$

По модели $S^0_1(A)$ можно выбрать единственный объект A_k с заданным набором контуров $F(A_k)$; модель $S^0_2(A)$ может содержать в себе различные варианты A_k с одинаковым количеством, но разным составом элементов. Модель $S^0_3(A)$ может содержать варианты A_k, различающиеся и количеством и составом входящих элементов.

Классы упорядочивающих моделей $S_i(A)$ определяются строками булевой матрицы

$$[S \times F(S)] = \begin{array}{c} \begin{array}{cccc} F_G & F_n & F_\lambda & F' \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \\ S_{11} \\ S_{12} \end{array} . \qquad (15.26)$$

Для поиска стандартных или образованных фреймов используются модели класса S_1, остальные модели используются для получения унифицированных и индивидуальных фреймов при наличии их

вариантов и необходимости оптимизации решения. Модели класса S_2 используются для синтеза новыхфреймов, различающихся только составом элементов; модели S_3, S_7 и S_9—для синтеза фреймов, различающихся только структурой связей между элементами, а остальные модели —для синтеза фреймов, различающихся и составом элементов, и структурой связей между ними.

Модели класса S_1 называются табличными. Структура табличной модели описывается матрицей контуров вида (15.3), строки которой упорядочены в соответствии с отношением порядка между элементами $a_i, a_j \in A$. Граф $G = (A, C)$ связей элементов при образовании фрейма не используется.

Модели классов S_2, S_5, S_7, S_8 и S_{11} называются сетевыми. Структура сетевой модели описывается ориентированным графом $G = (A, C)$, не имеющим ориентированных циклов. В такой модели может содержаться несколько вариантов образовываемго фрейма объекта A_k, но во всех вариантах A_k сохраняется неизменным отношение порядка между входящими элементами. Варианты фреймов A_k, обазовыаемые по модели S_2, различаются только составом элементов, а по модели S_5 — и количеством, и их составом. Варианты фреймов A_k, обазовыаемые по модели S_7, различаются только структурой связей, т. е. составом дуг в графе $G_k = (A_k, C_k)$. Варианты, фреймов , обазовыаемые по моделям S_8 и S_{11}, различаются структурой связей, а также составом — в модели S_8 или количеством и составом элементов — в модели S_{11}.

Модели классов S_3, S_4, S_6, S_9, S_{10} и S_{12} называются перестановочными. Структура перестановочной модели описывается либо графом $G= (A, C)$, содержащим ориентированные циклы, либо наборами отношений вида $R_k^S (a_i)$ или $R_k^L (a_i)$, определяющих отношение порядка между элементами образовываемого фрейма объекта. Все варианты A_k, образовываемые по перестановочной модели, различаются отношениями порядка между входящими в них элементами; кроме того, в модели S_4 варианты A_k различаются составом, а в модели S_6 — количеством и составом элементов. Варианты A_k, образовываемые по моделям S_9, S_{10} и S_{12}, различаются и отношениями порядка, и структурой связей между элементами объекта, причем в модели S_{10} варианты $A_к$ различаются еще и по составу элементов, а в модели S_{12} — по количеству и их составу.

Для повышения эффективности автоматизированного образования фреймов логические отношения между элементами и контурами модели порождающей среды и образовываемых фреймов объектов приводятся к регламентированному виду. Так, зависимость между составом элементов образовываемого фреймов объекта и

порождающей среды выражается отношением (15.6) в следующем виде:

если A_k должен содержать фиксированный набор элементов $a_i \in A$, то

$$A_k = a_{k_1} \wedge a_{k_2} \wedge \ldots \wedge a_{k_m} = \bigwedge_{i=1}^{m} a_{k_i}; \quad (15.27)$$

если A_k может содержать любой элемент $a_i \in A$, то

$$A_k = a_1 \vee a_2 \vee \ldots \vee a_n = \bigvee_{i=1}^{n} a_i, \quad (15.28)$$

а если только какой-либо единственный элемент $a_i \in A$, то

$$A_k = a_1 \nabla a_2 \nabla \ldots \nabla a_n = \nabla_{i=1}^{n} a_i, \quad (15.29)$$

Если A_k может содержать любые наборы групп элементов вида (15.27), то

$$A_k = \bigvee_{j=1}^{N} \left(\bigwedge_{i=1}^{m} a_{k_i} \right)_j, \quad (15.30)$$

а если может содержать любые наборы групп элементов вида (15.29), то

$$A_k = \bigvee_{j=1}^{N} \left(\nabla_{i=1}^{m} a_{k_i} \right)_j. \quad (15.31),$$

Если в A_k должны входить наборы из N групп элементов вида (15.28), то

$$A_k = \bigwedge_{j=1}^{N} \left(\bigvee_{i=1}^{m} a_i \right)_j, \quad (15.32)$$

а если из N групп элементов вида (15.29), то

$$A_k = \bigwedge_{j=1}^{N} \left(\nabla_{i=1}^{m} a_i \right)_j; \quad (15.33)$$

если в A_k могут входить элементы групп обоих видов, то

$$A_k = \left[\bigwedge_{j'=1}^{N'} \left(\bigvee_{i'=1}^{m'} a_{i'} \right)_{j'} \right] \vee \left[\bigwedge_{j''=1}^{N''} \left(\nabla_{i''=1}^{m''} a_{i''} \right)_{j''} \right]. \quad (15.34)$$

Зависимости между составом контуров $F(A_k)$ и контурами порождающей модели выражаются отношением (15.8). Эти зависимости по составу сентенциональных связок и смысловому содержанию аналогичны отношениям (15.27)—(15.34):

$$F(A_k) = \bigwedge_{i=1}^{n} F_{k_i}; \quad (15.35)$$

$$F(A_k) = \bigvee_{i=1}^{m} F_i; \quad (15.36)$$

$$F(A_k) = \underset{i=1}{\overset{m}{\nabla}} F_i; \quad (15.37)$$

$$F(A_k) = \bigvee_{j=1}^{N} \left(\bigwedge_{i=1}^{n} F_{k_i} \right)_j; \quad (15.38)$$

$$F(A_k) = \bigvee_{j=1}^{N} \left(\underset{i=1}{\overset{n}{\nabla}} F_i \right)_j; \quad (15.39)$$

$$F(A_k) = \bigwedge_{j=1}^{N} \left(\bigvee_{i=1}^{n} F_i \right)_j; \quad (15.40)$$

$$F(A_k) = \bigwedge_{j=1}^{N} \left(\underset{i=1}{\overset{n}{\nabla}} F_i \right)_j; \quad (15.41)$$

$$F(A_k) = \left[\bigwedge_{j'=1}^{N'} \left(\bigvee_{i'=1}^{n'} F_{i'} \right)_{j'} \right] \vee \left[\bigwedge_{j''=1}^{N''} \left(\underset{i''=1}{\overset{n''}{\nabla}} F_{i''} \right)_{j''} \right]. \quad (15.42)$$

Аналогичный вид имеют отношения (15.10). Отношения (15.12) между F_j *(A)* и одноименными контурами F_j (a_i) элементов A по виду и смысловому содержанию также аналогичны отношениям (15.35)—(15.42):

$$F_j(A) = \bigwedge_{i=1}^{m} F_j(a_{k_i}); \quad (15.43)$$

$$F_j(A) = \bigvee_{i=1}^{n} F_j(a_i); \quad (15.44)$$

$$F_j(A) = \underset{i=1}{\overset{n}{\nabla}} F_j(a_i); \quad (15.45)$$

$$F_j(A) = \bigvee_{k=1}^{N} \left(\bigwedge_{i=1}^{m} F_j(a_i) \right)_k; \quad (15.46)$$

$$F_j(A) = \bigvee_{k=1}^{N} \left(\underset{i=1}{\overset{m}{\nabla}} F_j(a_i) \right)_k; \quad (15.47)$$

$$F_j(A) = \bigwedge_{k=1}^{N} \left(\bigvee_{i=1}^{m} F_j(a_i) \right)_k; \quad (15.48)$$

$$F_j(A) = \bigwedge_{k=1}^{N} \left(\underset{i=1}{\overset{m}{\nabla}} F_j(a_i) \right)_k; \quad (15.49)$$

$$F_j(A) = \left[\bigwedge_{k'=1}^{N'} \left(\bigvee_{i'=1}^{m'} F_j(a_{i'}) \right)_{k'} \right] \vee \left[\bigwedge_{k''=1}^{N''} \left(\underset{i''=1}{\overset{m''}{\nabla}} F_j(a_{i''}) \right)_{k''} \right]. \quad (15.50)$$

Таким образом, классификационными признаками типовых

структурно-логических моделей порождающей среды являются их классы S^0_i *или* S_i и логические отношения вида (15.27)—(15.50) между элементами и контурами модели и образовываемого фрейма объекта. Структурно-логические модели применяются при определении возможной структуры и структурной оптимизации образовываемого фрейма объекта. Числовые расчеты и параметрическая оптимизация осуществляется на уровне количественных свойств и отношений при известной структуре этого образовываемого фрейма объекта.

15.3. Математическое моделирование процессов образования фреймов

При математическом моделировании САОФ рассматривается в двух аспектах: как система образовывающая фреймы объектов, и как система операторов, характеризующих процессы образования фреймов. Оператор τ_k характеризует часть процесса образования фреймов T_i, при реализации которой образовываемый фрейм объекта приобретает какие-либо заданные свойства. Состав операторов T_i есть часть множества

$$T = \{\tau_1, \tau_2, \ldots \tau_R\}$$

операторов всей САОФ. Одному оператору может соответствовать этап процесса образования фрейма, группа операций образования фрейма, отдельная операция образования фрейма. Объекты САОФ образуют множество

$$\Pi = \{\pi_1, \pi_2, \ldots, \pi_Q\},$$

а полный состав элементов САОФ описывается множеством $P=T \bigcup \Pi$. Для установления отношений между свойствами объекта A и САОФ P при решении задач образования фрейма используется пространство контуров F, единое для всех элементов объекта и САОФ:

$$\left.\begin{array}{c} \mathbf{F} = \{F_1, F_2, \ldots, F_N\}; \\ \forall F(A_k),\ F(P)\,(F(A_k),\ F(P) \subseteq \mathbf{F}). \end{array}\right\} \quad (15.51)$$

Бинарные отношения между элементами САОФ P описываются булевыми матрицами $[P \times P]$, каждую из которых можно представить как блочную матрицу

$$[P \times P] = [(T,\ \Pi) \times (T,\ \Pi)] = \begin{array}{c} \begin{array}{cc} T & \Pi \end{array} \\ \left[\begin{array}{c|c} [T \times T] & [T \times \Pi] \\ \hline [\Pi \times T] & [\Pi \times \Pi] \end{array}\right] \end{array} \begin{array}{c} T \\ \\ \Pi \end{array}.$$

В этой матрице блоки [Т × Т] и [П × П] описывают бинарные отношения между элементами одного множества *Т* или П, например, взаимосвязь операторов по возможной очередности их реализации или наличие сопряжений между элементами конструкции приспособления. Матрица *[Т × Т]* может рассматриваться как матрица смежности вершин графа *G = (T, C),* матрица [П × П] — как матрица смежности вершин графа G = (П, С). Блоки *[Т* × П] и [П × *Т]* описывают бинарные отношения между элементами различных множеств: между $\tau_i \in T$ и $\pi_j \in \Pi$, или наоборот.

Составы контуров $F(\tau_k)$ операторов $\tau_k \in T$ описываются булевой матрицей, аналогичной (15.3):

$$\| c_{i\,(j)} \|_{T,\ F\,(T)} = [T \times F\,(T)]. \quad (15.52)$$

Бинарные отношения между контурами операторов описываются булевыми матрицами вида (15.2), например,

$$\| c_{i\,(j)} \|_{F\,(T)} = [F\,(T) \times F\,(T)].$$

При совместном описании объекта и САОФ они характеризуются одними и теми же контурами. Взаимосвязь между контурами *F (A)* и действительными свойствами САОФ описывается булевой матрицей

$$\| c_{i\,(j)} \|_{F\,(A),F\,(P)} = [F\,(A) \times F\,(P)],$$

где $c_{i(j)}$= 1, если контур F_j *(P)* влияет на реализацию контура F_i *(A)* в процессе образования фреймов.

Основными видами математических моделей в системе ИСТРА являются модели S (*A*) объектов и модели S *(P)* САОФ; последние могут распадаться на модели S*(T)* операторов, моделирующие содержание процессов образования фреймов, и модели S (П) объектов оснащения САОФ (модели технических средств САОФ). Эффективность автоматизированного образования фреймов зависит прежде всего от свойств структурно-логических моделей, поскольку они отражают такие свойства САОФ, как уровень унификации и типизации ее элементов, возможные варианты процессов образования фреймов, технических средств САОФ. Классы моделей *S (P), S (T)* и S (П) соответствуют булевым матрицам (15.25) и (15.26). Табличные, сетевые и перестановочные модели при образовании фреймов используются для получения решений различного уровня унификации.

Модель S_i *(T)* операторов образования фреймов может быть использована для процесса образования фрейма A_j только при выполнении условий

$$F(A_j) \subseteq F(T);$$

$$R^S(F(A_j)) \subseteq R^S(F(T));$$

$$R^L(F(T)) \vdash R^L(F(A_j));$$

$$R^N(F(T)) \rightarrow R^N(F(A_j)).$$

Свойства фрейма A_j могут по-разному объединяться в контуры F_i (A_j), вследствие чего состав контуров F (A_j) и отношения (15.48) между ними могут иметь различный вид. Поэтому, если в какой-либо группефреймов, процессы образования которых целесообразно осуществлять по одной модели S_i *(T),* логические отношения описываются различными функциями, то их необходимо привести к виду (15.35)—(15.42) путем перераспределения составов свойств, описываемых контурами *F (A).* Например, анализируя состав контуров деталей (см. рис. 15.3), можно заметить, что каждая из них имеет контуры F_1, F_2, F_4 и F_9, поэтому отношение вида (15.8) между множеством F *(A)* всех контуров деталей и отдельными контурами можно представить как

$$F(A) = (F_1 \wedge F_2 \wedge F_4 \wedge F_9) \vee F_3 \vee F_5 \vee F_6 \vee F_7 \vee F_8.$$

Однако, если не учитывать условие одновременного вхождения контуров F_1, F_2, F_4 и F_9 в состав контуров каждой детали, то получим

$$F(A) = F_1 \vee F_2 \vee F_3 \vee F_4 \vee F_5 \vee F_6 \vee F_7 \vee F_8 \vee F_9,$$

что соответствует отношению (15.36). Далее, если ввести контур более высокого уровня $F'_1 = \{ F_1, F_2, F_4, F_9\}$, то получим

$$F'(A) = F'_1 \vee F_3 \vee F_5 \vee F_6 \vee F_7 \vee F_8.$$

Наконец, если ввести контуры F''_1 = F (a_1), $F''_2 = F$ (a_2), ..., F''_6 =F (a_6), то получим

$$F''(A) = F''_1 \nabla F''_2 \nabla F''_3 \nabla F''_4 \nabla F''_5 \nabla F''_6,$$

что соответствует отношению (15.37). Таким образом, одна и та же совокупность фреймов может иметь, в зависимости от способа группирования их контуров, различные формы отношений вида (15.8) между F (A) и $F_i \in F$ *(A).*

Типовые отношения устанавливаются и для контуров САОФ. Если каждый оператор $\tau_k \in T$ может быть использован для реализации любого контура F_i либо совместно, либо отдельно от других

контуров, то R^L *(F* (T)) — отношение вида (15.8) — имеет вид, аналогичный (15.36), а если только для реализации какого-либо единственного контура, то (15.37) Если реализуемые контуры разделены на группы по природе описываемых свойств фрейма (например, группа геометрических форм обрабатываемых поверхностей, группа классов точности, группа применяемых материалов и т. п.), то R^L *(F (T))* имеет вид, аналогичный (15. 40) или (15.41), а в общем случае — (15.42). Аналогичный вид имеют типовые отношения R^L *(F* (τ_k)) — отношения вида (15 10) — между контурами каждого оператора, соответствующие строкам матрицы контуров (15.52) модели операторов.

Отношения вида (15.12) между одинаковыми контурами разных операторов, соответствующие столбцам матрицы контуров (15.52), определяют составы операторов, с применением которых может быть реализован данный контур. Если контур F_j может быть реализован только определенной совокупностью операторов, то R^L (F_j (T)) имеет вид, аналогичный (15. 43). В этом случае контур F_j включает в себя *m* контуров более низкого уровня, каждый из которых и реализуется определенным оператором; варианты выполнения процесса образования фреймов по контуру F_j в этом случае отсутствуют.

Если контур F_j может быть реализован совокупностью *N* операторов, но каждый раз лишь единственным оператором из каждой *k*-й группы, $k \in N$, то R^L *(*F_j *(T))* имеет вид, аналогичный (15. 49). В этом случае в каждой *k*-й группе возможны *m* вариантов использования различных операторов, реализующих F_j и, следовательно, различные варианты процессов образования фреймов. Если R^L *(*F_j*(T))* имеет вид, аналогичный (14.44) или (15.48), то в САОФ есть операторы, каждый из которых может реализовать контур F_j либо без участия других операторов, либо совместно с другими операторами.

Наличие вариантов использования различных наборов операторов может быть представлено и с помощью отношений R^L *(T),* определяющих возможность вхождения операторов в процесс образования фреймов T_i образования конкретного образования фрейма: если в T_i может входить любой набор операторов $\tau_k \in T$, то R^L *(T)* имеет иид (15.28); если существует группа операторов, из которых в T_i может входить лишь какой-либо единственный оператор, то R^L *(T)* имеет вид (15.31), и т. д.

Процесс образования фреймов осуществляется на основе установленных отношений между свойствами фреймов и САОФ. Так, состояние контуров фрейма $F\ (A)_k$ после воздействия оператора τ_k определяется как функция от предшествующего состояния $F\ (A)_{k-1}$ контуров фрейма и контуров F (τ_k) оператора, которые

взаимодействуют с контурами фрейма. Состав контуров $F\ (\tau_k)$ операторов определяется свойствами элементов материальной системы САОФ.

Одному оператору τ_k может соответствовать один или несколько элементов процесса образования фреймов (процедур, операций, и т. д.), выполняемых с помощью различных элементов САОФ (технических средств). Взаимодействие $F_i\ (\tau_k)$ с $F_i\ (A)_{k-1}$ возможно только в том случае когда все эти элементы обладают необходимыми свойствами, позволяющими получить контур $F_i\ (A)$ фрейма. Эти свойства САОФ описываются формулой

$$F_i(\tau_k) = \bigwedge_{j=1}^{m} [F_i(\tau_k)_j \wedge (F_i(\pi_k)_1 \wedge F_i(\pi_k)_2 \wedge \ldots \wedge F_i(\pi_k)_n)_j],$$

где $(\tau_k)_j$— элемент процесса образования фреймов; $(\pi_k)_j$ — элемент САОФ; m — число элементов процесса образования фреймов, используемых при реализации τ_k; n — число элементов САОФ, применяемых при использовании τ_k.

Полный состав контуров фреймов может рассматриваться либо как просто множество $F\ (A)$, либо как булево векторное пространство **F** (A); в последнем случае отдельные состояния фрейма — $F\ (A)_{k-1}$, $F\ (A)_k$ и т. д. — рассматриваются как булевы векторы в пространстве **F** (A). Точно так же полный состав контуров, которые могут быть реализованы в данной САОФ P, представляется как булево векторное пространство **F** (P), а операторы τ_k характеризуются булевыми векторами $F\ (\tau_k)$ в пространстве **F** (P). Структура пространства **F** (P) аналогична структуре **F** (A); более того, согласно выражению (15.51) **F** (A) и **F** (P) могут рассматриваться как составные части единого пространства контуров фрейма и САОФ.

При математическом моделировании процесса образования фреймов в булевом векторном пространстве используются две формы представления свойств фрейма и САОФ — дизъюнктивная и конъюнктивная. При дизъюнктивной форме исходное состояние контура $F_i(A)$ фрейма, подлежащего реализации, обозначается F_i (A) = 0, а конечное (после реализации) — $F_i\ (A)$ = 1. Состояние контура $F_i\ (\tau_k)$ оператора, реализующего контур $F_i\ (A)$, обозначается $F_i\ (\tau_k)$= 1; воздействие оператора на фрейм по контуру F_i определяется с помощью операции логического сложения (дизъюнкции)) контуров F_i (A) и $F_i\ (\tau_k)$. Матрица контуров САОФ в этом случае называется дизъюнктивной. Примером такой матрицы является матрица (рис. 15.7, а) контуров табличной модели операторов механической обработки группы деталей (см. рис. 15.3).

		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_1'	F_2'	F_3'	F_4'	F_5'	F_6'
Подрезка торца	τ_1	●									●	●	●	●	●	●
Обтачивание	τ_2		●								●	●	●	●	●	●
Обтачивание	τ_3			●							●	●	●	●	●	
Сверление	τ_4				●						●	●	●	●	●	●
Сверление	τ_5					●					●		●			
Растачивание конуса	τ_6						●				●	●	●			
Обтачивание	τ_7							●			●	●			●	
Обтачивание фаски	τ_8								●		●					
Отрезка	τ_9									●	●	●	●	●	●	●

а) б)

Рис. 15.7. Табличные модели операторов изготовления группы деталей (см. рис. 15.3): *а* — дизъюнктивная модель, б — конъюнктивная модель.

В этом случае логические отношения между контурами $F\ (A)$ всей группы деталей описываются функцией (15.36), отношения $R^L\ (F\ (\tau_к))$, соответствующие строкам матрицы контуров, описываются функцией, аналогичной (15.37), а $R^L\ (F_j\ (\mathrm{T}))$ — функцией, аналогичной (15.45).

При конъюнктивной форме представления исходное состояние контура $F_i\ (A)$, подлежащего реализации, обозначается F_i (А) =1; состояние контура $F_i(\tau_k)$ оператора, участвующего в реализации контура F_i (А), также обозначается $F_i\ (\tau_k)$= 1. Воздействие оператора на фрейм по контуру F_i определяется с помощью операции логического умножения (конъюнкции) контуров $F_i\ (A)$ и $F_i\ (\tau_k)$. Матрица контуров САОФ в этом случае называется конъюнктивной матрицей контуров. Пример такой матрицы показан на рис. 15.7, *б*. Здесь логические отношения между контурами $F\ (A)$ всей группы фрейма деталей описываются функцией (15.37), так как каждый контур F'_i соответствует фрейму детали a_i в целом; отношения $R^L\ (F\ (\tau_k))$, соответствующие строкам матрицы контуров, описываются функцией, аналогичной (15.36), а $R^L\ (F_j\ (T))$ — функцией, аналогичной (15.43).

При дизъюнктивной форме состав контуров фрейма после выполнения k-й операции определяется по формуле

$$F\,(A_k) = F\,(A)_{k-1} \vee F\,(\tau_k). \quad (15.53)$$

Поскольку между контурами $F_i \in F$ (А) существуют логические связи вида (15.9) по условиям их существования, фактический состав контуров $F\ (A)_k$, реализуемых при воздействии оператора τ_k, может оказаться большим, чем вычисляемый по формуле (15.53). Поэтому

после вычисления $F(A)_k$ по формуле (15.53) необходимо определить контуры, не входящие в $F(\tau_k)$, но реализуемые из-за логических связей внутри $F(A)$, и включить их в состав $F(A)_k$.

Возможны четыре различные группы контуров $F_i(A)$, состояния которых характеризуют возможности оператора τ_k:

1) контуры, которые могуг быть реализованы оператором τ_k:

$$F^{\mathrm{I}}(A)_k = \overline{F}(A)_{k-1} \wedge F(\tau_k), \qquad (15.54)$$

где $\overline{F}(A)_{k-1}$ — результат инверсии вектора $F(A)_{k-1}$;

2) контуры, которые не могут быть реализованы оператором τ_k:

$$F^{\mathrm{II}}(A)_k = \overline{F}(A)_{k-1} \wedge \overline{F}(\tau_k),$$

где $\overline{F}(\tau_k)$ — результат инверсии $F(\tau_k)$;

3) контуры, которые уже реализованы, но могли бы быть реализованы оператором τ_k:

$$F^{\mathrm{III}}(A)_k = F(A)_{k-1} \wedge F(\tau_k);$$

4) контуры, которые уже реализованы, но не могли бы быть реализованы оператором τ_k:

$$F^{\mathrm{IV}}(A)_k = F(A)_{k-1} \wedge \overline{F}(\tau_k).$$

Если известна последовательность T_i операторов τ_k, $k = 1, 2, ..., m$, то состав контуров, которые могут быть реализованы данной последовательностью операторов, определяется по формуле

$$F(T_i) = \bigvee_{k=1}^{m} F(\tau_k). \qquad (15.55)$$

Состав операторов, входящих в T_i, является достаточным для образования фрейма A при условии

$$F(A) = F(T_i) \wedge F(A), \qquad (15.56)$$

где $F(A)$ — булев вектор, соответствующий образованному фрейму.

Если это условие не выполняется, в дополнение к T_i можно взять другие последовательности из множества $T_J = \{T_i, T_j, ..., T_k\}$ — такие, что

$$F(A) = F(T_J) \wedge F(A), \qquad (15.57)$$

где

$$F(T_J) = F(T_i) \vee F(T_j) \vee \ldots \vee F(T_k). \qquad (15.58)$$

Тогда состав операторов, входящих в T_i, T_j, ..., T_k, является достаточным для образования фрейма A.

В случае, когда $F\{A)_{k-1}$, $F(A)_k$ и $F(\tau_k)$ представляют собой множества контуров, равных единице, состав контуров $F(A)_k$ после выполнения k-*й* операции равен

$$F(A)_k = F(A)_{k-1} \cup F(\tau_k).$$

В этом случае состав контуров фрейма, которые могут быть реализованы при использовании оператора τ_k, определяется по формуле

$$F(A)_k = F(\tau_k) \setminus F(A)_{k-1}$$

как разность множеств $F(\tau_k)$ и $F(A)_{k-1}$. Состав контуров, которые могут быть реализованы последовательностью T_i операторов τ_k, $k = 1, 2, ..., m$, равен

$$F(T_i) = \bigcup_{k=1}^{m} F(\tau_k). \qquad (15.59)$$

Состав операторов T_i является достаточным для образования фрейма A, если выполняется условие

$$F(A) = F(T_i) \cap F(A). \quad (15.60)$$

Если условие (15.60) не выполняется, в дополнение к T_i можно взять другие последовательности из $T_J = \{T_i, T_j, ..., T_k\}$, такие, что

$$F(A) = F(T_J) \cap F(A), \quad (15.61)$$

где

$$F(T_J) = F(T_i) \cup F(T_j) \cup \ \ldots \ \cup F(T_k); \qquad (15.62)$$

тогда состав операторов, входящих в $T_i, T_j, ..., T_k$, является достаточным для образования фрейма A.

При конъюнктивной форме состав контуров фрейма $F(A)_k$, реализуемых при выполнении k-й операции, определяется по формуле

$$F(A)_k = F(A) \wedge F(\tau_k). \quad (15.63)$$

Группы контуров, характеризующих технологические возможности операторов при конъюнктивной форме связи, определяются следующим образом: контуры, в реализации которых может участвовать оператор τ_k, — по формуле (15.63), а контуры, в реализации которых не может участвовать оператор τ_k, — по формуле

$$F^{\text{II}}(A)_k = F(A) \wedge \overline{F}(\tau_k).$$

Состав контуров, в реализации которых может участвовать последовательность T_i, операторов τ_k, определяется по формуле (15.55), а несколько последовательностей — по формуле (15.58). Состав операторов достаточен для образования фрейма A, если выполняется условие (15.56) или (15.57). Если $F(A)_{k-1}$, $F(A)_k$ и $F(\tau_k)$ представляются как множества контуров, равных единице, то состав контуров $F(A)_k$, равен

$$F(A)_k = F(A) \cap F(\tau_k).$$

При этом состав контуров, в реализации которых может участвовать совокупность T_i операторов τ_k, определяется по формуле (15.59), а несколько совокупностей — по формуле (15.62). Состав операторов достаточен для образования фрейма A, если выполняется условие (15.60) или (15.61).

15.4. Автоматизированное образование фреймов процессов по типовым структурно-логическим моделям

Содержание и взаимосвязь процедур и операций при автоматизированном образовании фреймов процессов зависят от класса модели операторов и вида логических отношений и связей между контурами фрейма и операторов.

Образование фрейма процесса по табличной модели $S_1(T)$ сводится к поиску в матрице контуров (15.52) строк, соответствующих требуемому составу контуров фрейма. Если матрица контуров табличной модели дизъюнктивная (см. рис. 15.7, а), то образование фрейма процесса T_i осуществляется следующим образом: а) поочередно, начиная с первой строки, определяются операторы τ_k, у которых булев вектор $F^1(A)_k$, вычисляемый по формуле (15.54), не равен нулю — каждый такой оператор включается в состав процесса образования фрейма процесса T_i; б) образование T_i заканчивается, если выполнено условие (15.56).

Если матрица контуров конъюнктивная (см. рис. 15.7, *б),* то образование фрейма процесса осуществляется следующим образом: а) поочередно, начиная с первой строки, определяются операторы τ_k, у которых булев вектор $F(A)_k$, вычисляемый по формуле (15.63), не равен нулю — каждый такой оператор включается в состав процесса T_i; б) формирование T_i заканчивается, когда просмотрены все строки матрицы контуров и выполнено условие (15.56).

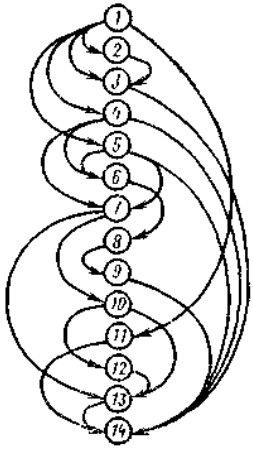
Табличные модели, различающиеся видом логических отношений между контурами в матрице контуров, в разной степени раскрывают технологические возможности САОФ. Например, по матрице на рис. 15.7, *а* можно сформировать фрейм технологического процесса для любой новой детали, которая по составу и взаимосвязи обрабатываемых поверхностей и других контуров войдет в группу деталей вида показанных на рис. 15.3.

Иное положение с матрицей, приведенной на рис. 15.7, *б:* она не может быть использована для формирования фрейма технологического процесса изготовления новой детали данной группы до тех пор, пока в матрицу не будет добавлен новый столбец, соответствующий контуру новой детали.

Сетевая модель включает в себя матрицу контуров, описание логических отношений между контурами и граф G = *(T, C)* взаимосвязи операторов по возможной последовательности их реализации. Например, на рис. 15.8 приведена сетевая модель S_5 *(T)* механической обработки деталей вала с шестерней (см. рис. 15.4).

		F_1	F_2	F_3	F_4
Черновая обработка	τ_1	●	●	●	●
Сверление	τ_2				●
Протягивание	τ_3				●
Точение I	τ_4	●	●	●	●
Точение II	τ_5	●	●	●	●
Шлифование I	τ_6		●		
Нарезание зубьев	τ_7	●			
Термообработка	τ_8		●		
Шлифование II	τ_9		●		
Шлифование зубьев	τ_{10}	●			
Фрезерование паза	τ_{11}				●
Притирка	τ_{12}	●			
Обкатка	τ_{13}	●			
Контроль	τ_{14}	●	●	●	●

а)



б)

Рис. 15.8. Сетевая модель операторов механообработки контуров: *а* — матрица контуров операторов; б —граф взаимосвязи технологических операторов

Матрица контуров этой модели является конъюнктивной; логические отношения R^L *(F* ($\tau_к$)*)* в ней описываются функцией нида (15.36), а R^L *(*F_j *(T))* — функцией, аналогичной (15.49). Логические отношения R^L *(F (A))* между контурами деталей описываются функцией (15.36).

Образование фрейма процесса по сетевой модели S_2 *(T)* или S_5 (*T*) осуществляется следующим образом: а) вычисляется очередной путь μ_i в графе *G* = (*T*, C); б) из матрицы контуров [T× *F (T)]* формируется новая матрица [T_i × *F* (T_i)], состав и порядок строк в которой соответствуют составу и порядку вершину и пути μ_i; в) по матрице [T_i × *F* (T_i)], как по табличной модели, формируется процесс

образования фрейма T_i; если он не обеспечивает получения всех контуровфрейма , определяется следующий путь в графе G = *(T, C)* и т. д.

Граф *G = (T, C)* процесса образования фрейма, образовываемого по сетевой модели S_7 *(T)*, S_8 *(T)* или S_{11} *(T),* — либо дерево, либо граф общего вида. Такая структура процесса образования фрейма может быть получена объединением нескольких простых элементарных путей в графе *G = (T, C)* сетевой модели, таких, что совокупность $T_J = \{T_i, T_j, ..., T_k\}$, соответствующих этих путям процессов образования фрейма, удовлетворяет условию (15.57).

Каждая перестановочная модель включает матрицу контуров, аналогичную матрице контуров табличной или сетевой модели, и набор отношений, определяющих условия существования процесса образования фрейма как перестановки из операторов, удовлетворяющей заданным ограничениям. Процесса образования фрейма T_m возможен, если для каждого оператора $\tau_к \in T_m$ выполнены условия:

1) среди ранее реализованных операторов, образующих подмножество $T^0_k \subset T_m$, есть такие, которые образуют хотя бы один набор B_i (τ_k) операторов; после реализации их возможна реализация τ_k:

$$\forall \tau_k \in T_m [\exists B_i (\tau_k) (B_i (\tau_k) \subseteq T^0_k)]; \quad (15.64)$$

2) в составе $T_к$ нет таких операторов, которые образуют хотя бы один набор W_j (τ_k) операторов, после реализации которых реализация τ_k невозможна:

$$\forall \tau_k \in T_m [\forall W_j (\tau_k) (W_j (\tau_k) \not\subset T^0_k)]. \quad (15.65)$$

К условиям такого вида приводятся условия обеспечения определенности базирования, условия доступа в зону выполнения операции, условия обеспечения требуемой точности замыкающего звена технологической размерной цепи и т. п. Формирование процесса образования фреймов по перестановочным моделям S_4 (T), S_6 (T), S_{10} (T) и S_{12} (T) начинается с определения по матрице контуров *[T × F (T)]* состава $T_i = \{\tau_1, \tau_2, ..., \tau_n\}$ операторов, удовлетворяющих условию (15. 56); в моделях S_3 *(T)* и S_9 (T) таким составом будет полный состав T операторов модели.

Далее формирование процесса образования фреймов по моделям S_3 *(T)*, S_4*(T)* и S_6 (T) осуществляется следующим образом:

а) вычисляется очередная перестановка

$$T_m = (\tau_{m_1}, \tau_{m_2}, \ldots, \tau_{m_n});$$

б) проверяется выполнение условий вида (15.64) и (15.65); если эти условия выполняются, T_m является возможным вариантом процесса образования фреймов.

Формирование процесса образования фреймов по моделям S_9 (Т), S_{10} (Т) и S_{12} (Т) при известном составе операторов включает построение графа G_i = (T_i, C_i,) смежности операторов и установление отношений порядка между вершинами этого графа в соответствии с условиями (15.64) и (15.65).

Реальной подсистеме соответствует набор нескольких моделей *S (T), S* (П) и S *(P)* различных классов. Эти модели различаются по видамповерхностей, степени детализации описания процессов образования фреймов и т. д. Взаимосвязь моделей определяет содержание и последовательность решения задач образования фреймов. Для выбора оптимальной структуры процесса образования фреймов в САОФ необходимо моделировать сам процесс образования фреймов. В системе ИСТРА процесс образования фреймов моделируется в виде многоуровневой иерархической системы Т, компоненты которой описываются операторами $\tau_k \in T$, характеризующими отдельные операции, процедуры или этапы образования фреймов - геометрические построения, прочностные расчеты, расчеты размерных цепей и т. п. Воздействие τ_k на модель объекта образования фреймов аналогично воздействию оператора на объект образования фреймов — оно приводит к изменению определенных свойств и отношений в модели объекта. Поэтому, с математической точки зрения, моделирование процесса образования фреймов аналогично моделированию производства изделия. Операторы τ_k осуществляют следующие структурные преобразования исходной модели образовываемого фрейма *A:* исключение из *A* элемента a_i; добавление в *A* элемента a_i; замена в *A* элемента a_i на a_j; исключение связи между a_i и a_j (т. е. замена в матрице $[A \times A]$ элемента $c_{i\,(j)} = 1$ на $c_{i\,(j)} = 0$); введение связи между a_i и a_j и т. д. Одновременно с изменением структуры элементов объекта, как следствие, изменяется и структура его контуров — например, введение нового элемента a_i в фрейм конструкции приспособления *A* требует введения контура сопряжения a_i с другими элементами конструкции.

Преобразование исходной модели *A* может осуществляться и путем преобразования структуры контуров *F (A).* Преобразование модели на уровне логических свойств и отношений осуществляется изменением состава логических переменных и логических связок между ними, а на уровне количественных свойств и отношений — изменением числовых величин и количественных связей между ними.

Если взаимосвязь операторов в моделях S_i *(T)* СОАФ установлена заранее, то такие модели аналогичны табличным, сетевым и перестановочным моделям операторов, и разработка процессов проектирования фреймов по таким моделям аналогична образованию фреймов технологических процессов. Если назначение и вхождение проектных операторов в конкретные процессы проектирования заранее не определены, то модель S_i *(T)* включает

$$\{[T \times \bar{F}(T)],\ [T \times F^{+}(T)],\ [T \times F^{0}(T)],\ G = (T,\ C)\},$$

где [Т × *F* — (Т)] —матрица заходящих контуров, а $[T \times F^{+}(T)]$ — матрица исходящих контуров проектных операторов, характеризующих исходные данные и данные о фрейме проектируемого объекта, получаемые при реализации каждого оператора τ_k;

$[T \times F^{0}(T)]$ — матрица собственных контуров модели, характюризующих функциональные свойства и технико-экономические показатели операторов $\tau_k \in T$ в процессе проектирования; G = (T, C) — граф взаимовязи операторов по возможной последовательности их выполнения в процессе образования фреймов. Например, при расчете погрешности замыкающего звена размерной цепи заходящим контурам соответствуют погрешности составляющихзвеньев, а исходящим контуром будет величина погрешности замыкающего звена размерной цепи. Разработка процесса образования фрейма по такой модели осуществляется следующим образом.

1. По матрице $[T \times F^{+}(T)]$ выбирают проектный оператор τ_k, состав исходящих контуров $F^{+}(\tau_k)$ которого содержит требуемую информацию об образовываемом фрейме объекте *A,* необходимую на завершающем этапе образования фрейма; оператор τ_k будет конечным оператором искомого процесса образования фрейма.

2. По матрице $[T \times F^{-}(T)]$ определяют состав $F^{-}(\tau_k)$ заходящих контуров оператора τ_k. Затем по графу G = (Г, С) определяют совокупность операторов, предшествующих τ_k и смежных с ним; из их числа выбирают набор $\{\tau_{k_1}, \tau_{k_2}, \ldots, \tau_{k_m}\}$ таких операторов, объединение исходящих контуров (τ_{ki}) которых включает в себя, как подмножество, состав $F^{-}(\tau_k)$ заходящих контуров τ_k:

$$\left(\bigcup_{i=1}^{m} F^{+}(\tau_{k_i})\right) \supseteq F^{-}(\tau_k),$$

или, если $F^{+}(\tau_k)$ и $F^{-}(\tau_k)$ рассматриваются как булевы векторы — строки матриц контуров, то

$$\left(\bigvee_{i=1}^{m} F^{+}(\tau_{k_i})\right) \wedge F^{-}(\tau_k) = F^{-}(\tau_k).$$

Далее по матрице $[T \times F^{-}(T)]$ определяются составы $F^{-}(\tau_k)$ заходящих контуров операторов τ_{ki}^{-}, а по графу $G = (T, C)$ — совокупность предшествующих τ_{ki} операторов и т. д. Разработка процесса образования фрейма завершается, когда определен исходный проектный оператор, воздействующий непосредственно на исходную модель объекта, представленную в задании на образование фрйма.

При автоматизированном образовании фреймов в САОФ должен быть обеспечен максимальный уровень типизации и унификации процессов образования фреймов. Уровень унификации образовываемых фреймов определяет возможность использования табличных, сетевых или перестановочных моделей при образовании фреймов. Применение единой системы математическою моделирования образования фреймов позволяет организовать автоматизированное образование фреймов в САОФ. Вначале используют табличные модели, и лишь при отсутствии стандартного (типового) решения — сетевые, а затем и перестановочные модели. При такой организации автоматизированного образования фреймов полученное решение будет максимально унифицированным.

16. Построение моделей элементов технических устройств и формализованный язык их описания для автоматизированного образования фреймов

16.1. Виды моделей элементов технических устройств и способы их декомпозиции

Для автоматизации образования фреймов необходима не вся информация, содержащаяся в системной модели детали, а только определенная ее часть, которая образует фрейм технологической модель обрабатываемой детали (ТМ-Д) и различных ее состояний от C_o до $C_к$. В связи с этим ТМ-Д строится путем исключения из общей модели одних системных характеристик и описания других с требуемой степенью обобщения и детализации.

Так, функция детали и предполагаемая история ее функционирования отображаются на стадии конструирования в конфигурации детали, размерно-точностной структуре и параметрах элементов. В связи с

этим влияние указанных системных характеристик на технологический процесс в большинстве случаев проявляется не непосредственно, а через структуру и параметры обрабатываемой детали.

Не все, а только некоторые сведения необходимы о связях детали с окружающей средой, т. е. с другими деталями механизма или изделия, с которыми рассматриваемая деталь связана и взаимодействует в процессе функционирования.

При образовании фреймов технологических процессов изготовления деталей следует задать только поверхности основных баз и выполняемые ими функции. Эти сведения наряду с другими данными будут использоваться при формировании операций и их последовательности на основе принципа совмещения конструкторских баз с технологическими. Функция базовых поверхностей задается отношениями базирования

$$БП_1\delta_1\{q_x,\ q_y,\ \ldots\},$$
$$БП_k\delta_k\{\varphi_x,\ q_r,\ \varphi_y\},$$

которые определяют степени свободы q_x, ..., φ_x, …, фиксируемые каждой базовой поверхностью *БП*. Например, функции поверхностей $П_{11}$, $П_{13}$ основных баз детали (см. рис. 2) описываются отношениями $П_{11}\delta_1 OZ$, $П_{12}\delta_2 Z$. Первое обозначает, что $П_{11}$ фиксирует ось *OZ*, а второе — положение детали вдоль этой оси. Совокупность отношений базирования характеризует упрощенную, но достаточную для образования фреймов технологических процессов модель связей детали с окружающей средой.

Наиболее подробные сведения для автоматизации процессов образования фреймов технологического проектирования требуются о структурном составе детали и взаимосвязях ее элементов на различных уровнях расчленения. В соответствии с методологией системно-структурного анализа детали машин как сложные объекты при конструктивно-технологическом анализе и образовании фреймов условно расчленяются на несколько взаимосвязанных компонент, последние на еще более простые части, и так до уровня базовых элементов, дальнейшее расчленение которых нецелесообразно с точки зрения решаемых задач. В результате формируется граф структурного состава детали *G(Q, U)*. Его вершинами *Q* служат компоненты детали различных уровней, а дугами *U* — признаки, по которым производится расчленение детали на каждом уровне.

В зависимости от целей и задач образования фреймов применяются различные способы представления и расчленения детали на компоненты. В системе автоматизации образования фреймов процессов ковки, объемной штамповки, литья, сварки деталь

рассматривается как совокупность объемных компонент различной сложности. В качестве исходных структурных элементов принимаются цилиндр, конус, шар, бочка, параллелепипед, призматические, конические многогранники и др. На этих элементах отмечаются обрабатываемые поверхности, так как на них назначаются припуски. На рис. 16.1, 16.2 показаны чертеж детали и графы ее структурного состава.

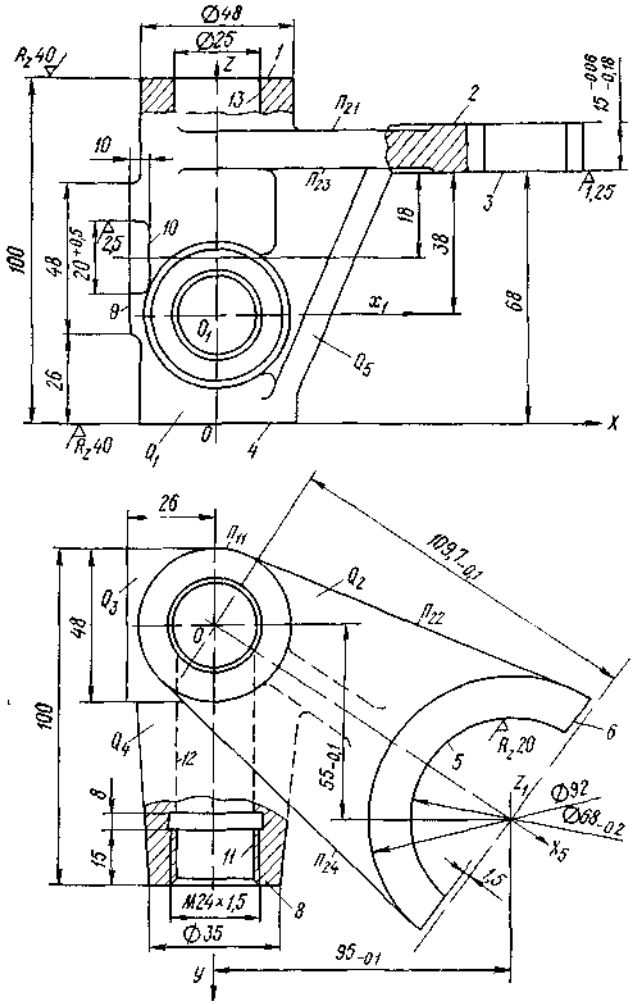


Рис. 16.1. Вилка переключения поверхности

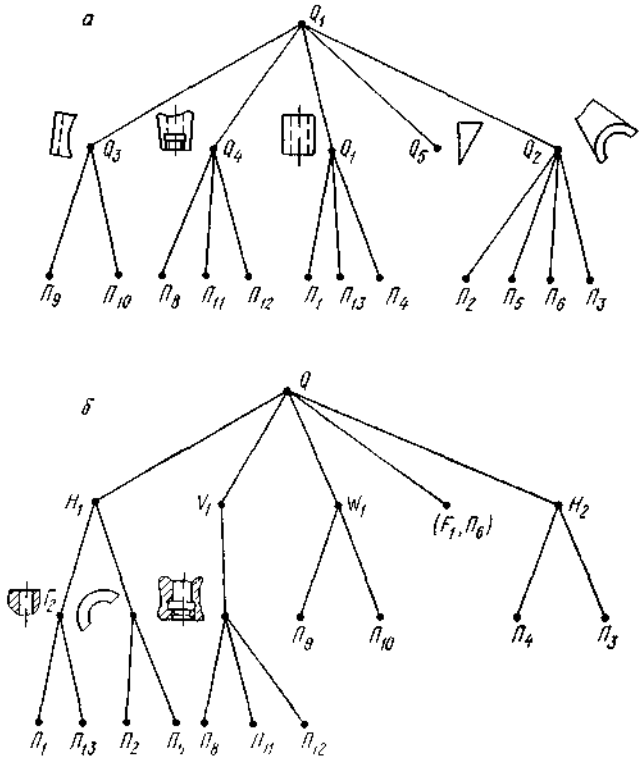


Рис. 16.2. Графы структурного состава детали при различных способах расчленения: *а* —на объемные элементы; *б*—на совокупности поверхностей

На первом уровне деталь расчленяется на крупные объемные элементы (ступица, вилка, ребро, бобышка), на втором выделяются совокупности обрабатываемых поверхностей.

В САОФ фреймы технологических процессов механической обработки детали и ее состояния представляются в виде пространственных фигур, ограниченных совокупностью поверхностей, вид, размеры и взаимное положение которых определяются выполняемой деталью функцией или требованиями последующей обработки. За исходные структурные элементы целесообразно принимать элементарные, нормализованные и типовые поверхности. Это вызвано тем, что указанные поверхности непосредственно связаны с первичными структурными элементами

технологического процесса (переходами) и обрабатываются стандартным или нормализованным инструментом.

При образовании фрейма конструктивно-технологической модели необходимо выбрать рациональный способ расчленения и построить граф структурного состава обрабатываемой детали. Решение этих вопросов можно проиллюстрировать на примере работы технолога-модельщика при разработке технологии изготовления деревянных моделей сложной формы. В процессе чтения чертежа модельщик определяет, из каких основных частей должна состоять модель и как они между собой связаны.

Получившиеся части он разделяет на более мелкие объемные элементы и устанавливает связь между ними, и так до тех пор, пока не получатся простые объемные элементы: призмы, плиты, ребра, цилиндрические и конические бобышки и т. д. Выявив исходные объемные элементы, модельщик в соответствии с принятым иерархическим порядком расчленения составляет технологию изготовления модели.

Естественно, что вариантов разделения сложной детали на составные части и синтеза ее как целого может быть много, технолог, анализируя их, выбирает более рациональный, который обеспечивает наименьшую трудоемкость изготовления модели, последующей формовки и отливки детали.

На основе опыта кодирования и разработки алгоритмов проектирования технологии обработки сложных деталей можно указать некоторые системные рекомендации «расчленения» детали и формирования графа структурного состава *G(Q, U).*

1. Расчленение надо производить так, чтобы на различных уровнях форма детали могла быть представлена в виде сочетания типовых и часто встречающихся компонент (соосные отверстия, концентрично расположенные элементы, направляющие, карманы и др.). В этом случае образование фрейма технологии обработки приведенных элементов будет осуществляться посредством типовых алгоритмов.

2. При построении различных вариантов графа структурного состава *G(Q, U)* необходимо на основе анализа формообразующих и размерных связей выбрать вариант, характеризующийся наиболее слабыми и малочисленными связями между компонентами и более сильными связями элементов в составе каждой компоненты.

3. Расчленение должно быть таким, чтобы обрабатываемые поверхности не разрывались на отдельные куски, отнесенные к различным компонентам детали. В противном случае потребуются дополнительные алгоритмы анализа и восстановления формы и размеров неудачно расчлененных элементов.

Типовость компонент детали, слабые и малочисленные связи между ними обеспечивают построение более простых и эффективных алгоритмов синтеза фреймов технологии обработки отдельных компонент и детали в целом. На основании указанных рекомендаций при представлении детали в виде совокупности поверхностей выделяются компоненты нескольких уровней. На первом деталь рассматривается как сочетание основных, дополнительных сторон и контуров. В соответствии с правилами проекционного черчения в детали выделяются шесть основных сторон V_h V_{2l} H_h H_2, W_u W_2 и произвольное число дополнительных F, расположенных под разными углами к основным (рис. 16.3).

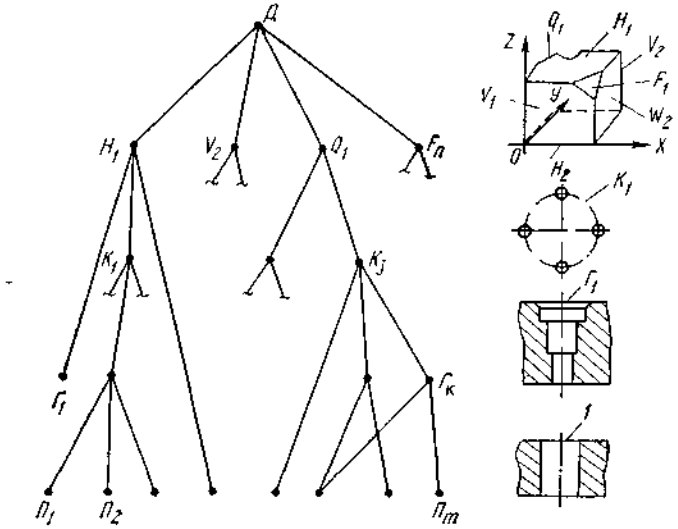


Рис. 16.3. Структурные уровни декомпозиции деталей машин

В деталях машин часто встречаются участки внешней конфигурации, которые нельзя отнести ни к основным, ни к дополнительным сторонам. Они представляют собой совокупность взаимосвязанных плоских и линейчатых поверхностей и образуют контур Q некоторой части детали.

К объектам второго уровня, обозначаемым *К,* относятся совокупности простых или сложных элементов, расположенных по определенному закону: по прямой, окружности, в виде прямоугольной сетки и др. Объекты Г третьего уровня представляют собой соединение взаимосвязанных простых поверхностей, образующих соосные

наружные и внутренние участки, многоступенчатые уступы, выступы, выемки, направляющие, карманы, колодцы и др. Последний уровень расчленения образует элементарные, типовые и нормализованные поверхности. Число уровней декомпозиции детали может быть произвольным и определяется ее конфигурацией.

На рис. 16.2, *а* показан граф структурного состава детали при представлении ее в виде совокупности поверхностей. На первом уровне все поверхности отнесены к четырем основным сторонам детали H_1, H_2, V_1, W_2 и одной дополнительной F_1, расположенной под углом к основным. На втором уровне каждая сторона расчленяется на отдельные поверхности и т. д. Представление детали в виде сочетания сторон и расположенных на них обрабатываемых поверхностей наиболее просто согласуется с моделями, описывающими рабочие зоны станков, т. е. взаимное расположение в пространстве шпинделя, стола и величины их перемещений.

При изготовлении деталей на станках с цифровым программным управлением и автоматизации вычерчивания операционных чертежей необходим следующий уровень расчленения, основными структурными элементами которого будут линии, точки. Из них образуются траектории обработки поверхности и составляются операционные чертежи.

При комплексной автоматизации образования фреймов, включающей в себя фреймы процессов получения заготовок, их механической обработки на станках с ручным и программным управлением, вычерчивания операционных чертежей и схем наладок, нужны все три вида представлений детали. Кроме того, указанные способы описания состояний детали должны быть взаимосвязаны так, чтобы обеспечить простые алгоритмы преобразования из одного представления в другое.

Совокупность сведений о системных характеристиках детали, необходимых и достаточных для образования фреймов процессов ее изготовления, образует фрейм технологической модели обрабатываемой детали

$$Q_{\text{кт}} = \begin{cases} \langle H_y,\ G(Q,\ U),\ (S_\text{ф},\ S_p,\ S_\delta),\ Z_0 \rangle^1, \\ \{\langle (S_{\text{ф}i},\ S_{pi},\ S_{\delta i}),\ Z_i \rangle^2\},\ i = 1 \div n, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ \langle K\Phi_q,\ Z_q \rangle^k,\ q = 1 \div m. \end{cases}$$

С помощью первого соотношения описываются системные характеристики детали как целого: основные конструкторские базы детали и степени свободы, которые они фиксируют $H_y=\{(БП\delta_q)_j\}$, структурный состав детали $G(Q,U)$, ее формообразующую и размерные структуры $S_\text{ф}$, S_p, S_δ, а также общие сведения о детали Z_0. На

промежуточных уровнях расчленения описывается структура и параметры объектов *i* уровня. На последнем *К* уровне задаются только форма *КФ* и параметры Z_k базовых структурных элементов детали. ТМ-Д является структурно-параметрической моделью. Полнота описания структуры и параметров обрабатываемой детали позволяет на основе этой модели образовать фрейм операционной технологии и управляющие программы для станков с ЧПУ.

Для решения в САОФ других задач могут потребоваться более простые модели, описывающие укрупненно отдельные системные характеристики детали и различных ее состояний в процессе изготовления. Так, при автоматизации образования фреймов технологических процессов на основе типизации и группового метода инвариантные черты формы детали, общие для всего типа, группы, отражаются в структуре типового и группового процессов и соответствующих им алгоритмов. В связи с этим кодируются только те особенности формы конкретной детали, которые влияют на выбор требуемых операций и переходов. Например, для зубчатых колес к таким особенностям формы относятся наличие ступицы и проточек справа и слева от зубчатого венца, вид центрального отверстия и др. ТМ-Д для этих целей включает код формы детали, общие сведения о ней, подграф разменных связей некоторых элементов детали $P_1(\Pi,\ L) \subset P(\Pi,\ L)$ и основные их параметры $Z_э \subset Z$:

$$Q_{тмд} = \begin{cases} КФ,\ P_1(\Pi, L),\ Z_д, \\ \{Z_{эi}\}. \end{cases}$$

Для решения задач автоматизированного поиска фреймов деталей и процессов-аналогов, классификации и группирования необходимо задать код формы *КФ,* общие сведения о детали Z_n и некоторые параметры отдельных поверхностей $Z' \subset Z_э$

$$Q_{кд} = \begin{cases} КФ,\ Z_д, \\ \{Z'_i\}. \end{cases}$$

Таким образом, методологическая основа системно-структурного анализа позволяет создавать фреймы технологических моделей деталей требуемой степени детализации и функционального назначения, а также проблемно-ориентированные языки для САОФ.

16.2. СТРУКТУРНО-КИНЕМАТИЧЕСКИЕ МОДЕЛИ И МЕТОДЫ КОДИРОВАНИЯ ПОВЕРХНОСТЕЙ

Большое конструктивное разнообразие типовых и нормализованных поверхностей, встречающихся в деталях машин, затрудняет их непосредственное выявление и систематизацию с целью создания научно обоснованных способов кодирования в автоматизированных системахобразования фрейсов. Задача резко упростится, если в основу систематизации положить закономерности формообразования поверхностей в сочетании с конструктивно-технологическими характеристиками, учитывающими технологию их изготовления. Существует несколько способов задания поверхностей: аналитический, кинематический, каркасный и др.

Большинство поверхностей в деталях машин относится к числу кинематических, т. е. они могут быть получены перемещением образующей линии l относительно направляющей m по определенному закону, описываемому оператором формообразования ω. Форму таких поверхностей будем задавать в виде отношения

$$\Phi_п = l\omega m.$$

Оператор ω определяет вид перемещения образующей (поступательное, вращательное) или характер изменений углового положения образующей относительно направляющей в процессе движения

$$\omega = \omega_o + \Delta\omega\ (m),$$

где ω_0 — вектор начального положения образующей: $\Delta\omega(m)$ — изменение вектора начального положения образующей при ее движении по направляющей.

В процессе движения формообразования образующая может сохранять свои размеры и форму или непрерывно изменять их. В результате ее размеры B и форма Φ будут изменяться в зависимости от положения на направляющей

$$B_l = B_0 + \Delta B\ (m),$$

$$\Phi_l = \Phi\ (m),$$

где B_0 — вектор исходных размеров образующей; ΔB (m) — изменение вектора начальных размеров образующей при ее движении по направляющей.

Образующими l и направляющими m могут быть как простые (прямая, окружность, кривые второго порядка, таблично-заданные кривые и др.), так и структурно-сложные линии, полученные из отрезков указанных простых линий

$$l,\ m \in \{Л_1,\ Л_2,\ \ldots,\ Л_k,\ L_i(Л,\ E)\},$$

где $Л_k$, $L_i(Л, E)$—соответственно простые и структурно-сложные линии.

Структурно-сложные линии описываются графом $L(Л, E)$, множеству вершин $Л$ которого соответствуют простые линии, а множеству дуг E — отношения, характеризующие виды сопряжений этих линий (пересечение, касание) и их взаимное расположение. К структурно-сложным линиям относятся образующие прямоугольных и трапецеидальных пазов, зубьев шестерен, храповиков, шлицев. Направляющие линии сложных линейчатых поверхностей, таких, как «карманы», «колодцы», образованы отрезками прямых, дугами окружностей и другими кривыми.

Увеличение номенклатуры образующих и направляющих линий за счет структурно-сложных линий значительно расширяет возможности кинематического способа и позволяет по единым правилам строить модели как простых, так и типовых и нормализованных поверхностей, а также сложных участков и контуров деталей машин. В целом математическая модель структурно-кинематического способа формообразования поверхностей характеризуется следующими соотношениями:

$$\Phi = l\omega m,$$

$$l,\ m \in \{Л_j,\ L_i(Л,\ E)\},\quad i = 1 \div n_1,\quad j = 1 \div n_2,$$

$$\omega = \omega_0 + \Delta\omega(m),$$

$$B_i = B_0 + \Delta B(m),$$

$$\Phi_i = \Phi(m).$$

Эта модель носит конструктивный характер и отражает как структурные, так и кинематические закономерности формообразования поверхностей. К структурным относятся расчленение поверхностей на образующие и направляющие линии различной сложности, а к кинематическим — задание закона движения образующей и изменений ее размеров и формы.

На основе этой модели построим классификацию поверхностей в зависимости от наличия изменений положения, размеров и формы образующей при ее движении вдоль направляющей. Она представляет собой таблицу, в строках которой фиксируется тип изменений профиля образующей, а в столбцах — положения образующей относительно направляющей. Пересечение строк и столбцов определяет класс поверхностей (рис. 16.4).

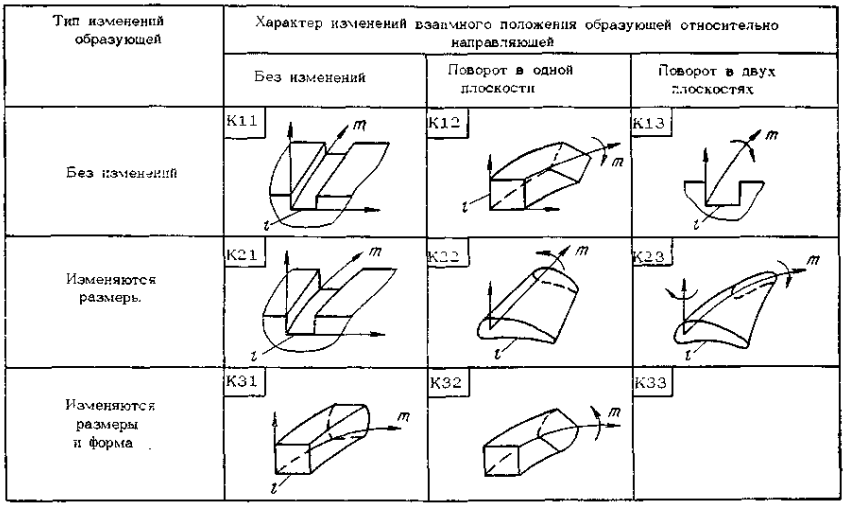


Рис. 16.4. Классификация поверхностей, задаваемых структурно-кинематическим методом

Для поверхностей первого класса характерно отсутствие изменений B_l, Φ_l, Ψ. Во второй класс входят поверхности, размеры профиля которых изменяются по определенным законам при неизменном начальном положении образующей относительно направляющей, например пазы, выступы торцовых муфт, крыло самолета и др.:

$$B = B_0 + \Delta B\,(m),$$

$$\omega = \text{const}.$$

Поверхности третьего класса задаются изменением формы и размеров образующей при ее движении вдоль направляющей $\Phi_l = \Phi(m)$. Отличительной особенностью поверхностей четвертого-шестого классов является изменение положения образующей относительно направляющей в одной плоскости (крутка). Поверхности более высоких классов характеризуются наличием указанных изменений в двух и более плоскостях

$$\omega = \omega_0 + \Delta\omega_x(m) + \Delta\omega_y\,(m).$$

Структурно-кинематический и аналитический способы задания поверхностей взаимосвязаны. Так, переход от аналитического описания к кинематическому для поверхностей вращения и переноса осуществляется путем построения по уравнению поверхности совокупности плоских сечений в трех взаимно перпендикулярных плоскостях. В результате из общего уравнения поверхности

$\Phi(X, Y, Z) = 0$ будут получены уравнения образующих и направляющих линий. Например, при сечении однополостного гиперболоида $x^2/a^2 + y^2/b^2 — z^2/c^z = 1$ двумя взаимно перпендикулярными плоскостями имеем в сечении, перпендикулярном оси поверхности Z = 0, окружность $x^2/q^2 + y^2/b^2 = 1$ (направляющую линию), а в осевом сечении $x = 0$ — гиперболу $y^2/b^2 — z^2/c = 1$ (образующую).

Задание поверхности в виде совокупности отдельных компонент обеспечивает, как будет показано ниже, построение простых, но информативноемких семантических моделей поверхностей деталей машин в проблемно-ориентированных языках САОФ.

Таким образом, структурно-кинематический метод является дальнейшим развитием и формализацией широко известного в начертательной геометрии кинематического метода задания поверхностей.

Математическая модель структурно-кинематического способа образования поверхностей наиболее просто и естественно отображает способы конструирования сложных поверхностей и технологические методы их формообразования на металлорежущих станках с ручным и программным управлением. Так, профиль образующей для одних методов определяет форму режущей части инструмента, а для других — одну из траекторий движения подачи. Аналогичным образом вид направляющей характеризует траекторию главного движения или подачи. Ориентация образующей относительно направляющей определяет взаимное расположение детали и инструмента в процессе обработки, а остальные соотношения— кинематику дополнительных движений подачи, необходимых для получения сложных поверхностей.

Указанная взаимосвязь математической модели поверхности с технологическими закономерностями ее формообразования дает возможность построить фреймы моделей и алгоритмов выбора методов обработки, формы режущей части инструмента и траектории движения инструмента и детали на станках с числовым программным управлением.

Каждый класс фреймов характеризуется большим разнообразием форм поверхностей. В машиностроении и приборостроении наибольшее распространение получили детали машин, образованные поверхностями первого и второго классов. Поверхности остальных классов встречаются в деталях, взаимодействующих со средой (лопатки турбин, крылья самолетов, лопасти гребных винтов и др.). В дальнейшем рассматриваются классификация и методы формализованного описания фреймов поверхностей первых двух классов.

Различные виды типовых и нормализованных поверхностей этих классов образуются за счет большого разнообразия форм образующих при сравнительно небольшом числе направляющих линий. Так, для плоских поверхностей в качестве направляющей используется прямая линия, для поверхностей вращения — окружность, для резьб — винтовая линия и т. д. В связи с этим в основу классификации поверхностей принята следующая иерархия признаков: вид направляющей *m,* расположение образующей относительно направляющей *ψ,* конструктивный тип и форма образующей *k, l,* характер поверхности по длине *q:*

$$\Phi_{п} = \langle\, m,\ \psi,\ k,\ l,\ q\ \rangle\,.$$

По первому признаку поверхности делятся на подклассы: плоские, поверхности вращения, винтовые, линейчатые и фасонные. В отдельный подкласс выделены многозубные поверхности (зубчатые колеса, шлицы, храповики, звездочки), так как они характеризуются дополнительным признаком, описывающим вид поверхности, на которой они расположены (цилиндр, конус, эллипсоид и др.).

Рассмотрим геометрические закономерности образования поверхностей некоторых подклассов и соответствующие им фреймы информационных моделей и методы кодирования. Плоские поверхности образованы поступательным перемещением $\omega_п$ произвольной образующей $l \in L$ в доль прямой $m_{пр}$.

$$ПЛ = (l \in L)\, \omega_п m_{пр}.$$

В зависимости от конструктивного типа образующей линии плоские поверхности разделены на плоскости, уступы, пазы и выступы, окна и многогранники. По форме образующей последние делятся на несколько видов: например, пазы прямоугольного профиля, трапецеидальные, угловые, радиусные, Т-образные и т. д. Плоскости разделены на два вида: прямоугольного и произвольного контура. В последнем случае, помимо кода поверхности, задается описание ее границ. Четвертая ступень классификации определяет характер поверхности по длине (открытые, полуоткрытые, закрытые) и форму ее концов в продольном направлении.

Поверхности вращения образуются вращением $\omega_в$ произвольной образующей *l* вокруг оси *i* и описываются выражением

$$ПВ = (l \in L)\, \omega_в m\,(i).$$

В зависимости от расположения образующей относительно направляющей поверхности вращения делятся на две разновидности: продольные $\psi(l,\ m)=90°$ и торцовые $\psi(l,\ m)=0$. Далее следует деление поверхностей на простые, типовые и

нормализованные. К простым отнесены цилиндр, сфера, тор, эллипсоиды, параболоиды, гиперболоиды и др. В качестве типовых и нормализованных приняты канавки различного профиля (рис. 16.5).

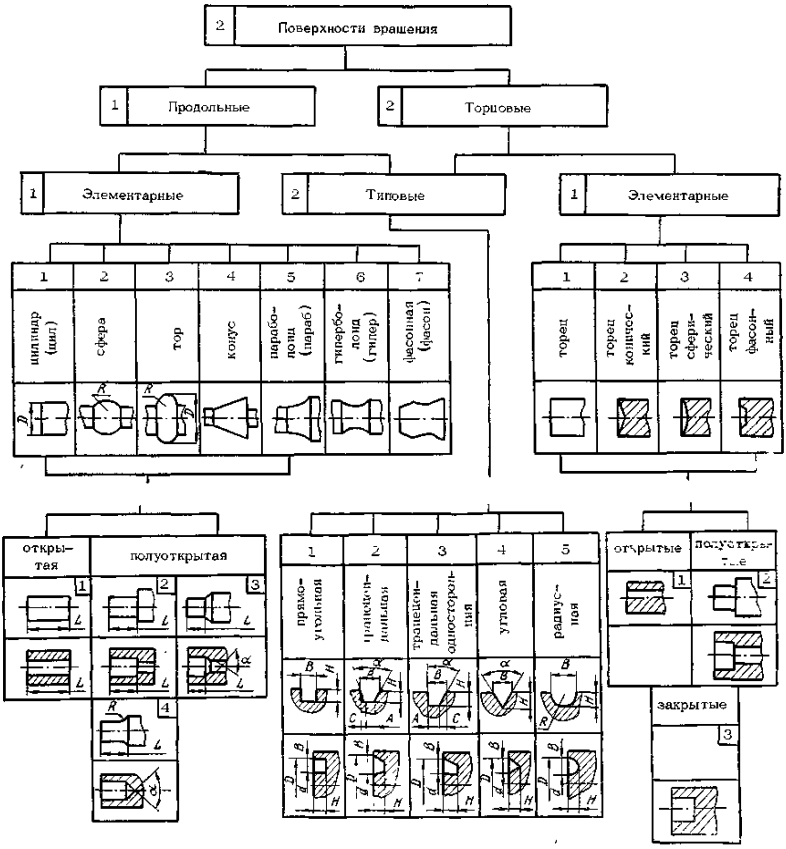


Рис. 16.5. Классификация и кодирование поверхностей вращения

При формализованном описании формы поверхностей каждая ступень классификации может описываться словами естественного языка или цифровым кодом, при цифровом кодировании форма поверхности задается позиционным пятизначным цифровым кодом. Положение каждой цифры кода обозначает один из признаков классификации (подкласс, группу, вид, разновидность), а значение цифры детализирует и уточняет характеристику признака. Перед кодом ставится 0 или 1 в зависимости от расположения поверхности

относительно массы металла, который она охватывает. Цифрой 1 кодируются наружные (шейки валов, выступы, плоскости), а 0 — внутренние поверхности (отверстия, пазы, окна и др.). Например, наружный цилиндрический уступ описывается кодом 121112.

Учитывая требование наглядности, принято обозначать понятия первого и второго уровней классификации, сокращенным названием поверхности на русском языке, а характеристики третьего и четвертого уровней — цифровыми кодами (ЦИЛ 2, КОН 1 и т. д.) . В классификационных таблицах сокращенные наименования типов поверхностей приведены в скобках, а цифровые коды — в верхней части колонок.

16.3. Операции и отношения используемые при образовании фреймов формы детали и размерных сязей ее элементов

16.3.1. Модели формы деталей машин

В детали отдельные ее части связаны между собой конструктивно, размерами и техническими требованиями на точность взаимного расположения в пространстве. Для описания связей вводятся специальные конструктивные операции и отношения, характеризующие специфику процесса проектирования и конструктивно-технологического анализа деталей машин. Это вызвано тем, что существующие алгебраические, логические, теоретико-множественные операции не в полной мере отражают специфику построения моделей реальных конструкций, не удобны для проектирования и описания деталей.

Для описания формы сложных элементов и детали в целом введем конструктивные операции «соединение» и «отсечение», обозначаемые соответственно $\bigcup_q(p)$ и $\backslash q(p)$. Операция «соединение» заключается в построении сложного объекта Q из более простых $П_1$ и $П_2$ с указанием взаимного пространственного расположения соединяемых объектов q ихарактера их сопряжения p:

$$Q = П_1 \bigcup_q (p)\, П_2.$$

Индекс q обозначает взаимное расположение соединяемых объектов. Наиболее распространенными видами взаимного расположения соединяемых объектов являются: по оси —$\bigcup_1$, под прямым углом — $\bigcup_2$, под произвольным углом —$\bigcup_3$, по касательной —$\bigcup_6$, с

параллельными — осями — $\bigcup_7$. На рис. 16.6 показана конструктивная интерпретация различных видов операции соединения.

Вид пространственного расположения		Конструктивные операции и отношения		
		соединение	отсечение	пересечение
Соосное (по оси)		$\Pi_1 \cup_1 \Pi_2$	$\Pi_1 \setminus_1 \Pi_2$	
Под прямым углом	с пересекающимися осями	$\Pi_1 \cup_2 \Pi_2$	$\Pi_1 \setminus_2 \Pi_2$	$\Pi_1 \cap_2 \Pi_2$
	с перекрещивающимися осями	$\Pi_1 \cup_3 \Pi_2$	$\Pi_1 \setminus_3 \Pi_2$	$\Pi_1 \cap_3 \Pi_2$
Под произвольным углом	с пересекающимися осями	$\Pi_1 \cup_4 \Pi_2$	$\Pi_1 \setminus_4 \Pi_2$	$\Pi_1 \cap_4 \Pi_2$
	с перекрещивающимися осями	$\Pi_1 \cup_5 \Pi_2$	$\Pi_1 \setminus_5 \Pi_2$	$\Pi_1 \cap_5 \Pi_2$
По касательной		$\Pi_1 \cup_6 \Pi_2$		$\Pi_1 \cap_6 \Pi_2$
С параллельными осями		$\Pi_1 \cup_7 \Pi_2$	$\Pi_1 \setminus_7 \Pi_2$	$\Pi_1 \cap_7 \Pi_2$

Рис.16.6 Разновидности конструктивных операций и отношений

Характер сопряжения p зависит от сложности соединяемых компонент детали. Для объектов, состоящих непосредственно из простых поверхностей, p обозначает вид переходной поверхности b в сопряжении $П_1$ и $П_2$ (фаски, радиус, канавка):

$$Q = П_1 \cup_q bП_2.$$

При описании формы соосных отверстий или группы наружных поверхностей вращения в характеристике операции p указываются номер оси и направление соединения поверхностей по отношению к ней. Знак «—» ставится, если соединение описывается против оси. Так, форма наружных контуров валов или групп соосных отверстий, образованная последовательным соединением по оси OX любого числа поверхностей вращения, описывается выражением

$$Г = П_1 \cup_1 OXb_1П_2, \ \ldots, \ \cup_1 OXb_kП_k = \bigcup_{i=1}^{k} {}_1OXb_iП_i.$$

При описании детали на уровне объемных компонент p задается совокупностью выражений, которые описывают, сопряжением каких поверхностей объектов Q_1 и Q_2 производится их соединение. Например, вид сопряжения вилки Qi и ступицы рычага Q_2 (см. рис. 16.1) определяется следующими соотношениями:

$$Q_1 \cup_2 (p)\, Q_2,$$

$$p = \begin{cases} П_{11} \cup_6 (П_{22}, \ П_{24}), \\ П_{11} \cup_2 b_2 (П_{21}, \ П_{23}). \end{cases}$$

Здесь p обозначает, что объекты Q_1 и Q_2 соединены так, что боковые плоскости $П_{22}$, $П_{24}$ касательны к цилиндрической ступице, а плоскости $П_{21}$ и $П_{23}$ перпендикулярны к ней. Такое задание p необходимо при автоматизации образования фрейма «Проектирование технологии изготовления заготовок», а при механической обработке — в тех случаях, когда деталь обрабатывается «кругом».

Операция «отсечение» $\setminus_q(p)$ заключается в построении нового более сложного объекта (части детали), который образуется в результате отсечения от исходного объекта $П_1$ объекта $П_2$:

$$Q = П_1 \setminus_q (p)\, П_2,$$

где q — взаимное расположение в пространстве объектов $П_1$ и $П_2$ (рис. 16.6). Характер сопряжения p задается номером оси, по которой производится отсечение, и видом переходной поверхности в сопряжении поверхностей $П_{1y}$ и $П_2$. Знак «—» обозначает, что отсечение производится со стороны, противоположной положительному направлению оси.

Конструктивные операции «соединение» и «отсечение» — формообразующие. Посредством этих операций описывается форма деталей или их участков, образованная из различных простых и структурно-сложных поверхностей. Другим важным фактором, определяющим форму детали, янляется совокупность отношений, задающих характер относительного взаимного расположения поверхностей и объемных элементов:

$$П_1 E_q (П_2, П_3, \ldots, П_t),$$

$$П_{k-1} E_q П_k,$$

E_q — вид отношения; $П_1$, $П_2$, ..., $П_k$— поверхности, связанные отношениями.

Рассматриваются следующие виды взаимного расположения: принадлежность (инцидентность) одного объекта другому — E_o, соосность — E_1, перпендикулярность— E_2, параллельность — E_3, симметричность — E_4, расположение по прямой — E_5, по окружности — E_6 и др. На рис. 16.7 приведены геометрическая интерпретация ука кпшых отношений и способы их описания.

Вид расположения поверхностей		Эскиз	Способ описания на ФТЯ
Принадлежность		$П_1$ P_1 $П_2$	E_0 $(П_1П_2)E_0P_1$
Соосность		$П_1$ $П_2$	E_1 $П_1E_1П_2$
Перпендикулярность		$П_1$ $П_2$	E_2 $П_1E_2П_2$
Параллельность		$П_1$ $П_2$	E_3 $П_1E_3П_2$
Симметричность		z $П_1$ $П_2$	E_4 $П_1E_4(OZ)П_2$
Расположение по прямой	равномерное	0 x M	E_5 $МГ_5O_x$
	неравномерное	0 x M	E_6 $МГ_6OX$
Расположение по окружности	равномерное	OK	E_7 $МЕ_7OK$
	неравномерное	OK	E_8 $МЕ_8OK$
Матричное расположение (равномерное)		y 0 x	E_9 $МЕ_9XOY$

Рис. 16.7. Формализованное описание видов относительного взаимного расположения поверхностей

Отношение $П_1 E_q \{П_2, П_3 ..... П_l\}$ обозначает, что элемент $П_1$ находится в отношении E_q к каждому элементу из множества $\{ П_2, П_3 ..... П_l \}$. Например, равномерное расположение этих элементов по окружности $ОК_1$ описывается выражением $\{ П_2, П_3 ..... П_i \} E_6 ОК_1$. Отношение инцидентности характеризует принадлежность одного элемента другому и будет записано в виде $(П_1, П_2, ...) E_0 П_k$.

Пересечение поверхностей $П_i$ и $П_j$ задается в виде $П_i \pi_q(p) П_j$. Символы q, p имеют тот же смысл, что и в операциях «соединение» и «отсечение». На рис. 16.6 показаны конструктивные разновидности отношения пересечения.

Важным является отношение эквивалентности объектов

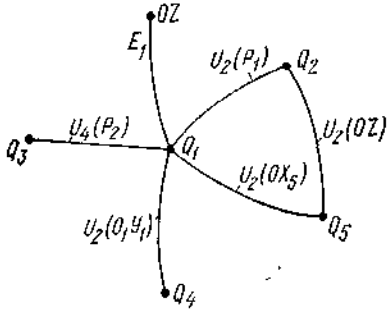
$$П_1 E_{10} \{П_2, П_3, \ldots, П_k\}.$$

Объекты $\{ П_2, П_3 ..... П_k\}$ считаются эквивалентными объекту $П_1$, если их форма, размеры и физико-механические свойства одинаковы. С помощью этого отношения лаконично описываются группы одинаковых поверхностей, так как полностью задаются параметры только одной поверхности, а для остальных указывается, что они эквивалентны рассмотренной.

Конструктивные операции и отношения описывают некоторые очень лажные особенности формы детали, воспринимаемые только при чтении чертежа: например, отсечение, пересечение, соосность, параллельность, перпендикулярность поверхностей и др. Никакими дополнительными количественными параметрами эти особенности на чертеже не задаются. Даже, наоборот, на чертеже указываются только те размеры, которые в сочетании с конструктивными отношениями и операциями однозначно определяют положение поверхности в пространстве. Без указания этих отношений невозможно достаточно точно описать форму детали как машиностроительного объекта, а следовательно, и выбрать правильное конструктивно-технологическое решение при автоматизированном образовании фреймов.

Каждая конструктивная операция и отношение интерпретируются дугой графа, вершинами которой являются связанные отношением объекты Я* и *П]*. Совокупность конструктивных операций и отношений между элементами детали образует граф формы детали *Ф(П, U)* или отдельных ее компонент на различных уровнях расчленения. Вершинам графа соответствуют элементы детали, а дугам — конструктивные операции и отношения, характеризующие формообразующие связи и взаимное расположение элементов. Граф

Ф(П, U) является математической моделью формы детали. На рис. 18 показан граф формы детали на первом уровне расчленения.



Рме 18 Граф формы детали на первом уровне расчленения

Для удобства ввода в ЭВМ граф формы детали может быть представлен поэлементно в виде таблицы связей. В первом и во втором столбцах записываются номера инцидентных вершин, в третьем и четвертом — вид конструктивной операции или отношения и их дополнительные характеристики. Вначале задаются отношения привязки объемных элементов к соответствующим осям, затем следуют отношения и конструктивные операции, посредством которых осуществляется построение конфигурации детали.

16.3.2. Математическая модель размерных связей элементов детали

Компоненты детали любого уровня расчленения связаны между собой размерными связями. Размерная связь между двумя элементами, которыми могут быть точки, линии, поверхности, задается видом размера α, его численной величиной *l* и описывается отношением

$$П_i \alpha l П_j.$$

В дальнейшем линейные размеры будем обозначать буквами *X, Y, Z,* если они параллельны соответствующим координатным осям, и буквой *l* — размеры, произвольно расположенные в пространстве, угловые размеры — φ, а диаметральные — *D* или *R*. Например, $П_i X 50 П_j$— это расстояние 50 мм вдоль оси *OX* между элементами $П_i$ и $П_j$, $П_i \varphi 60 П_j$ — угол 60° между этими элементами.

Совокупность отношений образует размерные цепи детали, которые в зависимости от расположения их элементов в пространстве подразделяются на линейные, плоские и пространственные.

Математически любая размерная цепь описывается графом *P(П, L),* в котором вершинам *П* соответствуют элементы детали (точки, линии и поверхности), а дугам *L* — вид и численная величина размера между элементами. Корнем графа *P(П, L)* конструкторской размерной цепи по любому направлению принимаются поверхности основных баз детали. Вершины следующего уровня соответствуют поверхностям, непосредственно с ними связанным, и т. д. На рис. 16. показано дерево размерных связей элементов детали по оси *OZ* и в плоскости *XOY.*

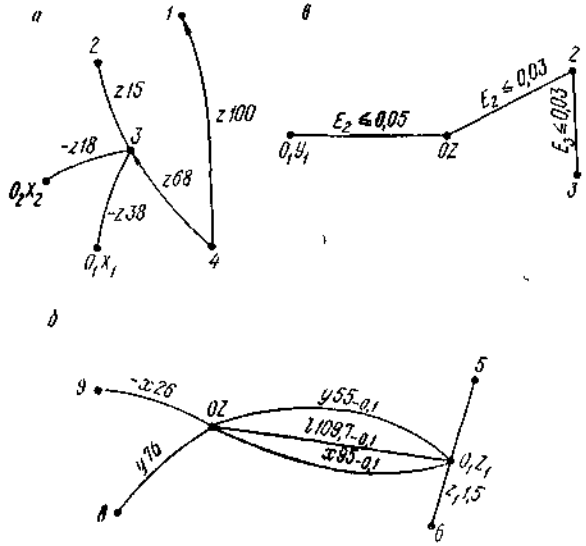


Рис. 16.9. Граф размерных связей элементов детали: *а* — по оси *OZ; б* — в плоскости *XOY; в* — технические требования

Корнем дерева в первом случае является вершина *4* (основная база), а во втором — ось *OZ.* Размер считается со знаком « + », если он отсчитывается от основной базы или промежуточных вершин графа в положительном направлении координатных осей *OX, OY, OZ* и со знаком «—» и противном случае. Для угловых размерных цепей размер принимается со знаком « + », если он отсчитывается от корня графа или промежуточных вершин по часовой стрелке.

На чертежах часто встречаются плоские и пространственные размерные цепи, в которых одни размеры заданы в прямоугольной, а другие в цилиндрической системах координат. В связи с этим моделью плоской размерной цепи служит мультиграф, так как в нем между любыми двумя вершинами может быть две дуги и более, соответствующие линейным, угловым и диаметральным размерам. На

рис. 16.9 показан мультиграф размерных цепей в плоскости *XOY*, который образуется объединением деревьев размерных цепей по различным направлениям:

$$M(\Pi,\ L) = P_x(\Pi,\ X) \cup P_y(\Pi,\ Y) \cup P_\varphi(\Pi,\ \Psi)\ldots$$

Технические требования на точность взаимного расположения поверхностей детали определяются видом отклонения $\overline{E}$, его численной величиной δ и описываются отношением $\Pi_i\,\overline{E}\,\delta\Pi_j$. На рис. 16.7 показано описание видов относительного взаимного расположения поверхностей. Отклонение от заданной точности взаимного расположения обозначается $\overline{E}_i$. Например, неперпендикулярность поверхности Π_2 относительно Π_3 не более 0,03 задается отношением

$$\Pi_2\overline{E}_2\,0{,}03\Pi_3.$$

Так же, как линейные и угловые размерные цепи, совокупность отношений, характеризующих технические требования на точность взаимного расположения, описывается графом *T(П,* Δ). Вершинам графа соответствуют элементы детали (точки, линии, поверхности), а дугам — вид и численная величина технических требований.

16.4. Формализованный входной язык САОФ и правила образования на нем фреймов

Методология системно-структурного анализа и математические модели деталей машин являются основой для построения формализованного входного языка (ФВЯ) САОФ. К ФВЯ САОФ предъявляются следующие основные требования: наличие средств для описания системных характеристик объектов различных уровней сложности: от детали в целом до отдельных поверхностей и линий; универсальность указанных средств, позволяющая эффективно описывать фреймы математических моделей деталей машин любой степени детализации от самой подробной, применяемой при проектировании операционной технологии и управляющих программ для станков с ЧПУ, до наиболее сокращенной при проектировании на основе методов типизации и групповой технологии; простота и естественность построения языковых моделей за счет гибкого сочетания преимуществ свободного и табличного форматов и широкого использования конструкторской терминологии и математической символики.

Выполнение первого и второго требований обеспечивает возможность описания только той информациив фрейме, которая необходима для решения конкретных задач. Два последних требования способствуют ускорению изучения языка и облегчают пользование им. В связи с однотипностью моделей, описывающих системные характеристики объектов различных уровней сложности, синтаксис языка должен определять правила описания множества $M=\{Z_i\}$ параметров детали и различных ее элементов, а также графов формообразующих, размерных структур *Ф(П, U), Р(П, L), Т*(П, Δ) и отношений базирования.

Элементы синтаксиса. Структурными элементами языка являются набор символов, слова, выражения, предложения и массивы. Основные символы — буквы русского и латинского алфавита, цифры, специальные символы клавиатуры входных устройств ЭВМ.

Слова — произвольная последовательность букв и цифр, начинающаяся с буквы. Слово состоит ***не более чем из шести символов*** и является наименьшей синтаксической единицей языка, обладающей смыслом. Словами описываются имена и виды объектов, их свойства и отношения. В языке применяются ключевые слова, модификаторы, идентификаторы и наименования параметров.

Ключевые слова используются для обозначения видов структурных элементов деталей машин различного уровня сложности (ПАЗ 12, ОКНО 3, ЗУБ и т. д.), конструктивных операций и отношений (U1, ×2, ЕЗ и др.). В классификационных таблицах приведены правила образования ключевых слов для описания формы элементов.

Наименования ***параметров*** — это слова, характеризующие вид параметра или физико-механические свойства объекта, например, диаметр —*D,* длина — *L*, термообработка — *ТО* и т. д. В указанных таблицах приведены условные обозначения размерных параметров поверхностей.

Модификаторы применяются для уточнения смысла ключевых слов и параметров в случае их неоднозначности. В ФВЯ используются следующие модификаторы, уточняющие пространственное расположение объектов: большое — *Б,* малое — *М,* справа — *П,* слева — *Л,* верх — *В,* низ — *Н,* внутри — *А,* снаружи — *Р*. Например, длина, наибольшая или наименьшая, обозначается *LБ, LМ.* Расположение объекта *Q* слева от поверхности $П_1$ на расстоянии *L* описывается отношением $П_1LЛQ$.

Идентификаторы *(ИД)* —это слова, используемые для обозначений структурных элементов деталей машин, например, поверхности обозначаются *П1, П2,* ..., группы соосных поверхностей — *Г1, ГЗ,* . . . ,

комплексы — *К1, К2* и т. д. На рис. 16.3 приведены обозначения объектов разных уровней.

Совокупность ключевых слов, модификаторов, идентификаторов и наименований параметров образует ***словарь входного языка***. Для конкретных САОФ этот словарь будет иметь различный состав слов, отражающих специфику образовываемых фреймов.

Следующей по сложности структурной единицей языка являются ***выражения.*** Они представляют собой последовательность слов, отделенных друг от друга разделителями, и предназначены для описания размерных параметров физико-механических свойств и величин технических требований на точность взаимного расположения. В соответствии с этим выделим три типа выражений. Синтаксис выражения *(ВР1)* для описания размерных параметров, используя символику Бэкуса, определяется следующим образом:

$$BP1 :: = \left\langle \begin{matrix}\text{наименование}\\ \text{параметра}\end{matrix} \right\rangle \left\langle , \right\rangle \left\langle \begin{matrix}\text{численное}\\ \text{значение}\end{matrix} \right\rangle \Big| \left\langle \begin{matrix}\text{численное}\\ \text{значение}\end{matrix} \right\rangle \left(\left\langle \begin{matrix}\text{допустимое}\\ \text{отклонение}\end{matrix} \right\rangle \right)$$

$$\left\langle \begin{matrix}\text{допустимое}\\ \text{отклонение}\end{matrix} \right\rangle :: = \left\langle \begin{matrix}\text{верхнее}\\ \text{отклонение}\end{matrix} \right\rangle \Big| \left\langle \begin{matrix}\text{нижнее}\\ \text{отклонение}\end{matrix} \right\rangle \Big| \left\langle \begin{matrix}\text{верхнее}\\ \text{отклонение}\end{matrix} \right\rangle \left\langle , \right\rangle \left\langle \begin{matrix}\text{нижнее}\\ \text{отклонение}\end{matrix} \right\rangle \Big| \left\langle \begin{matrix}\text{обозначение}\\ \text{посадки}\end{matrix} \right\rangle .$$

Примеры выражений:
Д'50,
Д'50 (0,05 —0,15),
В'16(Х3).

Синтаксис выражений второго типа *ВР2,* используемых для описания вида термообработки и физико-механических свойств поверхности, имеет вид

$$BP2 :: = \langle \text{вид термообработки} \rangle \langle ; \rangle \left\langle \begin{matrix}\text{вид параметра}\\ \text{твердости}\end{matrix} \right\rangle \left\langle \begin{matrix}\text{численное}\\ \text{значение}\end{matrix} \right\rangle \left(\left\langle \begin{matrix}\text{допустимое}\\ \text{отклонение}\end{matrix} \right\rangle \right) .$$

Например, закалка поверхности *ТВЧ* до твердости H_{RC} = 48-52 или цементация до твердости H_{RC} =54-60 на глубину *h*=l,2-l,5 будут описаны следующими выражениями:
ТВЧ; *R*С48 (4),
ЦЕМ; h'1.2(0,3); *ЗАК; RC'54(6).*

Синтаксическая структура выражений третьего вида *ВР3,* предназначенная для описания отклонений от точности расположения поверхностей, имеет вид

$$BP3 :: = \langle \text{вид отклонения} \rangle \quad \langle \text{'} \rangle \left\langle \begin{matrix}\text{значение} \\ \text{отклонения}\end{matrix} \right\rangle$$

$$\left\langle \begin{matrix}\text{значение} \\ \text{отклонения}\end{matrix} \right\rangle :: = \langle \text{число} \rangle \mid \langle \text{число} \rangle \ \langle \text{:} \rangle \left\langle \begin{matrix}\text{величина} \\ \text{размера}\end{matrix} \right\rangle .$$

Например, неперпендикулярность 0,03 на длине 250 описывается выражением *Е2* ' 0,03 : 250.

Предложения представляют последовательность слов и параметрических выражений, отделенных друг от друга разделителями. Они предназначены для описания множества параметров $M=\{Z_i\}$ и отдельных отношений, характеризующих формообразующие и размерные связи между элементами. В языке используются предложения двух типов: с частично упорядоченной последовательностью слов и выражений и с жесткой структурой.

Синтаксис предложений первого типа будет следующим:

$$\langle \textit{ИД} \text{ объекта} \rangle = \langle \text{вид объекта} \rangle \quad \langle \text{;} \rangle$$

$$\left\langle \begin{matrix}\text{список выражений} \\ BP1,\ BP2\end{matrix} \right\rangle$$

$$\left\langle \begin{matrix}\text{список} \\ \text{выражений}\end{matrix} \right\rangle :: = \langle \text{выражение} \rangle \mid \left\langle \begin{matrix}\text{список} \\ \text{выражений}\end{matrix} \right\rangle \langle \text{;} \rangle$$

$$\langle \text{выражение} \rangle .$$

Ниже приводятся примеры описания $Z=\{Z_i\}$ детали в целом (см. рис. 16.1), группы соосных отверстий *Г2,* поверхностей *10 и 13.* Общие сведения о детали характеризуются множеством параметров, описывающих технологический класс (вал, втулки, рычаг и т. д.), габаритные размеры — *L, B, H,* массу — *M,* материал — *МТ,* шероховатость — *Ш* поверхностей, не обозначенных на чертеже, размер партии запуска — *N,* вид термообработки — *ТО* и др.

NД = РЫЧ32; Н'100, М'1.74, *МТ'СЧ 18, Ш', N'5.*

Параметры группы соосных отверстий *Г1* будут включать в себя обозначение ее вида, наибольшего и наименьшего диаметров *ДБ, ДМ,* а в ряде случаев и общей длины *L:*

$Г^1$ *= СТУП1; ДБ'* 35, *ДМ'* 20 (0,02).

Параметры отдельных поверхностей, например *П13* и *П10,* описываются на основе обозначений, содержащихся в классификационных таблицах:

П13 = ВЦИЛ1; Д'25 (0,023); *L* Л00; *Ш'7.*

П10 = ПА311; B'20(0.05); *Н'10; Ш'5,* L'48.

Синтаксис предложений с жесткой структурой определяет правила описания отношений < $ИД_i$ объекта >

$\langle ИДj$ объекта $\rangle$ $\left\langle \begin{matrix} \text{вид конструктивной} \\ \text{операции} \end{matrix} \right\rangle$ | $\langle$ вид отношения $\rangle$.

Более подробно синтаксис отношений рассмотрим на примере описания формообразующих, размерных связей и технических требований на точность взаимного расположения. Форма детали и отдельных ее компонент при представлении их в виде совокупности взаимосвязных поверхностей могут быть описаны в виде графа *Ф(П, U)* формообразующих отношений; кортежа < $П_i$, b_i> основных и переходных поверхностей с указанием общей для них конструктивной операции $Г_j = \bigcup_q$, $OX<П_i\ b_i>$; множества элементов, входящих в состав рассматриваемого объекта, если формообразующие связи элементов очевидны и их не требуется описывать; комбинированного способа, т. е. в виде указанных вариантов.

Синтаксис формообразующих отношений соединения и отсечения объектов характеризуется следующим выражением:

< *i* объект > < *j* объект > < вид конструктивной операции > < характер сопряжения >

< характер сопряжения *p* > : : = < № переходной поверхности > |

< № оси > < № переходной поверхности > .

Данная синтаксическая структура отображается таблицей связей (табл. 2), каждая строка которой соответствует одному формообразующему отношению, а каждый столбец— одному из его компонент.

Таблица 2

Формообразующие связи элементов детали

i элемент	*j* элемент	Связь	
		вид	номер переходной поверхности
Q_1	Q_2	$\bigcup_2$	P_1
—	Q_3	$\bigcup_1$	P_2
—	Q_4	$\bigcup_2$	—
—	Q_5	$\bigcup_2$	—
Q_2	Q_5	$\bigcup_2$	—

Синтаксис формы объекта, заданной в виде кортежа связанных конструктивной операцией поверхностей, описывается выражением: ⟨ № объекта ⟩ = ⟨вид конструктивной операции⟩ ⟨ № оси ⟩ (⟨ кортеж $П_i b_i$ ⟩).

Например, форма ступенчатого отверстия *Г2* может быть задана в следующем виде:

ФГ2= ⋃ 1, — *ОХ1 (П8, b1,* П11, *b2, П12).*

Если форма объекта (стороны детали) представляет собой множество поверхностей, соосных групп и комплексов, то его синтаксис задается выражением: <№ объекта *k* уровня> = <множество объектов *k*—1 уровня>.

Например, форма стороны *Н1* описывается перечнем трех объектов: *ФН1 = Г1, ГЗ: П15,* а форма детали в целом в виде множества сторон: *ФД=V1, H1, H2, W1, F1.* В ряде случаев форма стороны задается комбинированным способом, т. е. одна ее часть в виде совокупности отношений, а другая — множества объектов. Синтаксис отношения $П_i \alpha l П_j$, определяющего размерную связь $П_i$ и $П_j$ объектов, описывается следующим выражением:

⟨ *i* объект ⟩ ⟨ *j* объект ⟩ ⟨ вид размера ⟩ ⟨числeнное значение⟩ ⟨ предельные отклонения ⟩. Данному выражению соответствует структура таблицы размерных связей объектов (табл. 3).

Таблица 3

Размерные связи элементов

i элемент	*j* элемент	Связь		
		вид	значение	допуск
1	4	Z_1	100	—
2	3	Z	15	—
3	4	Z	68	—
—	O_1X_1	Z	—38	—
—	O_2X_2	Z	—18	—

Отношение $\{П_i\} \overline{E} \delta\{П_j\}$, описывающее вид и величину технических требований на точность взаимного расположения поверхностей, имеет следующую синтаксическую структуру:< *i* объект > | < множество $П_i$ >,

< j объект > | < множество $П_j$ > < вид отклонения $\overline{E}$ > < численное значение отклонения > .

В табл. 4 приведено описание графа технических требований в соответствии с синтаксисом.

Таблица 4

Технические требования на точность взаимного расположения

i элемент	j элемент	Связь элементов	
		вид	значение
2	OZ	ε_1	0,03
—	3	ε_1	0,003
OZ	O_1Y_1	ε_1	0,05

Массивы — это совокупности однородных предложений, отделенных друг от друга разделителями «/». В соответствии с типами объединяемых предложений выделим **массивы параметрические и отношений**. Массивы первого типа могут быть табличной структуры и свободного формата, второй тип удобно представлять в виде таблиц связей элементов.

Большая трудоемкость и возможные ошибки при ручном кодировании и подготовке данных для ввода в ЭВМ требуют разработки более совершенных средств и методов построения моделей обрабатываемых деталей. Одним из перспективных способов решения этой задачи является создание подсистемы **диалогового построения фреймов информационных моделей деталей** на основе серийно выпускаемых устройств отображения алфавитно-цифровой и графической информации.

Возможность организации диалогового режима для этих целей определяется идентичностью состава системных характеристик и их моделей, описывающих объект любого уровня сложности: от детали в целом до отдельной ее поверхности. Так, фрейм модели объекта произвольной сложности задается **<u>графами формообразующей и размерных структур и множеством параметров</u>** Z

$$Q^k = \langle \Phi(П,\ U)^{k-1},\ P(П,\ L)^{k-1},\ T(П,\ \Delta)^{k-1},\ Z^k \rangle .$$

Для объектов базового уровня указываются код формы $КФ$ и набор параметров Z.

Диалог осуществляется под управлением ЭВМ на основе предварительно разработанной управляющей программы. В качестве основной диалоговой процедуры применяются фреймы запросы от управляющей программы к технологу. Фреймы директивы использу-

ются для корректировки фрагментов полученных описаний. Общая структура диалогового алгоритма построения фрейма информационной модели детали показана на рис.16.10.

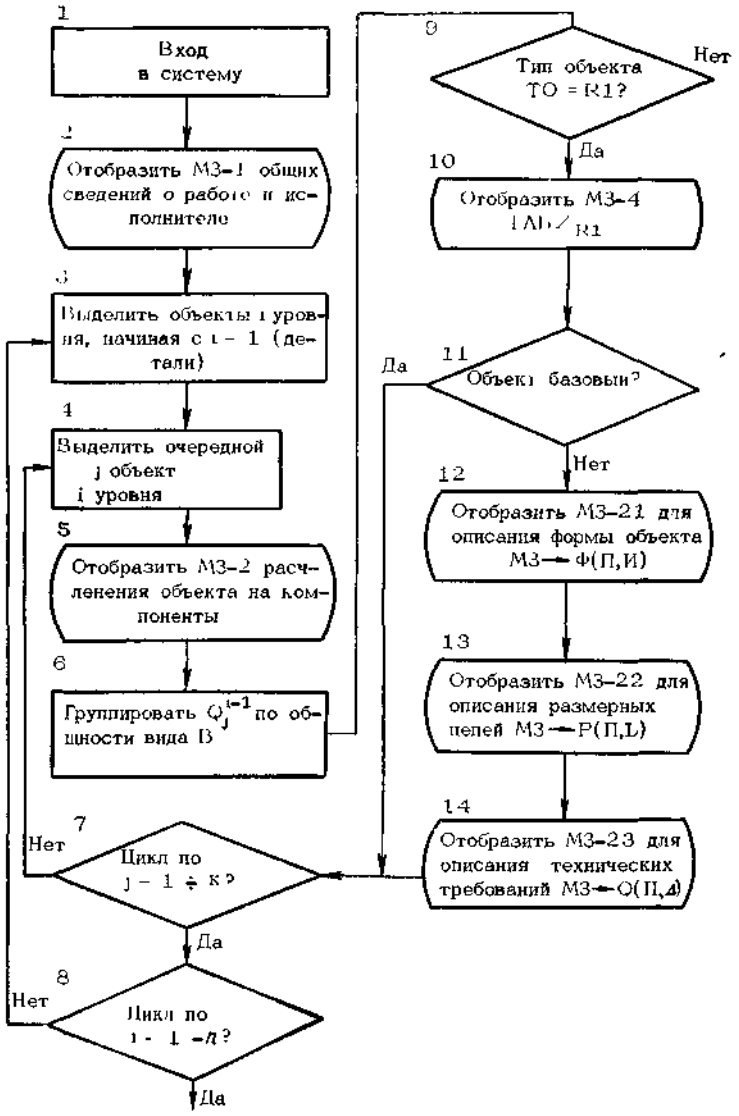


Рис. 16.10 Диалоговый алгоритм построения фрейма информационой модели обрабатываемых деталей

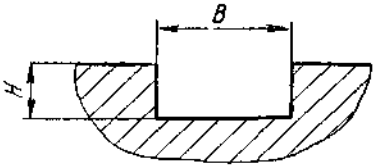
Алгоритм состоит из двух частей. В первой блоками 1—*6* осуществляются процедуры последовательного построения модели структурного состава обрабатываемой детали по уровням от верхнего до уровня простых поверхностей. Во второй при помощи блоков *9—14* производится кодирование общих сведений Z^k, формообразующей и размерной структур $\Phi(\Pi, U)^{k-1}$, $P(\Pi, L)^{k-1}$, $T(\Pi, \Delta)^{k-1}$ объекта *k* уровня расчленения. Операторы цикла *16, 17* повторяют этот процесс для каждого объекта на всех уровнях расчленения.

В начале процесса кодирования диалоговая программа посредством модуля запроса МЗ-1 запрашивает у технолога его табельный номер, вид работы (кодирование, образование фрейма «Проектирование технологии», подготовка управляющих программ и т. д.), номер и класс детали, тип ее технологической модели (ТМД). В САОФ могут быть модели четырех типов: ТМД-1—для образования фреймов процессов классификации, группирования и расцеховки; ТМД-2— для образования фреймов проектирования маршрутной технологии; ТМД-3 и ТМД-4 — для образования фреймов проектирования операционной технологии и подготовки управляющих программ для станков с ЧПУ. С помощью оператора *3* выделяются объекты *i* уровня, начиная с детали в целом и кончая базовыми объектами требуемой степени детализации. Оператор *4* для кодирования выделяет очередной *j* объект *i* уровня. Затем на экран дисплея отображается модуль МЗ-3, который содержит обращение к технологу: расчленить объект Q^i_j на более простые компоненты и ввести их номера и типы *B* в систему, т. е.

$$R : Q^k \rightarrow \{(Q_1^{k-1}, B_1), (Q_2^{k-1}, B_2), \ldots\}.$$

Оператором *6* производится группирование полученных объектов по общности конструктивно-технологических признаков. Посредством логического оператора *9* анализируется тип Q^k объекта и в зависимости от этого на экране дисплея отображается модуль запрос с просьбой заполнить соответствующую таблицу общих сведений *Z* об объекте. Для объектов различных типов состав параметров модуля запроса будет разный, например, таблицы описания общих сведений для плоских, поверхностей вращения, винтовых и др. Группирование объектов по общности их формы позволит создать для каждого класса объектов типовую структуру таблицы параметров. Технологу необходимо их заполнить, пользуясь клавиатурой устройства отображения. Для того чтобы видно было, какие параметры конкретной поверхности необходимо занести в таблицу, в верхней части модуля запроса помещается эскиз формы кодируемого объекта (рис. 16.11).

МЗ. Заполните таблицу сведений
об объекте Q_j



№ п.п	Вид	B		H		L
		значения	допуск	значения	допуск	
П1	ПА 31	10	—	8	0,25	50
П2	ПА 31 1	12	±0,15	10		36

Рис. 16.11. Форма модуля запроса для кодирования общих сведений об элементах детали

Далее, посредством модулей *12, 13, 14* технолог заполняет таблицу формообразующих размерных связей и технических требований на точность взаимного расположения (табл. 2—4).

Логический оператор *11* для базовых объектов исключает модули запроса по кодированию их структуры, так как это не требуется по условиям задачи. В результате под управлением ЭВМ технолог формулирует фрейм технологической модели детали требуемой степени детализации в виде совокупности массивов, с которыми могут взаимодействовать программы, написанные на алгоритмических языках. Применение диалоговых методов построения фреймов информационных моделей деталей позволяет в 2—2,5 раза сократить время кодирования и подготовки данных по сравнению с существующими ручными способами и значительно уменьшить число ошибок

17. Синтез схем процесса образования фреймов

17.1. Некоторые закономерности образования фреймов

Образование фреймов будем рассматривать на примерах образования фреймов в области процессов обработки деталей, принятых в технологии машиностроения.

Первый уровень проектирования процессов обработки и соответствующая ему начальная стадия детализации проектных решений в технологии машиностроения менее всего исследованы и обоснованы. В трудах многих ученых приводятся разные рекомендации по составу и содержанию проектных задач, решаемых на этой стадии. Так Д. В. Чарнко в начальный период выделяет разработку основных направлений проектирования технологических процессов, с учетом которых в последующем можно формировать различные варианты технологии изготовления конкретных деталей. Обращается внимание на то, что при определении основных направлений следует больше освещать вопросы выбора видов заготовок, термических операций и методов обработки основных поверхностей, при этом меньше уделять внимания выбору оборудования и структуре операций, так как эти задачи решаются при проектировании маршрута обработки и операций.

Несколько иную точку зрения на степень детализации технологических решений на первой стадии проектирования высказывает В. П Фираго. К начальной стадии проектирования он относит составление плана процесса, включающего формирование операций и определение их последовательности, выбор оборудования и технологических баз, разработку операционных эскизов, увязку выбранных операций с этапами технологического процесса и др.

Если начальный период проектирования у Д. В. Чарнко связан с определением основных направлений разработки технологического процесса и характеризуется наименьшей степенью детализации проектных решений, то план процесса в трактовке В. П. Фираго включает в себя решение всех основных задач по формированию маршрута обработки детали. При традиционных методах проектирования, как отмечалось выше, формирование основных

направлений или разработка плана процесса производятся технологом на основе общих рекомендаций и имеющегося опыта. Формализация процесса технологического проектирования и в том и другом случае затруднена. В первом это связано с недостаточной определенностью и четкостью структуры основных направлений, а во втором — вследствие недостаточности исходных и справочно-нормативных данных для формального решения всех задач, входящих в план процесса. В связи с этим возникает задача обоснования степени детализации технологических решений на первом уровне проектирования и определения системных характеристик принципиальной схемы процесса. При этом необходимо учитывать, что степень детализации технологических решений в принципиальной схеме должна быть такой, которую можно получить на основе имеющихся к началу проектирования исходных данных и технических ограничений. Исходными технологическими понятиями на данном уровне проектирования являются методы и объекты обработки (поверхности и объемные элементы деталей). Объекты обработки характеризуются начальным C_0, промежуточными C_i и конечным C_k состояниями. Между методами и объектами обработки существует связь, описываемая функцией метода

$$\mu_i : C_{i-1} \to C_i.$$

Указанное отношение задает технологическое преобразование с помощью метода μ_i объекта обработки из предшествующего состояния C_{i-1} в C_i с более высокими значениями точностных параметров или физико-механических свойств. Поэтому в качестве исходных аспектов будут выбраны такие системные принципы и технологические закономерности, которые оказывают влияние на выбор методов обработки, порядок их выполнения и параметры состояний обрабатываемых поверхностей деталей и их элементов.

Одним из важных системных принципов, оказывающих влияние на выбор элементов процесса обработки, является принцип технологической совместимости. Он определяет такую общность элементов технологической системы по совокупности структурных и функциональных свойств, благодаря которой элементы объединяются в систему фреймов и обеспечивается ее функционирование в соответствии с заданными требованиями. На основе этого принципа сформулируем следующий исходный аспект.

Аспект 1. Методы и объекты обработки технологически совместимы, если форма объекта, его размерные, точностные параметры и физико-механические свойства могут быть получены данным методом.

Следующих два исходных аспекта отражают погрешности, возникающие в результате механической обработки, которые

необходимо учитывать при образовании фреймов технологического процесса.

Аспект 2. При механической обработке ввиду податливости элементов СПИД погрешности предшествующей обработки или заготовки ε_{i-1} устраняются неполностью, а в уменьшенном и измененном виде передаются детали на выполняемой операции $\varepsilon_i=\eta\varepsilon_{i-1}$. Происходит уточнение детали по различным характеристикам точности.

Аспект 3. При механической обработке в результате снятия припусков и напусков происходит деформация детали от перераспределения внутренних напряжений и, как следствие этого возникают погрешности формы и взаимного расположения элементов детали.

Погрешности ε_σ, связанные с указанным видом деформации, особенно велики при обдирке и черновой обработке нежестких заготовок, когда снимаются большие припуски. При последующей получистовой и чистовой обработке они тем меньше, чем меньше величины снимаемых припусков. В полной мере эта деформация проявляется спустя некоторое время после обработки. Таким образом, погрешности ε_σ зависят от жесткости детали $J_д$, величин снимаемых припусков η и времени, прошедшего после обработки $\varepsilon_\sigma =(J_д, h, t)$.

Следующие три оспекта отражают явления, происходящие при обработке детали термическими методами, которые необходимо учитывать при построении принципиальной схемы процесса.

Аспект 4. Погрешности детали ε^Σ по различным характеристикам точности после термообработки увеличиваются по сравнению с погрешностями предшествующей механической обработки и определяются векторной суммой

$$\varepsilon^\Sigma = \bar{\varepsilon}^{м}_{i-1} + \varepsilon^{т}_{i} .$$

Передаточное отношение погрешностей, полученных после термообработки, к погрешностям предшествующей механической обработки вычисляется по формуле

$$\eta^{т}_{i,i-1} = \frac{\bar{\varepsilon}^{м}_{i-1} + \varepsilon^{т}_{i}}{\varepsilon^{м}_{i-1}} = 1 + \frac{\varepsilon^{т}_{i}}{\varepsilon^{м}_{i-1}} .$$

Оно всегда больше единицы. Деталь вследствие возникающих при термообработке деформаций $\varepsilon^{т}_{i}$ теряет ранее полученную точность.

Аспект 5. При термической и химико-термической обработке физико-механические свойства материала изменяются по глубине r сечения детали $H_{RC} = f(r)$. В связи с этим требуемая твердость H_{RC} и другие характеристики материала могут быть получены только до определенной глубины r_m. Например, при азотировании глубина слоя составляет $r\approx0{,}3$-$0{,}5$ мм, а зона наибольшей твердости распространяется на глубину $r_m\approx0{,}1$.

Аспект 6.. В результате термического и химико-термического воздействия, проводимого с целью получения высоких физико-технических свойств, обрабатываемость материала механическими методами ухудшается.

Принятые аспекты отражают наиболее общие и бесспорные функциональные свойства отношения $\mu_i : C_{i-1} \rightarrow C_i$, установленные технологической наукой и практикой и вытекающие из принципа технологической совместимости. Рассмотренные аспекты независимы друг от друга. Ни один из них не может быть получен из других путем логического вывода. Исходные аспекты не противоречивы. Это значит, что технологический факт, зафиксированный в одном аспекте, не противоречит другим фактам, полученным из других аспектах.

Совокупность исходных аспектов не является полной и замкнутой. При внедрении новых методов обработки и материалов, оказывающих существенное влияние на построение технологического процесса, к имеющимся исходным аспектам необходимо добавить новые. Они должны отображать закономерности новых методов, которые следует учитывать при построении технологического процесса. На основе аспектов 1—6 сформулируем ряд утверждений, определяющих выбор методов обработки, виды и взаимное расположение механической и термической обработки, обеспечивающих наиболее эффективное достижение требуемых точностных параметров и физико-механических свойств детали и отдельных ее элементов.

Выбор допустимоге метода обработки поверхности осуществляется сравнением его функциональных возможностей, т. е. параметров $C_i^{\text{М}}$ состояния, с требуемыми значениями точностных параметров и физико-механических свойств обрабатываемой поверхности. На основе аспекта 1 сформулируем утверждение, определяющее вид и структуру фреймов алгоритмов выбора допустимых методов обработки.

Утверждение 1. Допустимыми из заданного множества методов обработки $M = \{M_i\}$ будут такие методы, которые обеспечивают получение требуемых форм обрабатываемой поверхности $\Phi^{\text{п}}$ точности ее размеров $\tau^{\text{п}}$ и взаимного расположения δ_{ij}, шероховатости $Ш^{\text{п}}$ и физико-механиче-ких свойств $\text{Н}^{\text{п}}_{\text{RC}}$, т.е.

$$M_{\text{доп}} \in \{\text{M}_i\},$$

если

$$\Phi^{п} = \in \{\Phi_i^{м}\},$$
$$\tau^{п} = (\tau_{эк} \div \tau_{доп})^{м}, \qquad (5)$$
$$Ш^{п} = (Ш_{эк} \div Ш_{доп})^{м},$$
$$\delta_{ij}^{п} = (\delta_{эк} \div \delta_{доп})^{м},$$
$$H_{RC}^{п} = (H_{RC}^{1} - H_{RC}^{2})^{м},$$
$$H_{RC}(i-1) \leqslant H_{RC,доп}^{м}.$$

Первое соотношение (5) обозначает, что форма обрабатываемой поверхности должна быть элементом множества форм, получаемых с помощью данного метода. Соотношения два-пять указывают, что значения точностных параметров и физико-механических свойств обрабатываемой поверхности должны находиться в пределах от экономически целесообразных $\tau_{эк}$, $\delta_{эк}$, $Ш_{эк}$ до технически допустимых значений $\tau_{доп}$, $\delta_{доп}$, $Ш_{доп}$ для данного метода обработки. Величины экономических и технически достижимых значений τ, δ, $Ш$, H_{RC} получены на основе обобщения опыта изготовления деталей и содержатся в заводских нормалях и справочной литературе.

Неравенство шесть системы (5) характеризует величины допустимых для данного метода значений физико-механических свойств поверхности *(HB, H_{RC})*, полученных в результате предшествующей обработки. Эти значения должны быть такими, чтобы обеспечить нормальные условия резания и эффективную обработку поверхности выбранным методом. Соотношения (5) образуют математическую модель выбора допустимого метода обработки поверхности. Она служит основой для построения фрейиов алгоритмов решения указанной задачи.

Следующее утверждение определяет виды и последовательность методов обработки, необходимых для получения заданных чертежом параметров обрабатываемой поверхности из состояния заготовки.

Утверждение 2. Для получения заданных точностных параметров поверхности, исходя из учета влияния податливости системы обработки СПИД, виды, последовательность и количество переходов определяются следующими соотношениями:

$$\begin{aligned} &M_1 : C_0(\varepsilon_0) \to C_1(\varepsilon_1), \\ &M_2 : C_1(\varepsilon_1) \to C_r(\varepsilon_2), \\ &\ldots\ldots\ldots \\ &M_i : C_{i-1}(\varepsilon_{i-1}) \to C_i(\varepsilon_i), \\ &\ldots\ldots\ldots \\ &M_k : C_{k-1}(\varepsilon_{k-1}) \to C_k(\varepsilon_k). \end{aligned} \qquad (6)$$

Доказательство этого утверждения производится на основе аспекта 1. Первое соотношение (6) показывает, что из заготовки $C_0(\varepsilon_0)$ с точностными параметрами $\varepsilon_0 = (\tau_0, \delta_0, Ш_o)$ с помощью метода M_1 достигается промежуточное состояние $C_1(\varepsilon_1)$, точностные параметры которого в силу аспекта 1 будут более высокими, чем у заготовки $\varepsilon_0 > \varepsilon_{1i} > \varepsilon_k$. Обрабатывая эту поверхность вторично методом M_2, получаем новое промежуточное состояние $C_2(\varepsilon_2)$, характеризующееся более высокой точностью, чем $C_1(\varepsilon_1)$. Этот процесс продолжается до тех пор, пока не будет получена поверхность требуемой точности $\varepsilon_к \leq \varepsilon_{чер}$. В частном случае для достижения заданной точности поверхности может потребоваться один переход. Для получения требуемых физико-механических свойств поверхности в соотношения (6) включаются преобразования поверхности или детали в целом термическими или химико-термическими методами

$$M_i^т : C_{i-1}(\varepsilon_{i-1}^м) \to C_i^т(\varepsilon_i^т).$$

Последовательность методов обработки, необходимых для достижения требуемых размерных, точностных параметров и физико-механических свойств поверхности, будем называть элементарным процессом, или маршрутом обработки поверхности *МОП.* Представив преобразование $M_i : C_{i-1}(\varepsilon_{i-1}) \to C_i(\varepsilon_i)$ в виде отношений $C_{i-1}(\varepsilon_{i-1})\, M_i C_i(\varepsilon_i)$, получим функциональную модель *МОП* в виде последовательности преобразований поверхности из начального состояния C_0 в конечное $C_к$:

$$МОП = C_0(\varepsilon_0)\, M_1 C_1(\varepsilon_1) \ldots M_к C_к(\varepsilon_к).$$

В *МОП* каждому виду технологического преобразования $C_{i-1}(\varepsilon_{i-1})\, M_i C_i(\varepsilon_i)$ соответствует погрешность обработки, определяемая по формуле $\varepsilon_i = \eta, \ \varepsilon_{i-1}$. Подставив вместо ε_{i-1} его значение $\varepsilon_{i-1} = \eta_{i-1}\varepsilon_{i-2}$, а вместо ε_{i-2} соответственно $\varepsilon_{i-2} = \eta_{i-2}\varepsilon_{i-3}$, и так до последней операции, получим известную

формулу для определения результирующей погрешности на последнем переходе *МОП*

$$\varepsilon_к = \eta_к \eta_{к-1} \cdots \eta_1 \varepsilon_0.$$

Из этого утверждения вытекает

Следствие 1. Чем выше точностные параметры обрабатываемой поверхности и ее физико-механичекие свойства, тем большее число переходов при прочих равных условиях (материал детали, жесткость СПИД и др.) необходимо для их достижения. В связи с этим маршруты обработки главных поверхностей ввиду их высокой точности будут иметь большее число переходов, чем маршруты обработки вспомогательных поверхностей.

Положение термообработки «улучшение» в технологическом процессе определяется на основе следующего утверждения.

Утверждение 3. Если величина прокаливания $r_т$ меньше минимального размера B_i наибольшего S_{max} по величине сечения заготовки $r_Т < \min B_i(S_{max})$, то термообработку «улучшение» следует выполнять после предварительной механической обработки, когда основные припуски и напуски будут сняты; если $r_Т \geq \min B_i(S_{max})$, то — в заготовке перед механической обработкой.

В первом случае будет достигнута наибольшая глубина прокаливания детали, во втором обеспечивается прокаливаемость заготовки по всему сечению, а процесс механической обработки не прерывается термической операцией.

Утверждение 4. Если суммарная погрешность детали от предшествующей механической и термической обработки не превышает величин, допускаемых по чертежу $\varepsilon_{i-1}^{м} + \varepsilon_i^{т} \leq \varepsilon_{чер}$, то термообработка, проводимая для получения высоких физико-механических свойств, назначается в конце технологического процесса. В противном случае она должна предшествовать этапу окончательной механической обработки.

В первом случае, исходя из аспекта посылки 5, расположение термообработки в конце технологического процесса наиболее экономично и не вызовет увеличения трудоемкости механических операций, так как они не будут прерываться операциями термической обработки, выполняемыми, как правило, в отдельных цехах. Такой порядок применяется для неточных и жестких деталей или для неответственных поверхностей в деталях высокой точности. Расположение термической обработки перед завершающими этапами позволит уменьшить объем механической обработки труднообрабатываемого закаленного материала до величин,

минимально необходимых для получения требуемых точностных параметров детали.

На основе приведенных аспектов и утверждений сформулируем утверждение, определяющее фрейм структуры принципиальной схемы технологического процесса.

Утверждение 5. При недостаточной жесткости детали и необходимости ее термообработки для получения высоких точностных параметров и физико-механических свойств фрейм технологического процесса изготовления детали разделяется на ряд последовательных этапов (подфреймов), в каждый из которых включаются однородные по характеру и точности методы обработки различных поверхностей и детали в целом.

Доказательство этого утверждения основывается на аспектах 3, 6 и утверждениях 1—4. В соответствии с выбором поверхности в качестве базового структурного элемента детали общая функция технологического процесса расчленяется на совокупность подфункций элементарных процессов обработки отдельных поверхностей

$\Phi \rightarrow \{ f_1, f_2, \ldots, f_n \}$, описываемых следующими преобразованиями:

$$\begin{gathered} \lambda_{\text{п}1} : З_1(\varepsilon_1^0) \rightarrow П_1(\varepsilon_1^k), \\ \lambda_{\text{п}2} : З_2'(\varepsilon_2^0) \rightarrow П_2'(\varepsilon_2^k), \\ \ldots\ldots\ldots \\ \lambda_{\text{п}n} З_n(\varepsilon_n^0) \rightarrow П_n(\varepsilon_n^k), \end{gathered} \qquad \textbf{(7)}$$

где $\lambda_{\text{п}1}, \lambda_{\text{п}2}, \ldots, \lambda_{\text{п}n}$— элементарные процессы обработки поверхностей детали; $З_1(\varepsilon_1^0), \ldots, З_n(\varepsilon_n^0)$ — параметры обрабатываемых поверхностей в заготовке; $П_1(\varepsilon_1^k), \ldots$ $\ldots, П_n(\varepsilon_n^k)$ — точностные параметры и физико-механические свойства поверхности по чертежу.

На основании утверждения 2 общие выражения функций элементарных процессов запишем в развернутом виде как функциональные модели маршрутов обработки поверхностей детали. В результате соотношения (7) примут вид

$$\begin{gathered} З_1(\varepsilon_1^0)\, \text{М}_1 П_1(\varepsilon_1^1) \ldots \text{М}_l П_1(\varepsilon_1^k), \\ З_2(\varepsilon_2^0)\, \text{М}_2 П_2(\varepsilon_2^1) \ldots \text{М}_g П_2(\varepsilon_2^k), \\ \ldots\ldots\ldots\ldots \\ З_n(\varepsilon^0)\, \text{М}_c П_n(\varepsilon_n^1) \ldots \text{М}_t П_n(\varepsilon_n^k). \end{gathered}$$

Основываясь на аспекте 3, для устранения погрешностей, вызванных деформацией детали от перераспределения внутренних напряжений

при снятии больших припусков и напусков, необходим разрыв во времени между черновой, чистовой и окончательной обработкой поверхностей детали. За это время указанные деформации успеют проявиться и при последующей обработке будут устранены. В соответствии с разделением *МОП* на переходы различной точности механическая обработка высокоточных деталей делится на черновую, чистовую, окончательную и отделочную.

Кроме того, исходя из утверждений 3, 4, процесс механической обработки может многократно прерываться термическими операциями. На основании указанных соображений разделим полученное множество *МОП* по вертикали на непересекающиеся подмножества (этапы) так, чтобы методы обработки, общие по классу (заготовительные, механические, термические) и виду, в пределах одного класса входили в один этап

$$Э_i = \{\mathbf{M}_1 П_1(\varepsilon_1^1),\ \mathbf{M}_2 П_2(\varepsilon_2^1) \ldots \mathbf{M}_c П_n(\varepsilon_n^1)\}. \quad (8)$$

В результате такого расчленения соотношения (7) будут преобразованы в табл. 5.

Таблица 5

Этапы технологического процесса

Номера этапов	Наименование этапов	Назначение и характеристика этапов	Номера поверхностей детали			
			$П_1$	$П_2$		$П_n$
Э1	Заготовительный	Получение заготовки и ее термообработка		$\lambda_1 : C_м^0 \to C_3^1$		
Э2	Черновой	Съем лишних напусков и припусков		$\lambda_2 : C_3^1 \to C_D^2$		
Э3	Термический I	Термообработка—«улучшение», «старение»		$\lambda_3 : C_D^2 \to C_D^3$		
Э4	Получистовой I	Точность 11÷13 кв., шероховатость R_a=2,5	$\mu_1 C_1^4$	$\mu_2 C_2^4$		$\mu_1 C_n^4$
Э5	Термический II	Цементация		$\lambda_т^5 : C_D^4 \to C_D^5$		
Э6	Получистовой II	Съем припуска для предохранения от цементации	$\mu_1 C_1^6$			
Э7	Термический III	Закалка, «улучшение»		$\lambda_т^7 : C_D^6 \to C_D^7$		
Э8	Чистовой I	Точность 6, 7 кв., шероховатость R_a=1,25	$\mu_3 C_1^8$	$\mu_3 C_2^8$		$\mu_2 C_n^8$
Э9	Термический IV	Азотирование, «старение»		$\lambda_т^9 : C_D^8 \to C_D^9$		
Э10	Чистовой II	Съем припуска для предохранения от азотирования		$\mu_3 C_2^{10}$		
Э11	Чистовой III	Точность ⩾5 кв., шероховатость R_a=0,16	$\mu_4 C_1^{11}$	—		$\mu_4 C_n^{11}$
Э12	Гальванический	Хромирование, никелирование и др.		$\lambda_{хр}^{12} : C_D^{11} \to C_D^{12}$		
Э13	Доводочный	Шероховатость R_a=0,04	$\mu_6 C_1^{13}$	—		$\mu_5 C_n^{13}$

Каждая строка таблицы соответствует этапу технологического процесса, а каждый столбец — маршруту обработки отдельной поверхности. Одни этапы относятся ко всей детали (51, 53), другие — к совокупности отдельных поверхностей.

В выражении (8) совокупность методов обработки $\lambda = \{M_1, M_2 ..... M_c\}$ обобщенно характеризует вид технологического преобразования λ в i этапе, а множество состояний отдельных поверхностей

$$C_{эi} = \{\Pi_1(\varepsilon_1^!),\ \Pi_2(\varepsilon_2^!) \ldots \quad \ldots \Pi_n(\varepsilon_n^!)$$

— состояние детали, полученное в результате ее обработки в $Э_i$ этапе. В связи с этим функция $Э_i$ этапа определяется как преобразование детали из состояния $C_{э,\,i-1}$, полученного на предыдущем i — 1 этапе, в состояние $C_{э,\,i}$ на выполняемом этапе $C_{э,\,i-1}\lambda_i C_{э,\,i}$.

Этапы механгческой обработки в определенном порядке чередуются с термическими и образуют фрейм принципиальной схемы технологического процесса. Таким образом, основным структурным элементом фрейма принципиальной схемы является этап обработки.

При обработке поверхностей одного вида (например, плоскостей) применяются различные методы: строгание, фрезерование, шлифование, шабрение и др. В связи с этим количество вариантов *МОП* получается довольно большим, а их формирование осуществляется на основе следующего утверждения.

Утверждение 6. Если для обработки поверхности определенного вида $\Pi_к$ и точности $\varepsilon = (\tau, \delta, Ш)$ могут быть применены различные методы обработки $M = \{M_q\}$, то возможные варианты фреймов маршрутов обработки такой поверхности образуют граф *Н (С, М),* вершинам которого соответствуют состояния поверхности, а дугам — множество методов обработки, посредством которых эти состояния достигаются.

Функция метода обработки как преобразования М поверхности из одного состояния C_{j-i} в другое C_j, более точное, может быть интерпретирована дугой графа $C_{j-1}MC_j$ с вершинами C_{j-1} и C_j. Если для достижения точностных параметров C_j состояния поверхности применяются различные методы обработки $M_1, M_2, ... M_b$, то совокупность их функций образует подграф с вершиной C_j. Каждое из состояний $C_{j-1,\ q}$ j—1 этапа аналогичным образом отображается в множество состояний j—2 этапа и т. д., до тех пор, пока предшествующие состояния не будут совпадать с параметрами заготовки или начерно обработанной детали.

В результате получим граф фрейма возможных вариантов маршрута обработки поверхности определенного класса (рис. 17.1), в некоторые вершины которого может входить несколько дуг (вершина C_{2n}).

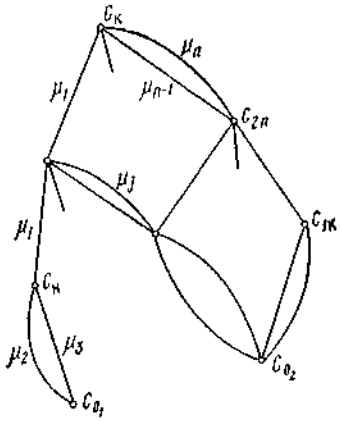


Рис.17.1. Гдаф возможных вариантов маршрутов обработки поверхности

Это означает, что два метода обработки характеризуются одинаковыми точностными параметрами промежуточных состояний. Любой путь на графе из C_o в C_K будет одним из вариантов *МОП.*

Состав и количество этапов (подфреймов), связанных с химико-термической обработкой *(ХТО),* зависят от ее вида и способа защиты поверхностей, предохраняемых от *ХТО.* Так, для цементации справедливо следующее утверждение.

Утверждение 7. Если предохранение от цементации осуществляется припуском *h,* то *ХТО* разделяется на два этапа (подфрейма): цементация $Э_ц$ и закалка $Э_{зак}$, между которыми вводится получистовой этап $Э_{пч}$ для снятия припусков с поверхностей, предохраняемых от цементации. При других способах защиты (омеднение, обмазка) или цементации детали «крутом» $Э_{пч}$ не требуется, а операции цементации и закалки объединяются в один этап (подфрейм) $Э_{хто}$:

$$ХТО = \begin{cases} Э_ц Э_{пч} Э_{цем}, & \text{если } P_h = 1, \\ Э_{2го}, & \text{если } P_h = 0, \end{cases}$$

где P_h — логический оператор, проверяющий способ защиты поверхностей от цементации.

Положение этапов (подфреймов) химико-термической обработки в фрейме принципиальной схемы технологического процесса определяется исходя из необходимости сохранения поверхностного слоя с высокими физико-механическими свойствами и трудностью его механической обработки.

На основе утверждения 4 *ХТО* должна предшествовать окончательному этапу $Э_{хто} < Э_{ок}$, в который выносится обработка точных поверхностей с малыми припусками. Остальные поверхности, если их $\varepsilon^{м}_{i-1} + \varepsilon^{т}_{i} \leqslant \varepsilon_{чер}$, должны быть обработаны окончательно в чистовом этапе, предшествующем *ХТО.* Следующее утверждение формулирует требования к точности и характеру обработки детали в этапах (полфреймах), предшествующих *ХТО.*

Утверждение 8. Для получения при *ХТО* поверхностного слоя с высокими и стабильными значениями физико-механических свойств необходимо:

а) в этапе (подфрейме), предшествующем *ХТО,* обеспечить высокую точность поверхностей, подлежащих цементации или азотированию, а также поверхностей, принимаемых за базы при последующей механической обработке:

$$\varepsilon^{м}_{i-1} \leqslant br_m - (\varepsilon^{т}_{i} + \varepsilon^{у}_{i+1});$$

б) для уменьшения деформаций от цементации нежесткие детали должны проходить правку $A_{пр}$ после цементации или стабилизирующую термообработку $Э_{то}$ после чернового этапа $Э_{чер}$:

$$S_{ХТО} = \begin{cases} Э_{цем} \cdot A_{пр} \cdot Э_{зак}, & \text{если} \quad P_k = 1, \\ Э_{чер} \cdot Э_{то} \cdot Э_{пч} \cdot Э_{цем}, & \text{если} \quad P_k = 0, \end{cases}$$

где P_k — оператор, определяющий целесообразность правки детали после цементации; $Э_{цем.}$, $Э_{зак}$, $Э_{пч}$ — этапы (подфреймы) цементации, закалки и получистовой обработки.

Доказательство первой части утверждения осуществляется на основе исходных аспектов 5, 6. Для сохранения наиболее износоустойчивой части поверхностного слоя припуск на последующую механическую обработку должен быть стабильным и не превышать определенной части r_m, т. е. $h \leq br_m$. Так, при азотировании этот припуск составляет 0,05—0,06 мм. В то же время суммарная погрешность предшествующей механической, химико-термической обработки и погрешности установки $\varepsilon^{у}_{i+1}$ при последующих механических операциях $\varepsilon^{\Sigma} = \varepsilon^{м}_{i-1} + \varepsilon^{т}_{i} + \varepsilon^{у}_{i+1}$ должна компенсироваться этой частью припуска:

$$br_m \geqslant \varepsilon^{м}_{i-1} + \varepsilon^{т}_{i} + \varepsilon^{у}_{i+1}.$$

Отсюда вытекает, что точность предшествующей механической обработки должна быть достаточно высокой и определяться из соотношения

$$\varepsilon^{м}_{i-1} \leqslant br_m - (\varepsilon^{т}_{i} + \varepsilon^{у}_{i+1}).$$

Так, поверхности, подлежащие азотированию, предварительно шлифуют с целью обеспечения небольших величин погрешностей предшествующей механической обработки, а перед азотированием обеспечивают хорошие установочные базы для последующей обработки.

В зависимости от важности выполняемых функций поверхности и элементы деталей разделяются на главные и вспомогательные. Главные поверхности характеризуются высокими значениями точностных параметров и физико-механических свойств. Точностные характеристики вспомогательных поверхностей невысокие. Естественно, что состав и количество этапов (подфреймов) для их обработки требуются разные. В связи с этим сформулируем следующее упюрждение.

Утверждение 9. Состав и порядок этапов (подфреймов) обработки детали, т. е. фрейм принципиальнуой схемы, определяют виды, точностные параметры и физико-механические свойстза главных поверхностей. Обработка второстепенных поверхностей производится в получистовых этапах и не оказывает существенного влияния на характер принципиальной схемы процесса.

Из следствия 1 видно, что для обработки главных поверхностей, характеризующихся высокими точностными параметрами и физико-механическими свойствами, необходимо выполнить большее число механических и термических операций, чем при обработке второстепенных. В связи с этим содержание матрицы этапов (табл. 5) будут определять маршруты обработки главных поверхностей. Маршруты обработки второстепенных поверхностей ввиду их невысокой точности будут содержать один, два перехода, выполняемых в одном из чистовых этапов.

На основе приведенных утверждений рассмотрим состав, последовательность и характеристики этапов (подфреймов) технологического процесса изготовления деталей различных классов.

17.2. Фрейм функциональной структуры принципиальной схемы технологического процесса

17.2.1. Этапы (подфреймы) технологического процесса

В целом функциональная структура принципиальной схемы процесса характеризуется последовательностью преобразований

обрабатываемой детали из начального состояния в заготовительном этапе в состояния $C_{э1}$, $C_{Э2}$,…, $C_{эn}$ на промежуточных и заключительном этапах

$$S^{т}(\lambda,\ C_{э}) = C_{0}\lambda_{1}C_{э1}\lambda_{2}C_{э2},\ \ldots,\ \lambda_{n}C_{эn}.$$

Временная структура принципиальной схемы технологического процесса представляет собой упорядоченную последовательность этапов обработки детали

$$S_{в}(Э,\ \Omega) = Э_{1}Э_{2},\ \ldots,\ Э_{i},\ ,\ldots,\ Э_{n}.$$

В соответствии с приведенными утверждениями рассмотрим виды и количество этапов (подфреймов) технологического процесса для обработки деталей различных классов в широком диапазоне точностных параметров и значений физико-механических свойств. В табл. 5 приведены состав этапов, их функции и условия выбора.

Подфрейм: «Заготовительный этап» свойствен большинству деталей. Исключение составляют детали, изготавливаемые из калиброванного прутка на токарных автоматах (условие P_1). Необходимость фрейма «Черновой этап» характеризуется рядом эвристических критериев, отражающих объем черновой обработки и некоторые организационные факторы, и определяется отдельным алгоритмом.

В связи с начальной степенью детализации технологических решений на первом уровне образования фреймов виды преобразований λ и состояния обрабатываемой детали $C_{Эi}$ на различных этапах (подфреймах) характеризуются обобщенными качественными и количественными параметрами. Для заготовительного и чернового этапов (подфреймов) преобразование $\lambda=\{M_1, M_2, ..., M_k\}$ обозначает соответственно методы получения заготовки и черновой обработки детали.

Состояние детали в заготовительном этапе (подфрейме) описывается заданием вида заготовки (пруток, труба, поковка, отливка и др.), а в черновом — перечнем участков детали, с которых снимаются напуски и припуски $C_{э2}= \{ Г_1, Г_2, ..., Г_q)$. Например, в деталях класса «тела вращения» указывается необходимость черновой обработки наружного контура и центрального отверстия, для плоскостных и корпусных деталей устанавливается перечень плоских участков и групп соосных отверстий, подлежащих черновой обработке.

В чистовом этапе (подфрейме) осуществляются получистовая и чистовая обработка основных поверхностей детали и окончательная — второстепенных (небольших и неточных отверстий, плоскостей, пазов и др.). В связи с этим в чистовом этапе (подфрейме) поверхности детали могут обрабатываться в один или несколько переходов (сверление — нарезание резьбы, сверление — расточка).

Дальнейшему разделению технологического процесса на этапы (подфреймы) в значительной степени способствуют термообработка и химико-термические операции, проводимые для получения требуемых физико-механических свойств детали или отдельных ее элементов. Положение этапов (подфреймов) цементации и азотирование определяются на основе утнорждсиия 8, в соответствии с которым точность предшествующей им механической обработки должна быть достаточно высокой, а последующая обработка — с малыми припусками. В связи с большими величинами поверхностного слоя при цементации, чем при азотировании, точность предшествующей механической обработки в первом случае может быть ниже, чем во втором. Так, цементируемые поверхности предварительно обрабатываются в этапе Э4, а азотируемые — в Э8. Соответственно разнесены этапы цементации Э5 и азотирования Э9.

Если предохранение от цементации осуществляется снятием цементационного слоя посредством механической обработки, то чистовой этап разделяется на два этапа: чистовой I и чистовой II. Между ними выполняется термический этап Э5, в котором производится цементация. В других случаях разделение чистовой обработки па этапы вызвано проведением термообработки азотирование или «старение» детали.

В выражении $\lambda_{\text{т}} : C^{\text{т}}_{\text{э},i-1} \rightarrow C^{\text{т}}_{\text{э},i}$ преобразование $\lambda_{\text{т}}$ обозначает вид термообработки, а $C^{T}_{Э,i-1}$ и $C^{T}_{Э,i}$ — состояние детали перед термообработкой и после нее. Состояние детали после термообработки описывается физико-механическими свойствами детали или отдельных ее поверхностей. Например, при цементации поверхностей $П_1; П_2, \ldots, П_к$ на глубину $r_{ц}$, состояние детали в этом этапе задается выражением

$$C_{\text{э}\text{т}} = \{П_1,\ П_2,\ \ldots,\ П_h\},\ r_{\text{ц}},\ H_{RC}.$$

В чистовых этапах механической обработки преобразование характеризуется множеством методов обработки поверхностей различных видов $\lambda_M = \{\text{M}_1,\ \text{M}_2,\ \ldots,\ \text{M}_n\}$, посредством которых деталь преобразуется из состояния $C_{Э,i-1}$ в $C_{Э,i}$. Указанные состояния описываются точностными параметрами обрабатываемых поверхностей $\varepsilon = \{\tau, \delta, Ш\}$

$$C_{\text{э},i} = \{П_1(\varepsilon_1),\ П_2(\varepsilon_2),\ \ldots,\ П_h(\varepsilon_h)\}.$$

Этапы Э1— Э 3 характерны для детали в целом, Э 4, Э 5, Э 7 — как для отдельных поверхностей, так и для детали, Э 6, Э 8, Э 9 относятся к обработке отдельных поверхностей. Для конкретной детали в зависимости от материала и вида термообработки некоторые этапы могут отсутствовать. Например, детали невысокой точности без закалки

обрабатываются в этапах Э1—Э4, в частном случае — в одном этапе. Это относится в первую очередь к обработке жестких небольших деталей из отливок, штамповок и пруткового материала. Однако даже при комплексной обработке деталей на агрегатных станках в отдельные позиции группируются черновая и чистовая обработки детали.

Рассмотрим некоторые фреймы алгоритмов формирования заготовительного, чернового этапов и фрейм алгоритма определения положения термообработки «улучшение» в начальных этапах технологического процесса.

17.2.2. Фреймы алгоритмов назначения заготовительного этапа

Исходными данными для решения этих задач служат системная модель обрабатываемой детали C_k=<S_j, $S_ф$, $S_и$, z> и программа выпуска N. Технические ограничения, влияющие па допустимые варианты принципиальной схемы,— множество применяемых на предприятии прогрессивных методов обработки, видов заготовок и их техническая характеристика:

$$M_k \in \{M_1, M_2, \ldots, M_l\},$$

$$З_q \in \{З_1, З_2, \ldots, З_f\}.$$

Задача заключается в том, чтобы определить целесообразность того или иного этапа и его функцию $\lambda_{Эi}$:$C_{Э,i-1} \rightarrow C_{Эi}$. В этом выражении неизвестны методы получения заготовок и черновой обработки λ_1 и λ_2, а также вид заготовки и начерно обработанной детали $C_{Э1}$\ и $C_{Э2}$. Выбор допустимых видов заготовки осуществляется на основе принципа технологической совместимости, устанавливающего такое соответствие материала, формы и размеров заготовки и детали, при котором могут быть получены заданные чертежом конфигурация, размерные и тчностные параметры обрабатываемой детали. Рациональным вид заготовки из полученного множества допустимых З={$З_1$, $З_2$, ..., $З_f$} выбирается на основе сравнения собестоимости изготовления деталей из заготовок различных видов. В связи с недостаточной детализацией технологических решений на первом уровне проектирования отсутствуют исходные данные для решения этой задачи. Поэтому выбор наиболее рационального вида заготовки осуществляется на основе таких показателей, как коэффициент использования металла при изготовлении детали из проката $W=M_{пр}/M_{дет}$, величина партии запуска и др. Совокупность этих

показателей с определенным приближением характеризует сравнительную себестоимость изготовления деталей из заготовок различных видов.

Рассматриваемая задача относится к классу задач выбора рациональных решений $З_р$ из их множества $\{З_i\}$, $i=1,2,\ldots,f$ по совокупности условий и показателей $\{q_j\}$, $j=1,2,\ldots,k$. В данном случае условия и показатели по степени их важности можно упорядочить. Тогда поиск решения будет представлять собой процедуру последовательного расчленения исходного множества $\{З_i\}$ на подмножества $З_{i1}$, $З_{i2}$ до тех пор, пока не будет получено одно или несколько наиболее рациональных решений.

Рассмотрим фрейм-алгоритм выбора вида стальных заготовок для деталей класса «тела вращения». В условиях мелкосерийного производства в качестве заготовок для деталей этого класса широко применяются пруток, труба, поковка, полученная методами свободной ковки или штамповки в подкладных штампах.

Для пустотелых деталей (цилиндры, втулки, диски) наиболее прогрессивными видами заготовок являются трубы. Заготовка труба назначается, если материал, диаметр и толщина стенки детали с учетом припусков позволяют выбрать трубу из применяемых на заводе типоразмеров. При этом необходимо, чтобы величина партии деталей была больше некоторой допустимой величины $n_{доп}$. Таким образом, задача заключается в том, чтобы из труб $З_{тр}=\{З_1, З_2, \ldots, З_i\}$, применяемых на заводе, выбрать трубу $З_{ik}$ наименьшего веса, отвечающую условиям: $M(З_{ik})\to\min$, $З_{ik}\in З_{тр}$, $MT_{дет}=MT_{ik}$, $n_{дет}\geqslant n_{доп}$, $D^{н}_{расч}\leqslant d_{ik}$, $H_{расч}\leqslant H_{ik}$. Реализация этой задачи осуществляется программным модулем выбора трубы наименьшего веса.

Для остальных деталей критериями выбора в качестве заготовки прутка или поковки служат коэффициент использования металла при изготовлении детали из прутка $W=P_{пр}/P_{дет}$ и величина партии n (рис. 17.2).

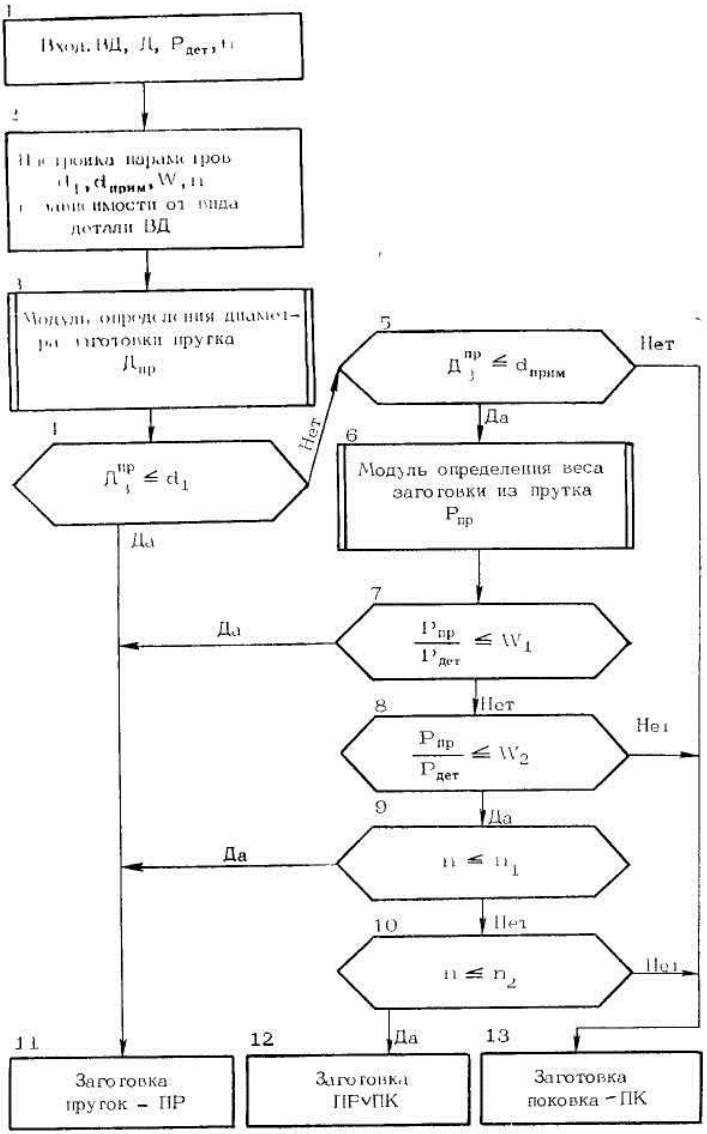


Рис.17.2. Модуль выбора вида заготовок для деталей класса «тела вращения»

Показатель *W* отражает косвенное влияние формы и размеров детали на выбор вида заготовки. Его смысл заключается в том, что чем больше металла необходимо снимать при изготовлении детали из прутка, тем менее целесообразно его использование в качестве заготовки. В таких случаях более эффективными будут заготовки-поковки.

Основной целью проектирования на данном уровне являются отсев нецелесообразных видов и методов получения заготовок и выделение для анализа на втором уровне только небольшого числа наиболее рациональные из них. Эту задачу позволяют решить простые критерии *W* и *n*. Ввиду эвристического характера критерия *W* его действие распространяется на определенный интервал диаметров заготовок d_1 - d_2. Для деталей различных классов устанавливаются минимальные значения d_1, до которых деталь целесообразно изготавливать из прутка. Например, для валов $d_1 \leq 45$ мм, втулок и дисков d_1 = 60 - 80 мм. В тех случаях, когда наибольший диамегр детали превышает максимальный диаметр прутка по применяемости $d_{max} > d_{прим}$, в качестве заготовки назначается поковка, так как деталь по размерам невозможно изготовить из прутка. Указанные условия проверяются операторами *4, 5* и *13.*

Для интервала диаметров заготовок d_1 - d_2 выбор з качестве заготовки прутка или поковки определяется в зависимости от значений *W* и *n*. Если $W \leq W_1$, то операторами 7 и 11 в качестве заготовки принимается пруток, для деталей с большими перепадами диаметров $W > W_2$— заготовка-поковка. В интервале значений $W = W_1 - W_2$ выбор заготовки зависит от величины партии деталей (операторы *8, 9, 10).* Если $n \leq n_1$, то назначается заготовка-пруток, если $n > n_2$ — заготовка-поковка. В интервале значений $n = n_1 - n_2$ ввиду эвристического характера выбранных критериев затруднительно однозначно определить вид заготовки. Поэтому за основу решения принимаются оба варианта (пруток, поковка). Окончательный выбор вида заготовки будет осуществлен на следующих уровнях проектирования при сравнении себестоимости изготовления детали из заготовок разных видов.

Метод получения заготовок-поковок для деталей различных видов зависит от их размеров и величины партии, например, для деталей зубчатые колеса штамповка в подкладных штампах целесообразна при $n > 80$ шт. По мере накопления опыта проектирования значения *d, W_1, n* уточняются и корректируются. На разных предприятиях величины этих критериев могут быть различными, они зависят от характеристик выпускаемой продукции и производственной системы предприятия.

Выбор рационального вида заготовки и метода ее получения для деталей сложной конфигурации представляет трудноформализуемую

задачу. Поэтому для ее решения целесообразно применение режима диалога технолога с ЭВМ. В управляющей программе этапа *Э1* имеются операторы Q_i обращения к технологу, посредством которых на экран устройства отображения выводятся модули обращения.

На рис. 17.3 показан управляющий алгоритм назначения вида заготовки и метода ее получения.

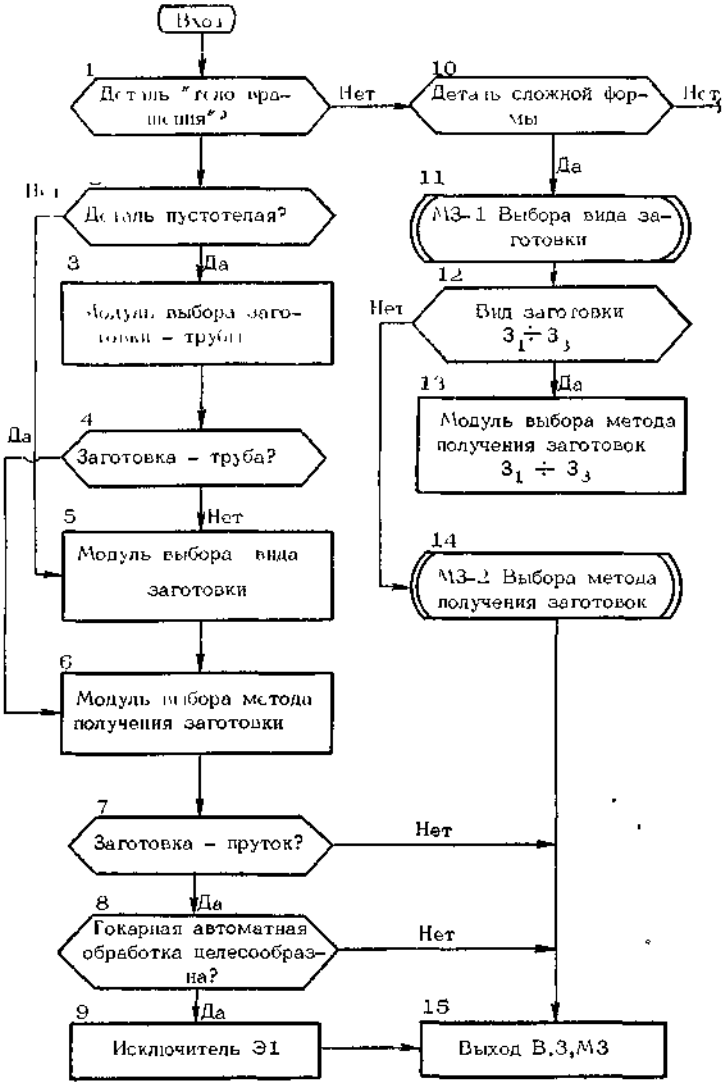


Рис. 17.3. Управляющий алгоритм выбора вида заготовки и метода ее получения

Он включает в себя ветви как программного, так и человеко-машинного вариантов решения этой задачи. Блоками *2, 3, 4* осуществляется проверка возможности выбора трубы в качестве

заготовки. Блоки *5, 6* служат для определения вида и метода получения заготовки для остальных деталей класса «тела вращения», блок 7 проверяет целесообразность токарно-автоматпой обработки деталей из прутка. Он включает в себя проверку ряда условий

$$\left.\begin{array}{l} d_{заг} \leqslant d_{доп}, \\ n \geqslant n_{доп}, \\ L_{дет} \leqslant l_{доп}, \\ \dfrac{L_{дет}}{d_{заг}} \leqslant q_{доп}, \\ K_{дет} \geqslant K_{доп}, \end{array}\right| \begin{array}{l} d_{доп} = 36, \\ n_{доп} \simeq 120, \\ l_{доп} \simeq 150, \\ q_{доп} \simeq 4{,}5, \\ K_{доп} = 4. \end{array}$$

Последнее соотношение обозначает, что число основных обрабатываемых поверхностей детали *К* должно быть больше некоторой допустимой величины $K_{доп}$. Если токарная автоматная обработка целесообразна, то блоком *8* делается отметка об исключении заготовительного этапа Э1, так как детали будут изготавливаться из прутков.

Для сложных деталей других классов (плоских, рычагов, корпусных деталей) логическими операторами *10, 11* управление передается операторам обращения к технологу, с помощью которых в режиме диалога осуществляется выбор вида заготовки и метода ее получения. Количество человеко-машинных ветвей алгоритма может быть различным и зависит от степени охвата автоматизацией проектирования деталей различных классов.

17.3. Фреймы алгоритмов формировании черновых и чистовых этапов (подфреймов)

После того как в этапе Э1 выбраны виды заготовок и методы их получения, определяются необходимость чернового этапа Э2, методы черновой обработки и состояние детали $C_{э2}$ после его выполнения. Целесообразность выделения черновой обработки детали в самостоятельный этап зависит от таких факторов, как вид заготовки, размер партии, состояние оборудования на заводе, квалификация рабочих-станочников, влияние черновой обработки на потерю точности станков, принятого способа организации производства в основных и заготовительных цехах (наличие участков для черновой обработки детали) и др.

При формировании чернового этапа задача заключается в том, чтобы разработать такой эвристический критерий, который бы интегрально учитывал влияние перечисленных выше факторов. На основе обобщения опыта проектирования технологических процессов обработки деталей класса «тела вращения» на заводах мелкосерийного производства в качестве локальных критериев для решения этой задачи выбраны вид и размеры заготовки, величина партии, отношение f суммарной длины черновых проходов при обработке наружного контура к длине детали

$$f = \frac{\sum_{i=1}^{r} m_i l_{0,i}}{L_{\text{дет}}},$$

где $m_i = \dfrac{d_{i+1} - d_i}{t_{\max}}$ — количество черновых проходов с глубиной $t_{\max}$ при обработке i ступени детали; $t_{0,i}$—-расстояние от торца детали до i ступени, включая ее длину; r--число обрабатываемых начерно ступеней детали.

Формула характеризует эвристически полученную зависимость влияния объема черновой обработки на целесообразность ее выделения в самостоятельный этап. Чем больше суммарная длина черновых проходов превышает длину детали, тем целесообразнее выделение черновой обработки в отдельные операции, выполняемые в этапе Э2. Эта формула справедлива для деталей, превышающих определенный диаметр $d_д \geq d_{доп}$. Значения f и величины $d_д$, n различны для деталей разных видов и составляют настроечные параметры алгоритма.

На рис. 17.4 показан модуль назначения этапа Э2 при обработке наружных поверхностей для деталей различных типов, изготавливаемых из прутка.

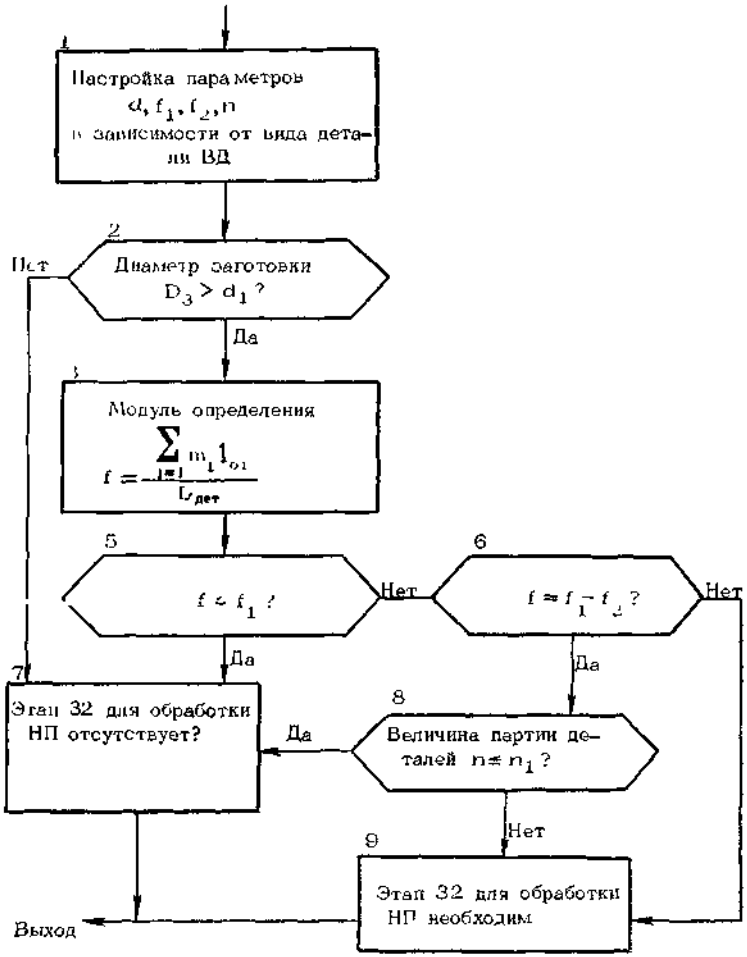


Рис. 17.4. Алгоритм назначения чернового этапа *Э2* для обработки наружных поверхностей *(НП)* деталей, изготавливаемых из прутка

Если диаметр заготовки не превышает допустимых значений $d_3 \le d_1$ (оператор *2),* то черновые переходы в самостоятельную операцию не выделяются, а выполняются вместе с получистовыми в этапе э4. Модулем *3* производится вычисление величины *f*. Для более крупных деталей каждого типа в зависимости от величины *f* возможны три варианта: $f \leqslant f_1$, $f = f_1 \div f_2$, $f > f_2$.

В первом случае ввиду небольшой длины черновых проходов они не выделяются в самостоятельную операцию (оператор 5), во втором при малых размерах партии $n \leq n_1$ черновая обработка в отдельную операцию также не выделяется, а при $n > n_1$ засылаются сведения о необходимости черновой обработки в этапе Э2 (операторы *6, 8).* В третьем случае в связи с большим объемом черновой обработки независимо от величины партии назначается этап Э2. В табл. 6 приведены конкретные значения параметров настройки $d_{i\ доп}$, f и n.

Таблица 6

Параметры настройки алгоритма

Параметры	Валы	Втулки	Диски
d_1	30	35	42,0
f_1	0,8	1,0	1,2
f_2	1,2	1,4	1,6
n	5,0	8,0	12,0

Черновая операция обработки центральных отверстий назначается в том случае, если хотя бы одно отверстие диаметром больше $d_д$ (обычно $d_д$ принимают равным 18—20 мм) и длиной $l > kd_д$. При меньших размерах отверстий обработку целесообразнее производить в получистовом этапе Э4. Заготовки-поковки, полученные свободной ковкой и в подкладных штампах, имеют большие припуски и напуски и всегда проходят черновую обработку в этапе Э2.

Разработка фреймов алгоритмов назначения чернового этапа и определения состояния детали после черновой обработки для деталей сложной формы представляет значительные трудности. Это связано с необходимостью алгоритмического анализа формы детали, ее жесткости, организационно-технических и других условий. Поэтому в системе предусмотрен режим диалога технолога с ЭВМ. В управляющем алгоритме этапа Э2 имеются операторы обращения ЭВМ к технологу. На экран устройства отображения выводится модуль обращения, содержащий запрос технологу: черновой этап Э1 необходим? Технолог, анализируя деталь, выбирает решение и с помощью функциональной клавиатуры вводит в ЭВМ ответы «да.», «пет» *. После этого на экран выводится модуль обращения, содержащий запрос:

укажите: 1) номера участков для черновой обработки;
2) метод обработки участка.

Технолог на основе анализа детали и предполагаемой конфигурации заготовки вводит в ЭВМ ответы на поставленные вопросы. Управляющий алгоритм назначения чернового этапа и определения состояния детали после черновой обработки показан на рис. 17.5.

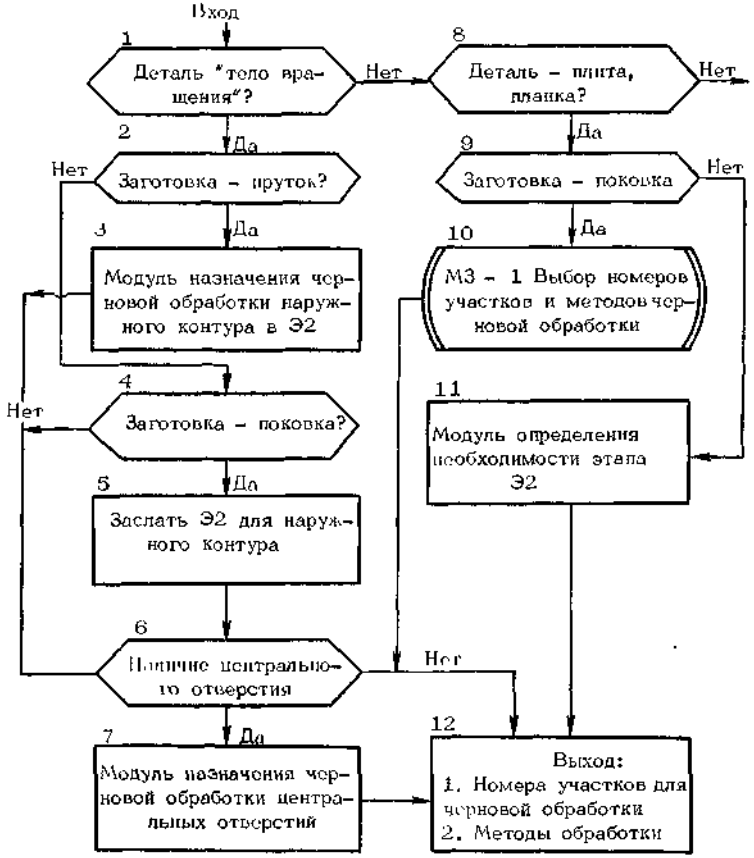


Рис.17.5. Управляющий алгоритм назначения чернового этапа

Он предусматривает решение задач в автоматическом и человеко-машинном режимах. Первый режим характерен для деталей класса «тела вращения», а второй — для более сложные деталей. Операторами 1, *2, 4, 5* производится назначение этапа Э2 при обработке наружного контура заготовок-поковок. Модулями *3, 7* определяется необходимость назначения этапа Э2 для обработки наружных поверхностей и центральных отверстий в деталях,

изготавливаемых из проката. Для деталей более сложной формы операторами *10* и *11* на экран устройства отображения выводится модуль запрос к технологу с просьбой указать номера участков детали и методы их черновой обработки.

Взаимное положение термообработки «улучшение» (ТО-У), чернового и получистового этапов определяетсяна основе утверждения 4.3, устанавливающего возможность проведения ТО-У в заготовке. Кроме того, учитывается необходимость выделения черновой обработки в самостоятельный этап (табл. 7).

Таблица 7

Алгоритм определения положения термообработки «улучшение»

Проверки	Заслать ТО-У в этапы			
	Э1		Э3	Э7
Наличие этапа черновой обработки Э2	Нет	Да	Да	Нет
Возможность ТО-У в заготовке	Да	Да	Нет	Нет

Для деталей сложной формы разработка алгоритма определения прокаливаемостн заготовки по всем сечениям представляет собой трудную задачу. В связи с этим в управляющей программе предусмотрен переход на режим человеко-машинного проектирования. Модуль обращения к технологу содержит просьбу указать, возможно ли ТО-У в заготовке. После ответа технолога на указанный вопрос определение этапа, в котором будет выполняться ТО-У, производится по табличному алгоритму. Если этап черновой обработки необходим и имеется возможность проведения ТО-У в заготовке, то ТО-У переносится в заготовительный этап и выполняется после получения заготовки, в противном случае она выполняется в этапе Э3 (см. 1-й и 2-й столбцы таблицы). Для тех случаев, когда черновая обработка не выделяется в самостоятельный этап и возможно проведение ТО-У в заготовке, она переносится в заготовительный этап. Если последнее условие не выполняется, то ТО-У назначается в этап Э7, т. е. производится после получистовой обработки детали в этапе Э4 (см. 3-й и 4-й столбцы таблицы).

17.3.1.Формирование чистовых этапов (подфреймов) принципиальной схемы технологического процесса

После того как выбраны вид заготовки, методы ее получения и черновой обработки, задача состоит в том, чтобы определить функцию чистовых и окончательного этапов обработки детали, т. е.

$$\lambda_{э4} : C_{э3} \to C_{э4},$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\lambda_{эk} : C_{э,k-1} \to C_{э,k}.$$

В этих выражениях известны только конечное состояние детали $C_к$ и после черновой обработки $C_{э1}$ или состояние заготовки $C_{заг}$, если черновой этап отсутствует. Задача состоит в том, чтобы выбрать методы обработки поверхностей $\lambda = \{\mathbf{M}_1, \mathbf{M}_2, \ldots\}$ и точностные параметры состояния детали в каждом этапе:

$$C_{эi} = \{\Pi_1(\varepsilon_1), \Pi_2(\varepsilon_2) \ldots \Pi_k(\varepsilon_k)\}.$$

Поскольку базовыми структурными элементами обрабатываемой детали в процессах механической обработки выступают поверхности, то синтез получистовых и чистовых этапов будем осуществлять на основе декомпозиции функции этапа $\lambda_{эi} : C_{э,i-1} \to C_{э,i}$ на совокупность подфункций $\mathbf{M}_j : C_{n,j-1} \to C_{n,j}$ обработки отдельных поверхностей. Алгоритм синтеза возможных вариантов *МОП* построен на основе утверждения 2 (рис. 17.6).

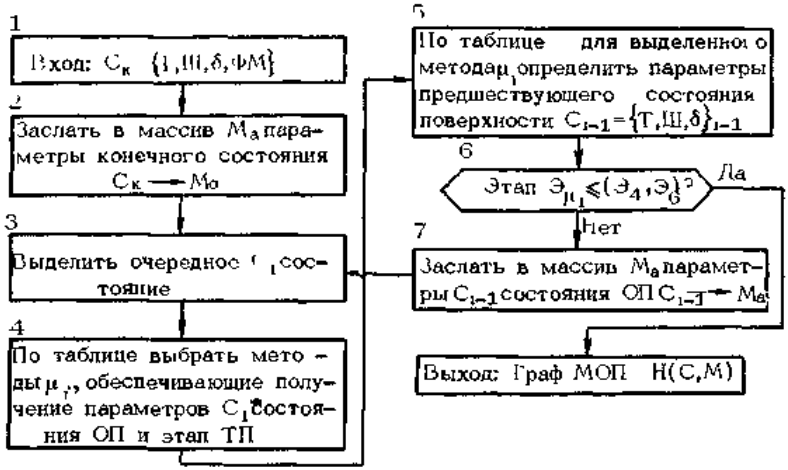


Рис. 17.6. Модуль алгоритм формирования маршрута обработки поверхности

В начале работы алгоритма оператор *2* засылает в исходный массив $M_а$ параметры конечного состояния обрабатываемой поверхности $C_к$={*τ, δ, Ш, МС)*. Оператор *3* выбирает из $M_а$ очередное C_i и пересылает его в рабочий массив $M_р$. Операторами *4, 5* осуществляется

синтез куста подграфа с вершиной в C_i путем построения непосредственно следующего за ней множества вершин $\{C_{i-1}\}$. Для этого пз таблицы технологических возможностей методов обработки выбираются наиболее рациональные и соответствующие им параметры состояния $\{M_l, C_{i-1}\}$. Оператор *6,* сравнивая номер этапа для выбранного метода обработки с номерами получистовых этапов, оканчивает построение графа *МОП* или передает управление опера тору 7, который засылает множество дуг $\{M_l, C_{i-1}\}$ в конец исходного массива M_a и организует новый цикл работы алгоритма до тех пор, пока не будут получены все целесообразные варианты *МОП.* Для резкого сокращения количества вариантов *МОП* в задачу синтеза включается ряд эвристических условий, полученных на основе обобщения опыта назначения методов обработки в отрасли или на конкретном предприятии. В результате на каждом этапе выбираются не все возможные методы обработки, а только наиболее рациональные. Это дает возможность в дальнейшем значительно повысить эффективность фреймов алгоритмов формирования маршрутов обработки деталей в целом, так как отпадает необходимость синтеза и анализа большого числа вариантов технологического процесса, основанных на базе малоэффективных *МОП.*

Рассмотрим фреймы алгоритмов назначения маршрута обработки поверхностей различных классов применительно к условиям мелкосерийного производства. Для того чтобы *МОП* в наибольшей степени отвечали требованиям построения рационального технологического процесса в фреймах алгоритмов синтеза, необходимо учитывать влияние на выбор метода обработки не только параметров отдельной поверхности, по и детали в целом (форма детали, материал и термообработка, технические требования на точность взаимного расположения поверхностей, программа выпуска).

Большинство алгоритмов имеет табличную структуру и состоит из набора применяемых на предприятии или в отрасли прогрессивных методов обработки поверхностей определенного класса и совокупности условий выбора того или иного метода обработки.

При поиске метода обработки параметры поверхности и детали сравниваются с табличными. В качестве искомого выбирается метод, обеспечивающий эффективное получение всех точностных параметров поверхности. Если хотя бы одно условие не выполняется, стандартная программа анализа табличного алгоритма выделяет следующий метод и производит проверку всех условий.

В качестве примера в табл. 8 приведен табличный алгоритм выбора методов и этапов обработки наружных поверхностей вращения.

Таблица 8

Алгоритм выбора методов и этапов обработки наружных поверхностей вращения

Условия, определяющие выбор методов и этапов обработки	Обозначение параметра	Методы обработки					
		точение в Э4	точение в Э6, Э4	точение или шлифование в Э4	шлифование в Э8	шлифование в Э11, Э8	доводочное шлифование в Э13
Точность поверхности, кв.	$T\geqslant$	11	11	6	5÷6	5÷6	5÷6
Шероховатость поверхности	$R_a\geqslant$	2,5	2,5	1,25	0,32	0,16	0,04
Технические требования на точность расположения поверхности	$\delta_{ij}\geqslant$	0,15	0,15	0,05	0,01	0,01	
Вид предшествующей термообработки	TO	—	Предохранение от цементации		Закалка ТВЧ	Азотирование, «старение»	—

Этот алгоритм состоит из трех частей. В первой на основе анализа таких параметров C_i состояния, как точность размера, шероховатость поверхности, точность взаимного расположения поверхности (биение) и вида предшествующей термической обработки, выбираются наиболее рациональные методы обработки этой поверхности.

Точением окончательно обрабатываются цилиндрические, конические и другие поверхности 11-го и более грубых квалитетов с шероховатостью $R_a \geq 2,5$. Поверхности 2—3-го классов точности с шероховатостью R_a=2,5-1,25 обрабатываются точением или шлифованием, если деталь не требует закалки. Окончательно метод обработки поверхности будет выбран на следующем уровне проектирования (при разработке маршрута). Поверхности с теми же параметрами, но в закаливаемых деталях, обрабатываются шлифованием. Для достижения высокой чистоты назначается переход доводочного шлифования, выполняемый на точных станках мелкозернистыми кругами. Два последних столбца содержат методы обработки поверхностей первого класса точности.

Методы обработки, соответствующие параметрам, приведенным в 1-м и 2-м столбцах табл. 8, засылаются в чистовой этап Э4, если химико-термическая обработка отсутствует или предохранение поверхности от цементации осуществляется посредством омеднения. В тех случаях, когда предохранение от цементации осуществляется припуском, соответствующие переходы засылаются в этап Э6 и выполняются после химико-термической обработки перед закалкой. Шлифуемые поверхности в закаленных деталях (см. 4-й столбец) обрабатываются в этапе Э8, доводочное шлифование производится в этапе Э12.

По заданным характеристикам поверхности на выполняемом переходе определяются точность операционных размеров и шероховатость на предыдущем переходе. На основе этих параметров по рассмотренному выше алгоритму устанавливаются наиболее рациональные методы обработки. Этот процесс продолжается до тех пор, пока параметры предшествующего перехода будут равны или близки к параметрам заготовки или поверхности после черновой обработки.

Для поверхностей других классов маршрут обработки в чистовых этапах будет однозначно определяться значениями параметров этих поверхностей и детали в целом. В качестве примера рассмотрим фрейм алгоритма назначения маршрута обработки плоских поверхностей (плоскостей, пазов, окон).

Первый маршрут обработки (табл. 9) включает только фрезерование в этапе Э4 и применяется для незакаленпых и улучшенных деталей при точности размеров грубее 8—9-го квалитстон, шероховатости поверхности $R_a \geq 2{,}5$ и технических требованиях на точность взаимного расположения, превышающих $\delta_{доп} \sim 0{,}15$.

Таблица 9

Алгоритм определения маршрута обработки плоскостей, пазов, многогранников

Условия, определяющие выбор маршрута обработки	Этапы	Маршруты обработки				
	Э13					Слесарная доводка
	Э8		Слесарная доводка	Фрезеровать	Шлифовать	Шлифовать
	Э4	Фрезеровать	Фрезеровать	Фрезеровать	Фрезеровать	Фрезеровать
Вид поверхности	*ВП*	—		Закрытая	—	—
Точность размера, кв.	$T \geqslant$	8÷9	8÷9	6÷7	6÷7	6÷7
Шероховатость поверхности	$R_a \geqslant$	2,5	2,5	0,63	0,63	0,16
Вид предшествующей термообработки	*ТО*	ТО-У	Закалка	ТО-У	—	—
Технические требования	$\delta_{ij} \geqslant$	0,15	0,1	—	0,05	—

Эти технические требования обеспечиваются установкой инструмента на размер с учетом допустимой величины погрешности обработки на фрезерных станках. Второй маршрут обработки используется при тех же технических требованиях, но для закаленных деталей и содержит наряду с фрезерованием в этапе Э8 слесарную операцию для снятия окалины и зачистки поверхности после термообработки.

Закрытые поверхности (пазы, канавки) в незакаленных или улучшенных деталях обрабатываются фрезерованием в этапе Э4 и в этапе Э8 на координатно-расточных станках (3-й маршрут обработки). Плоские поверхности 6—7-го квалитетов точности и шероховатостью R_a=1,254-0,63 независимо от термообработки и технических требований фрезеруются в этапе Э4 и шлифуются в этапе Э8 (4-й маршрут обработки). При более высокой шероховатости поверхности в дополнение к рассмотренным операциям в план обработки засылается доводочная операция.

При необходимости предохранения обрабатываемой поверхности от цементации фрезерная операция из этапа Э4 переносится в Э6, т. е. выполняется после цементации перед закалкой. Для окон с полукруглыми концами в этапе Э4 производятся сверление и фрезерование, а окна прямоугольной формы обрабатываются сверлением, фрезерованием и долблением. При малой длине окна $0{,}8 \leq l/b \leq 1{,}3$ фрезерование исключается, так как после сверления

становится возможным производить долбление окна. Эти особенности обработки комбинированных поверхностей учитываются операторами, записанными в примечании к маршруту обработки.

Результаты проектирования маршрутов обработки поверхности детали в получистовых и чистовых этапах можно представить в виде табл. 5. Ее столбцы соответствуют маршрутам обработки отдельных поверхностей, а строки — этапам технологического процесса. В каждую клетку таблицы записаны один или несколько близких по эффективности методов обработки поверхностей и ее точностные параметры, полученные после обработки.

В целом допустимые варианты принципиальной схемы образуют граф *G(C, M),* вершинам которого соответствуют состояния детали $C_{Эi}$ на различных этапах технологического процесса, а дугам — совокупность методов обработки поверхностей в этапе (рис. 17.7).

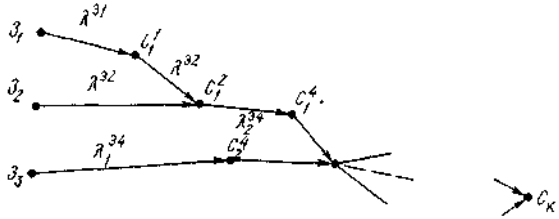


Рис. 17.7. Граф вариантов принципиальной схемы технологического процесса

Граф *G(C, M)* ориентированный, стрелки проставлены в направлении от черновых этапов к чистовым и окончательным.

На рисунке показано, что деталь может изготавливаться из заготовок трех видов: $З_1$, $З_2$ и $З_3$. Для некоторых вариантов принципиальной схемы необходим заготовительный этап Э1, для других вариантов он совмещен с черновой обработкой в этапе Э2. Любой путь из области *З* в вершину $С_к$, соответствующую конечному состоянию детали, представляет собой один из вариантов принципиальной схемы процесса. Наиболее рациональным был бы вариант принципиальной схемы, на основе которого на следующих уровнях будет спроектирован варлаит технологического процесса, имеющий наименьшую себестоимость. В связи с недостаточной детализацией проектных решений и эвристическим характером критериев оценки однозначный выбор такого варианта принципиальной схемы для деталей сложной формы не представляется возможным. Поэтому для дальнейшего проектирования в таких случаях оставляется не один, а два-три наиболее рациональных варианта или выбор наиболее

рациональных вариантов принципиальной схемы производится в режиме «технолог-ЭВМ». Для этого матрица маршрутов обработки поверхностей детали и граф возможных вариантов принципиальной схемы выводятся на экран устройства отображения.

Технолог в режиме редактирования может внести необходимые исправления и уточнения. Редактирование может касаться замены методов обработки, а в ряде случаев переноса их из одного этапа в другой. На графе *G(C, M)* отсекаются некоторые заведомо неэффективные пути или ветви процесса. В результате для дальнейшего проектирования остается меньшее число наиболее рациональных вариантов, что значительно повышает эффективность процесса проектирования.

18. Образование фреймов моделирования процесса проектирования технологических маршрутов

18.1. Закономерности синтеза фрейма маршрута механической обработки деталей

Постановка задачи. Исходными данными для проектирования на втором уровне, кроме сведений о детали и программе выпуска, служат полученные ранее временная структура принципиальной схемы процесса $S_B = Э_1, Э_2, .. ., Э_i ,..., Э_n$ и описание функции каждого этапа $\lambda_i = C_{i-1} \rightarrow C_i$. В этом выражении преобразование λ_i задано набором методов обработки поверхностей в этапе $\lambda_i = \{\mu_1, \mu_2, \ldots, \mu_k\}$, а состояние детали C_i — множеством обрабатываемых поверхностей $П_j$ и их точностными параметрами, т. е.

$$C_i = \{П_1(\varepsilon_1), П_2(\varepsilon_2), \ldots, П_k(\varepsilon_k)\}.$$

Техническими ограничениями, определяющими возможные варианты фрейма технологического маршрута, являются набор применяемых типов оборудования *СТ* и их техническая характеристика $Z_i^{ст}$:

$$CT \in \{CT_1, CT_2, \ldots, CT_m\},$$

$$Z_i^{ст} = \{Z_1, Z_2, \ldots, Z_q\}.$$

Это значит, что тип оборудования должен быть выбран из имеющегося на предприятии. Кроме того, при проектировании используются сведения о имеющейся на предприятии оснастке.

Фрейм структуры технологического процесса на втором уровне проектирования описывается совокупностью трех взаимосвязанных моделей: временной, функциональной и пространственной. Временная модель определяет состав и последовательность операций в каждом этапе

$$S_{э_i}(A, \Omega) = A_1 A_2, \ \ldots, \ A_j, \ \ldots, \ A_m,$$

где A_j — j операция в $Э_i$ этапе (подфрейме). Функциональная модель характеризует порядок преобразования обрабатываемой детали из одного состояния в другое:

$$S_ф(C, A) - C_0 A_1 C_1 A_2 C_2, \ \ldots, \ A_m C_m.$$

На рассматриваемом уровне проектирования функция операции представлена более подробно, чем функция этапа на первом уровне. Если состояние детали в этапе характеризовалось видами обрабатываемых поверхностей и их точностными параметрами $C_i=\{П_j(\varepsilon_j)\}$, $j=1,2,\ldots,k$, то при описании состояния детали в операции, помимо этих данных, необходимо указать виды выбранных базовых поверхностей *БП* и граф их размерных связей *В (Б, Р)*

$$C_i = \{B(Б, P), \ П_j(\varepsilon_j)\}.$$

Таким образом, степень детализации описания функции структурных элементов технологического процесса от уровня к уровню возрастает.

Фрейм пространственной модели технологического маршрута задан графом $S_п(Б, P)$, вершины *Б* которого соответствуют выбранным базовым поверхностям, а дуги *Р* — размерным и точностным связям между ними. На рис. 18.1 показан граф размерных связей базовых поверхностей операций одного этапа.

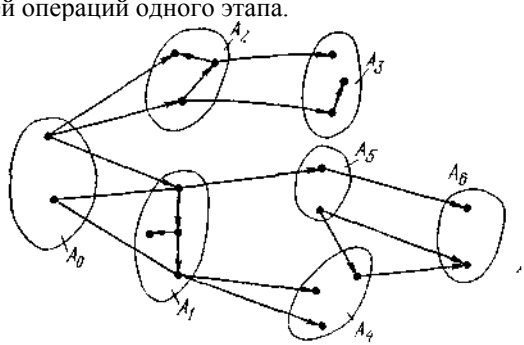


Рис.18.1. Граф размерных связей операций этапа (подфрейма)

В этом графе связи установлены только между базовыми поверхностями различных операций. Нахождение связей между базовыми и обрабатываемыми поверхностями внутри каждой операции производится на уровне проектирования операционной технологии.

Задача заключается в том, чтобы в каждом этапе (подфрейме) определить наиболее рациональные временную структуру маршрута обработки детали $S_в^Э = A_1, A_2, ..., A_r$, включающую в себя установление состава и последовательности операций; функцию каждой операции $\varphi : C_{i-1} \rightarrow C_i$ и функциональную структуру маршрута $S_ф(C, A)$; тип оборудования CT_j для осуществления A_j операции; базовые поверхности *БП* и схему установки детали в операции.

В соответствии с общей методологией системного проектирования синтез фреймов вариантов маршрута обработки будем производить декомпозицией общей функции этапа $Ф_{Эi}$ на функции отдельных операций $Ф_j$

$$ДК : Ф_{Э_i} \rightarrow \{Ф_1, Ф_2, \ldots, Ф_k\}.$$

Далее следуют процедуры установления пространственно-временных и функциональных связей операций процесса и соответствующих им структур

$$S_в(A, \Omega), \; S_ф(C, A), \; S_ц(Б, P).$$

Заключительным звеном синтеза служит выбор типа оборудования, с помощью которого реализуются функции полученных операций.

Для того чтобы сделать синтез фрейма маршрута обработки целенаправленным и эффективным, необходимо сформулировать исходные посылки и утверждения, определяющие технологические закономерности решения этих задач. Важным условием объединения набора методов обработки поверхностей в самостоятельные операции и установления их последовательности является наличие у них свойства технологической совместимости.

Две операции процесса попарно совместимы $A_{i-1} \sigma A_i$, если состояние детали на выходе одной операции C_{i-1} может быть исходным C_i для другой операции, т. е. $C_{i-1} \sigma C_i$. Деталь и приспособление технологически совместимы, если форма, размеры и пространственное расположение базовых, обрабатываемых и зажимных поверхностей соответствуют указанным параметрам базовых, зажимных и направляющих элементов приспособления. Можно говорить о совместимости совокупности методов обработки поверхностей с технологическими возможностями оборудования. На основе изложенного выше сформулируем исходную посылку или принцип технологической совместимости.

Посылка 1. Элементы технологического процесса и системы СПИД технологически совместимы, если их свойства и параметры обеспечивают функциональные и пространственно-временные связи и взаимодействие, направленные на выполнение функции более крупного структурного элемента процесса, в состав которого они входят. Исходя из указанного принципа, могут быть получены ряд утверждений, определяющие закономерности формирования операций и их последовательность.

Утверждение 1. Совокупность методов обработки этапа (подфрейма) $\lambda_{э_i} = \{\mu_1, \mu_2, \ldots, \mu_k\}$ разделяется на непересекающиеся подмножества

$$A_1 = \{\mu_1, \mu_2, \ldots, \mu_{l_1}\},$$

$$A_2 = \{\mu_3, \mu_5, \ldots, \mu_{l_2}\},$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$A_r = \{\mu_c, \mu_{c+q}, \ldots, \mu_{l_r}\}$$

так, чтобы в полученных подмножествах виды методов *щ,* точностные параметры обрабатываемых поверхностей $\Pi_j(\varepsilon_j)$, а также форма *ФД* и размеры *Р* детали были совместимы с технологическими возможностями оборудования по указанным характеристикам

$$\{\mu_i, \Pi_j(\varepsilon_j)\} \supseteq \{\mu_k(\varepsilon_k)\},$$

$$ФД_{дет} \supseteq КД_{тв},$$

$$P_{дет} \leqslant P_{тв}.$$

Полученные в результате расчленения подмножества A_2, A_2, ..., A_r образуют укрупненные операции этапа. Укрупненные операции, сформированные в соответствии с технологическими возможностями применяемого оборудования, характеризуются максимальной степенью концентрации методов обработки. Различное взаимное расположение и размеры входящих в укрупненную операцию поверхностей одного вида, например пазов, отверстий и других, приводят к необходимости переналадки оборудования. Это связано с увеличением операционного времени и затрат на технологическую совместимость указанных групп переходов внутри операции.

Разделение укрупненных операций на более простые, объединяющие обработку однородными методами однотипных поверхностей с одинаковыми размерными параметрами, позволяет сократить суммарное время на переналадку станка при переходе с обработки одной группы поверхностей на другую. При этом наладка станка для

выполнения простой операции производится один раз на всю партию деталей. Однако суммарное время на установку деталей и подготовительно-заключительное для совокупности простых операций будет больше, чем для одной укрупненной.

Утверждение 2. Дифференциация укрупненной операции на простые должна быть такой, чтобы обеспечить минимальные затраты $E^{тс}_i$ на технологическую совместимость групп переходов различных видов при обработке партии деталей *n:*

$$A_2^* = \min_{A_2^* \in A} \sum_{i=1}^{r} E_i^{тс}.$$

В мелкосерийном производстве при использовании универсальных и быстропереналаживаемых приспособлений и инструмента определение наиболее рациональной степени дифференциации укрупненной операции на простые может производиться на основе сравнения суммарного времени выполнения простых операций T_i с затратами времени на укрупненную операцию $T_{ук}$.

$$\sum_{i=1}^{k} T_i \leqslant T_{ук}. \qquad (9)$$

Включив в выражение $A = \{\mu_i\}$, описывающее структурный состав укрупненной операции, операторы установки и переустановки детали $У_j$ и операторы переналадки станка $H_{j,j+1}$ при переходе с обработки поверхностей одного видак другому, получим

$$A_{ук} = \{У_1, \mu_1, \mu_3, \ldots\}, \quad H_{12}\{У_2, \mu_2, \mu_5, \ldots\}$$
$$\ldots H_{q-1,q}\{У_q, \mu_{q_1}, \mu_{q_2}, \ldots, \mu_{q_r}\}.$$

Этому выражению соответствует формула операционного времени

$$T_{ук} = \left(t_{у1} + \sum_{1}^{n_1} t_{i_1}\right) + t_{п_{12}} + \left(t_{у_2} + \sum_{n_1}^{n_2} t_{i_2}\right) + \cdots$$
$$\cdots + t_{п_{q-1,q}} + \left(t_{у_q} + \sum_{c-1}^{c} t_{i_c}\right) + \frac{t^{пз}}{n}.$$

При дифференциации укрупненной операции на *q* простых соответствующие им формулы структурного состава и операционного времени будут иметь вид

$$A_1 = У_1,\ \mu_1,\ \mu_3,\ \ldots,\ \mu_{n1},\quad T_1 = t_{у_1} + \sum_{1}^{n_1} t_{i_1} + \frac{t_1^{пз}}{n},$$

$$A_2 = У_2,\ \mu_2,\ \mu_5,\ \ldots,\ \mu_{n_2},\quad T_2 = t_{у_2} + \sum_{n_1}^{n_2} t_{i_2} + \frac{t_2^{пз}}{n},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$A_q = У_q,\ \mu_{q_1},\ \mu_{q_2},\ \ldots,\ \mu_{q_r},\quad T_q = t_{у_q} + \sum_{c-1}^{c} t_{i_c} + \frac{t_q^{пз}}{n}, \qquad (10)$$

где $t_н$ — время подналадки; $t_i^{пз}$— подготовительно-заключительное время; t_{ic}— время обработки i методом c поверхности.

Подставляя в левую и правую части неравенства (9) их значения по формулам (10), получаем

$$\sum_{j=1}^{q}\left(t_{уj} + \sum_{n_j}^{n_{j+1}} t_{j,i} + \frac{t_j^{пз}}{n}\right) \leqslant$$

$$\leqslant \sum_{j=1}^{q} t_{уj} + \sum_{1}^{q}\sum_{n_j}^{n_{j+1}} t_{ji} + \sum_{1}^{q-1} t_{нj} + \frac{t_{ук}^{пз}}{n}.$$

Отсюда размер партии, начиная с которого целесообразно дифференцировать укрупненную операцию на q простых, выполняемых без переналадки, определяется по формуле

$$n \geqslant \frac{\sum_{1}^{q} t_j^{пз} - t_{ук}^{пз}}{\sum_{1}^{q-1} t_{нj-1,j}}.$$

Если в укрупненной операции все поверхности обрабатываются с одной установки, то в правой части неравенства

$$\sum_{j=1}^{q} t_{уj}$$

необходимо заменить на $t_у$. В этом случае размер партии деталей найдем по формуле

$$n \geqslant \frac{\sum_{j=1}^{q} t_j^{пз} - t_{ук}^{пз}}{\sum_{1} t_{н_{j-1},j} + t_у - \sum_{1}^{q} t_{уj}}.$$

Определение $t^{пз}$, $t_н$, $t_у$ может производиться по укрупненным нормативам, в которых в зависимости от параметров поверхности и детали по таблицам выбираются указанные составляющие времени обработки.

Форма детали оказывает существенное влияние на последовательность обработки поверхностей и более сложных элементов в каждом этапе. Рассмотрим влияние на последовательность обработки элементов детали таких ее системных характеристик, как уровень расчленения, устанавливаемый при конструктивно-технологическом анализе; ранг конструктивных операций «отсечение»; вид и характер пересечения элементов детали.

Указанные системные характеристики детали могут быть получены на основе графа формы *Ф(П, U).* В связи с этим его необходимо принять за основу при построении технологического графа *Т(П, М),* характеризующего порядок обработки структурных элементов детали *П* при формировании маршрута ее обработки.

Утверждение 3. В соответствии с многоуровневой структурой детали упорядочение обработки ее структурных элементов производится в порядке возрастания уровня расчленения, т. е. вначале устанавливается последовательность обработки сложных элементов Q^1_i первого уровня, затем для каждого из Q^1_i — последовательность обработки составляющих его элементов Q^2_{ij} второго уровня и т. д. до уровня простых и нормализованных поверхностей:

$$УП^1_{Q_0}: \{Q^1_1, Q^1_2, \ldots, Q^1_k\} \to \langle Q^1_3, Q^1_k, \ldots, Q^1_1 \rangle,$$

$$УП^2_{Q_1}: \{Q^2_{11}, Q^2_{12}, \ldots, Q^2_{k_2}\} \to \langle Q^2_{12}, Q^2_{11}, \ldots, Q^2_{k_2} \rangle,$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$УП^n_q: \{Q^n_{q_1}, Q^n_{q_2}, \ldots, Q^n_{q_k}\} \to \langle Q^n_{q_3}, Q^n_{q_2}, \ldots, Q^n_{q_1} \rangle,$$

где $УП^n_q$ — процедура упорядочения элементов Q^n_{qk} на *n* уровне расчленения в зависимости от влияния характера конструктивных операций и отношений;

$$\{Q^n_{q_1}, Q^n_{q_2}, \ldots, Q^n_{q_k}\}$$

— структурный состав сложного элементадетали на *n* уровне расчленения; $\langle Q^n_{q_3}, Q^n_{q_2}, \ldots, Q^n_{q_1} \rangle$ —кортеж, или частично упорядоченное множество тех же элементов.

Например, в деталях класса «тела вращения» в каждом этапе вначале устанавливается последовательность обработки наружного и внутреннего контуров, т. е. элементов первого уровня. Затем определяется порядок обработки поверхностей в пределах каждого из них. В корпусных деталях сначала формируются порядок обработки сторон и контуров, затем комплексов, групп соосных отверстий и поверхностей.

Таким образом, многоуровневое упорядочение позволяет учитывать влияние сложности формы детали на формирование последовательности ее обработки.

Следующее утверждение определяет влияние ранга конструктивной операции «отсечение» на последовательность обработки элементов детали.

Утверждение 4. На каждом уровне расчленения последовательность обработки структурных элементов детали Q_i осуществляется в порядке возрастания их ранга *r*, т. е. от основных к элементам более высоких рангов:

$$M = \langle \{Q^1_1, Q^1_3, \ldots\} \{Q^2_2, Q^2_5, \ldots\} \cdots \{Q^r_k \ldots\} \rangle .$$

На рис. 18.2 показан участок, образованный одной основной и несколькими поверхностями более высоких рангов, и соответствующие им граф формы и технологический граф *Т(П, М)*.

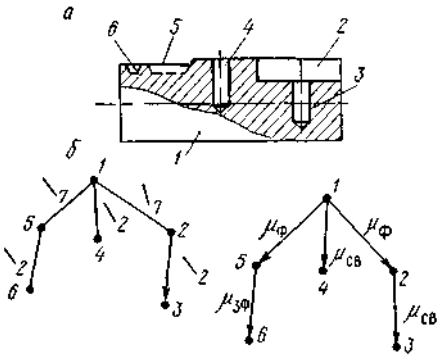


Рис. 18.2. Влияние ранга относительного взаимного расположения поверхностей на последовательность их обработки: *а* — эскиз детали; б — граф формы детали *Ф(П, U)* и граф последовательности обработки поверхностей детали

Этот граф определяет следующий порядок обработки детали. Вначале обрабатывается основная поверхность (цилиндр), затем — любая из трех поверхностей: паз *2,* плоскость *5,* отверстие *4.* В последнюю очередь выполняются отверстие, расположенное на дне паза, и зубья рейки 6.

Последовательность обработки поверхностей в порядке возрастания их ранга объясняется рядом технологических факторов: невозможностью формообразования таких поверхностей, как зубья, шлицы, резьбы и другие, до получения основных поверхностей, на которых они расположены; более высокой производительностью по сравнению с другим порядком за счет сокращения длины обработки и величин снимаемых припусков; спокойными, без ударов о ранее обработанные поверхности условиями резания; уменьшением объема слесарной обработки для снятия заусенцев.

Другой особенностью формы детали, оказывающей влияние на порядок ее обработки, является вид взаимного пересечения поверхностей. На основе анализа опыта проектирования и реализации технологических процессов обработки деталей различных классов сформулируем утверждение, определяющее ряд практических рекомендаций, учитываемых при формировании последовательности операций.

Утверждение 5. Порядок обработки пересекающихся поверхностей устанавливается таким, чтобы уменьшить увод инструмента и вероятность его поломки, ослабление жесткости детали и дополнительные затраты на слесарную обработку.

На рис. 18.3 показаны некоторые часто встречающиеся виды пересечений поверхностей и условия, определяющие порядок их обработки.

Вид пересечения поверхностей		Эскиз	Последовательность обработки поверхностей
Под прямым углом	с пересекающимися осями	d_2, $П_2$, $П_1$, d_1	$d_1 > d_2$: 1 — $П_1 \cdot П_2$; 0 — $П_2 \cdot П_1$
	с перекрещивающимися осями	h, d_1, $П_1$, $П_2$, d_2	$d_1 > d_2$, $h > kr_1$: 1 — $П_2 \cdot П_1$; 0 — $П_1 \cdot П_2$
Под произвольным углом		d_2, α, d_1, $П_1$	$d_1 > d_2$, $\alpha < \alpha_{доп}$: 1 — $П_2 \cdot П_1$; 0 — $П_1 \cdot П_2$
С параллельными осями		d_1, d_2, h, $П_1$, $П_2$	$d_1 > d_2$: 1 — $П_2 \cdot П_1$; 0 — $П_1 \cdot П_2$

Рис.18.3. Влияние вида пересечения поверхностей на последовательность их обработки

Для конкретного предприятия этот состав может быть изменен как за счет включения новых сочетаний, так и за счет исключения некоторых видов, не характерных для данного предприятия.

Важными факторами, оказывающими влияние на порядок операции в чистовых и получистовых этапах являются характер выполняемых обрабатываемыми поверхностями функций и структура размерных связей. На основе исходной посылки о совместимости элементов технологического процесса сформулируем утверждение определяющее условия совмещения технологических баз с конструкторскими.

Утверждение 6. Граф *Т(П, L)* технологических размерных связей операций этапа совместим с графом К (В, *L)* конструкторских размеров детали, если выполняются следующие условия: множество вершин *П* графа *Т(П, L)* является подмножеством *В* графа *К(В, L)* т. е. $П \subset В$; дуги графа *Т(П, L)* связывают те же вершины что и ребра графа *К(В, L);* ориентация дуг графа *Т(П, L)* осуществлена в соответствии со схемой размерных связей детали, т. е. от поверхностей основных конструкторских баз как корневых вершин в направлении к остальным обрабатываемым поверхностям.

В технологическом графе Т (П, *L)* конструкторские и технологические размеры образуют контуры, замыкающими звеньями которых служат чертежные размеры и припуски. Из теории графов известно, что величина любого звена контура равна алгебраической сумме составляющих его звеньев. Отсюда вытекает следующее положение

Утверждение 7. Погрешность замыкающего звена технологического размерного контура равна алгебраической сумме погрешностей составляющих звеньев

$$\Delta x_i = \sum_{j=1}^{k} \Delta b_j.$$

Из утверждения следует, что допуск чертежного размера Δx_i распределяется по технологическим звеньям контура Δb_j. Чем больше звеньев в контуре, тем больше при прочих равных условиях необходимо ужесточить допуски Δb_j на технологические звенья для обеспечения требуемого допуска на чертежный размер.

Следствие 2. При совмещении технологических баз с конструкторскими все замыкающие Δx_i и составляющие Δb_j дуги *Т* (П, *L)* образуют двухзвенные контуры

$$П_i \Delta x_i П_{i+1} \Delta b_i П_{i+1},$$

в каждом из которых, согласно утверждению 6, $\Delta x_i = \Delta b_i$ т. е. погрешности от несовмещения баз будут отсутствовать.

Из графа размерных связей операций этапа (см. рис. 18.1) видно, что базовые поверхности любой операции, обеспечивая заданную точность расположения детали, сами не являются обрабатываемыми. Отсюда вытекает следующее положение.

Утверждение 8. Базовые поверхности рассматриваемой операции должны быть обработаны с требуемой точностью в предшествующих операциях. Это положение справедливо и при проектировании сложных операций с несколькими переустановками детали. Следующее утверждение регламентирует последовательность операций на основе анализа графа *Т(П, L).*

Утверждение 9. Первой по порядку выполняется операция, имеющая в графе *Т(П, L)* наибольшую степень вершины, т. е. связанная с другими операциями наибольшим числом размеров. Порядок обработки других операций определяется по графу *Т(П, L),* так, чтобы их базовые поверхности были обработаны в предшествующих операциях.

Введем понятие вектора обработки поверхности выбранным методом, понимая под ним положение режущего инструмента относительно обрабатываемой поверхности и траекторию движения подачи. Сформулируем следующее утверждение, определяющее положение детали в рабочей зоне станка.

Утверждение 10. Деталь в рабочей зоне должна занимать такое положение, при котором векторы обработки V_i поверхностей, входящих в состав операции, были бы
параллельны и направлены в одинаковые стороны с вектором рабочих перемещений шпинделя, суппортов или стола панка $V_{ш}$, $V_{ст}$.

Это утверждение характеризует пространственную совместимость обрабатываемой детали с видами и направлением перемещений рабочих органов станка.

Выбор базовых поверхностей детали и распределение функций между ними определяются не только принципом совмещения баз, но и такими факторами, как форма, размеры, взаимное расположение обрабатываемых и базовых поверхностей и др. Выбранные установочные базы должны обеспечивать хорошую устойчивость детали в приспособлении при обработке; простоту установочно-зажимного приспособления; удобство установки детали; хорошие условия удаления стружки и другие эксплуатационные факторы.

Указанные показатели обобщенно характеризуют затраты на технологическую совместимость.

В одном случае базовые поверхности, выбранные на основе принципа совмещения баз, будут отвечать указанным выше требованиям, в других случаях такое решение приведет к чрезмерному усложнению приспособления, недостаточной устойчивости детали при обработке или к неудобству ее установки.

В связи с этим окончательный выбор технологических баз в операциях этапа представляет собой сложную многокритериальную задачу и производится на основании следующего утверждения.

Утверждение 11. Наиболее рациональный вариант технологических баз и связанной с ним последовательности операций в этапе должен быть таким, чтобы по всей совокупности операций обеспечить минимальное значение интегрального показателя качества

$$E = \sum_{i=1}^{n} (K_1 W_{\text{ис}} + K_2 W_{\text{ус}} + K_3 W_{\text{пр}} + K_4 W_{\text{экс}})_i,$$

где $i = 1,2,\ldots, n$ — число операций в этапе; $W_{\text{ис}}$, $W_{\text{ус}}$, $W_{\text{пр}}$, $W_{\text{экс}}$ — показатели, учитывающие соответственно влияние выбранных баз на увеличение стоимости обработки при несовмещении конструкторских баз с технологическими, устойчивость детали при обработке, удобство установки детали, удаление стружки и другие эксплуатационные факторы; K_1-K_4—весовые коэффициенты, с помощью которых задается степень важности локальных критериев W.

Коэффициенты K_1-K_4 принимают положительное или отрицательное значение (1 - (—1)) в зависимости от того, необходимо максимизировать или минимизировать тот или иной локальный критерий. Выбор значений локальных критериев W и весовых коэффициентов K для деталей сложной формы представляет собой трудноформализуемую задачу и должен производиться в режиме диалога. Возможны два варианта структуры диалога. В первом по заданным программам ЭВМ формирует базы и последовательность операций в этапе в соответствии с принципом совмещения баз. Результаты проектирования отображаются на экран. От технолога требуется откорректировать исходный вариант маршрута с учетом влияния всех локальных критериев. Во втором случае технолог составляет несколько вариантов схем базирования. По каждому варианту ЭВМ рассчитывает ожидаемые погрешности от несовмещения баз. Технолог проставляет значения W_j, K_j, а затем с учетом значений интегральных показателей E производит окончательный выбор варианта базирования и последовательности операций в этапах.

В ряде случаев минимизацию интегрального критерия качества, а следовательно, и затрат на технологическую совместимость обрабатываемой детали с выбранными типами оборудования обеспечивает правило постоянства установочных баз в операциях с однотипной схемой базирования. Например, в деталях класса «валы и круглые стержни» в качестве постоянных установочных баз при обработке наружных поверхностей вращения, резьб, шлицев, зубьев принимаются центровые отверстия. В операциях обработки плоскостей, пазов, нецентральных отверстий при назначении баз стараются придерживаться принципа совмещения баз. При обработке

корпусных и других деталей на агрегатных станках и автоматиче- - ских линиях принцип постоянства баз особенно эффективен, так как он обеспечивает однородность конструкции установочно-зажимных приспособлений и требуемую точность обработки.

18.2. Общая структура фреймов алгоритмов синтеза маршрута обработки деталей

На основе сформулированных утверждений построим фрейм структурной и алгоритмической модели процесса проектирования маршрута обработки деталей в различных этапах технологического процесса. Фрейм алгоритма построения маршрута состоит из следующих основных блоков:

1) определение признаков расчленения $\pi_1, \pi_2, \ldots, \pi_k$;

2) расчленение множества методов обработки поверхностей в этапе на укрупненные операции

$$\rho(\pi_i) : \{\mu_j\} \rightarrow \{A_q^{\text{у}}\};$$

3) если деталь «тело вращения», то 4, иначе — 6;

4) каждой укрупненной операции присвоить ранг *r*, равный наибольшему рангу поверхности, входящей в ее состав:

$$r_{\text{у}} = \max\{r_k\};$$

5) упорядочение укрупненных операций в порядке возрастания их ранга

$$У(r) : \{A_q^{\text{у}}\} \rightarrow \langle A_q^r \rangle;$$

6) дифференциация укрупненных операций на простые

$$\mathscr{P} : A_{\text{у}} \rightarrow \{A_i\};$$

7) выбор типа оборудования для каждой простой операции;

8) определение положения детали в зоне обработки Ψ : $Д_{но} \rightarrow Д_{ор}$;

9) определение степеней свободы детали, фиксируемых при базировании;

10) выбор технологических баз и схемы базирования детали;

11) формирование графа размерно-точностной структуры технологического процесса в этапе;

12) определение последовательности операций и графа функциональной структуры технологического процесса.

В соответствии с утверждением 1 модулем *1* производится формирование признаков расчленения $П_q=\{\pi_1, \pi_2, \ldots, \pi_k\}$ множества переходов этапа на укрупненные операции с учетом технологических возможностей оборудования и опыта проектирования маршрута обработки деталей определенного класса. Для каждого класса деталей

формируется свой набор признаков $П_{qk}$. Так, для деталей класса «тела вращения» признаками расчленения будут коды следующих подклассов поверхностен: наружные и внутренние поверхности основного контура, плоскости, пазы, нецентральные отверстия, окна, шпоночные канавки в отверстиях, шлицы, зубья и др. Указанные виды поверхностей в конкретной детали объединяются в самостоятельные укрупненные операции.

Расчленение исходного множества $M_и$ методов обработки поверхностей в этапе на укрупненные операции производится стандартным модулем, представляющим собой подпрограмму сортировки массива M_r по совокупности признаков $П_i$.

Дифференциация укрупненных операций на простые производится на основе утверждения 2 и осуществляется модулями определения рационального состава простых операций, выбора типа оборудования и установочных баз для выполнения каждой простой операции.

В укрупненную операцию входят участки и поверхности, ранг относительного взаимного расположения которых может быть различным. Например, в детали (см. рис. 18.2) отверстие *4* первого ранга, отверстие *3* второго ранга. В соответствии с утверждением 4 для их обработки в этапе будут сформулированы две отдельные операции, разделенные другими операциями. Такая последовательность с точки зрения получения требуемой точности приемлема, но экономически невыгодна, так как приводит к увеличению числа операций и возвратным потокам деталей в цехе.

Для устранения указанных противоречий модулем *4* каждой укрупненной операции присваивается ранг, равный наибольшему рангу поверхности, входящей в ее состав. В рассматриваемом примере укрупненной операции обработке нецентральных отверстий необходимо присвоить ранг, равный двум ($r = 2$).

Упорядочение укрупненных операций в порядке возрастания их ранга производится модулем *5*. В результате будет получен частичный порядок выполнения этих операций, отражающий влияние на маршрут обработки указанной характеристики формы детали.

В частично упорядоченном множестве встречаются группы укрупненных операций, имеющие одинаковый ранг. Например, операции обработки плоскостей, пазов, зубьев, нецентральных отверстий в деталях класса «тела вращения» могут быть одного ранга.

Установление рациональной последовательности укрупненных операций одного ранга производится на основе минимизации затрат на технологическую совместимость операций маршрута, т. е. с учетом минимизации затрат, связанных с возможным несовмещением конструкторских баз с технологическими, сложностью установочно-

зажпмпых приспособлений и расстановкой оборудования в цехах завода.

В тех случаях, когда обрабатываемые поверхности не связаны жесткими допусками и техническими требованиями или их величина настолько велика, что не оказывает влияния на последовательность обработки, последняя должна учитывать расстановку оборудования в цехе и прогрессивные технологические традиции проектирования технологии на конкретном предприятии.

Использование в фреймах алгоритмов опыта проектирования технологии на конкретном предприятии дает возможность учесть влияние организационных и других факторов на последовательность операций. Например, на ряде станкостроительных заводов обработка наружных резьб выносится в конец этапа, а при отсутствии термообработки — в конец технологического процесса. Это делается для того, чтобы при установке детали на других операциях и транспортировке не повредить резьбу. На других предприятиях этот фактор не принимают во внимание, но учитывается ряд других особенностей.

Для некоторых классов деталей можно построить типовые фреймы алгоритмов определения последовательности простых операций с учетом технических требований на точность взаимного расположения поверхностей и передовых технологических традиций.

На рис. 18.4 показан алгоритм формирования последовательности операций обработки различных поверхностей одинакового ранга для деталей класса «валы и круглые стержни».

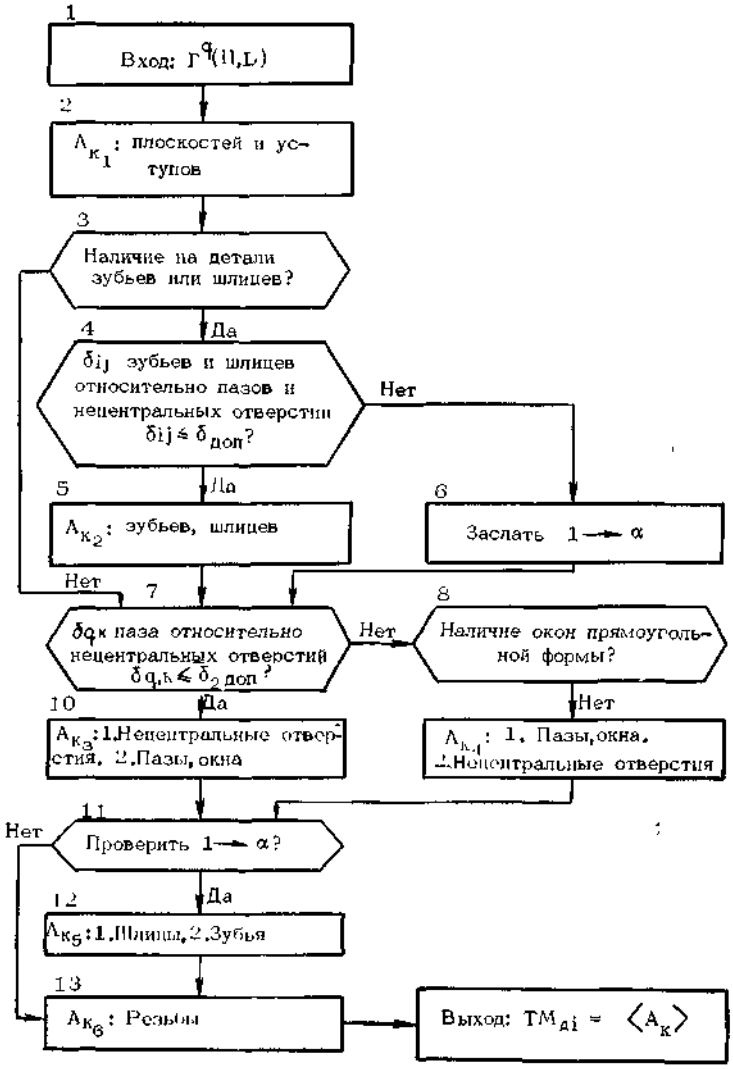


Рис. 32. Алгоритм формирования последовательности операций обработки различных поиорчпостсй одинакового ранга для деталей класса «валы»

Пазы и нецентральные отверстия, связанные техническими требованиями с зубьями или шлицами, выполняются после их обработки с базированием по зубу или шлицу (блоки *3, 4, 5).* При

отсутствии технических требовании шлицы и зубья обрабатываются в последнюю очередь (блоки *6, 12).*

Оператором 7 проверяется точность взаимного расположения пазов относительно нецентральных отверстий. Если $\delta_{q1k} \leq \delta_{2доп}$, то вначале выполняются нецентральные отверстия, затем пазы, окна (блок *10).*

При наличии на детали прямоугольных окон вначале следуют операции обработки нецентральных отверстий, с тем чтобы в этой же операции произвести сверление отверстий под последующую обработку окон (блоки *8, 9).*

Для учета конкретной производственной обстановки и внесения изменений этот алгоритм выполнен в виде быстросменного модуля.

18.3. Фреймы алгоритмов формирования маршрута обработки основных поверхностей деталей класса «тела вращения»

18.3.1. Дифференциация укрупненных операций в черновых этапах (подфреймах)

На основе сведений о характере черновой обработки наружного контура детали и группы центральных отверстий, полученных при проектировании на предыдущем уровне, необходимо определить состав и последовательность операций $S^*_ф$ *(C, A),* обеспечивающих минимальные суммарные затраты на черновую обработку детали:

$$S^*_ф\,(C,\ A) = \min_{S^* \in S} \sum_{i=1}^{n} E_i,$$

где E_i — затраты на i операцию; S — множество допустимых вариантов функциональной структуры черновой обработки детали.

Затраты E_i на этом уровне проектирования выражаются эвристическими зависимостями, связывающими размер партии и ряд логических условий. Общая задача образования фрейма структуры операций разделяется на две частных: выбор схемы обработки детали и ее детализация. Под схемой обработки для деталей класса «тела вращения» будем понимать порядок обработки наружного контура и группы центральных отверстий. Возможныдве схемы обработки деталей этого класса. По первой вначале обрабатываются поверхности наружного контура вала, а затем центрального отверстия $CX^1 \rightarrow A_{нп} \cdot A_{по}$. Для второй схемы характерен обратный порядок: вначале выполняется операция сверления центрального отверстия $A_{цо}$, затем точение

наружных поверхностей $A_{нп}$ и в последнюю очередь расточка отверстий со стороны торцов вала $A_{отв}$:

$$CX^2 \rightarrow A_{цсо} \cdot A_{нп} \cdot A_{отв}.$$

Условием выбора схемы обработки является целесообразность выделения сверления центрального отверстия в самостоятельную операцию. Это характерно для длинных пустотелых валов с диаметром центрального отверстия $d_{цот} > (d_{доп} \approx 20$ мм) и $l/d > (B_{ДОП} \approx 25)$. Алгоритм выбора схемы обработки задается следующей операторной формулой:

$$M_{сх} \rightarrow H_1^Б \cdot P_2^{\uparrow 5} \cdot P_3^{\uparrow 5} \cdot CX_4^1; \quad \overset{2,3}{\downarrow} CX_5^2,$$

где $H_1^Б$ — алгоритм назначения операций подготовки первичных технологических баз (обработка торцов, центрование); P_2, P_3— логические операторы, проверяющие условия выбора схемы обработки:

$$P_2 \rightarrow (d_{цсо} \leqslant d_{доп}); \quad P_3 \rightarrow (l/d \leqslant B_{доп}).$$

Только при невыполнении условий P_2 и P_3 назначается вторая схема обработки, в остальных случаях — первая. Разновидности CX^1 определяются необходимостью выполнения черновой обработки наружных поверхностей и отверстий в этапе Э2. Кроме того, при двухсторонней ступенчатости центрального отверстия расточка поверхностей каждой стороны выделяется в отдельные операции. Алгоритм, учитывающий эти особенности, описывается следующей операторной формулой:

$$CX^1 \rightarrow B_0 P_1^{\uparrow 3} \cdot A_{2пн} \cdot \overset{1\downarrow\uparrow 8\downarrow 7}{P_3} \cdot B_4(i) \cdot \overset{\uparrow 7}{P_5}(i) \cdot A_{6отв}(i) \cdot \overset{7\downarrow\uparrow 4}{П_7}(i) \cdot \overset{\downarrow 3}{Я_8},$$

где B_0, $Я_8$— входные и выходные операторы алгоритма; P_1 —оператор, проверяющий необходимость черновой обработки наружных поверхностей; P_3 — оператор проверки вида детали (вал пустотелый или сплошной); $B_4(i)$— оператор выделяет для анализа очередную i сторону детали; $P_5(i)$ —оператор проверяет необходимость выделения черновой обработки отверстий i стороны в отдельную операцию; $П_7(i)$ — оператор цикла по i повторяет операторы $B_4(i)$-$A_{6отв}(i)$ для следующей стороны детали.

Алгоритм функционирует следующим образом. Если оператор P_1 удовлетворяется, то засылается операция $A_{нп}$, в противном случае управление передается P_3. Для пустотелых валов (P_3=1) посредством группы операторов $B_4(i)$-$П_7(i)$ анализируется каждая сторона центрального отверстия и, если необходимо ($P_5(i)$ = 1), назначаются черновые операции для их обработки. При анализе сплошных валов

(P_3 = 0) указанная группа операторов опускается и управление передается выходному оператору *Я₈*. Для второй схемы обработки этот алгоритм используетсякак отдельный модуль

$$CX^2 \rightarrow A_{1бш} \cdot A_{2цсо} \cdot A_{CX_1},$$

где $A_{1бш}$ — токарная операция обработки базовых шеек под люнет; $A_{2цсо}$ - операция сверления центрального отверстия. Состав операций подготовки первичных технологических баз зависит от размеров заготовки и величины партии деталей. Фрезерно-центровальная операция назначается, если заготовка по размерам d_3, l_3 вписывается в рабочую зону станка и величина партии $n \geq n_{доп}$, т. е. выполняется сложное логическое условие

$$P_у = \begin{cases} d_3 \leqslant d_{доп}, \\ l_3 \leqslant l_{доп}, \\ n \geqslant n_{доп}. \end{cases}$$

Для более крупных деталей $d_3 = d^1_{доп} \div d^2_{доп}$ и $l_3 = l^1_{доп} \div l^2_{доп}$ торцы фрезеруются в горизонтально-фрезерной операции. Зацентровка торцов производится в отдельной центровальной оцерации. Обработка торцов и их зацентровка для валов еще больших размеров производятся в токарной операции $A_{ток}$. Общий алгоритм назначения операций подготовки первичных технологических баз записывается следующей операторной формулой:

$$A_{пб} \rightarrow B_1 \overset{\uparrow 4}{P^у_2} \cdot A_{3фц}; \quad \overset{2\downarrow \uparrow 7}{P^у_4} \cdot A_{5ф} \cdot A_{6цен}; \quad \overset{4\downarrow}{A}_{ток} \cdot E_8,$$

где $P^у_2$ сложное логическое условие назначения фрезерно-центровальной оцерации $A_{фц}$; $P^у_4$— логическое условие назначения двух операций: фрезерной и центровальной *($A_ф$,* $A_{цен}$).

Для различных предприятий параметры выбора решений в алгоритмах могут несколько отличаться. Учет конкретных особенностей предприятия осуществляется настройкой соответствующих модулей или их корректировкой.

18.3.2. Дифференциация укрупненных операций в получистовых этапах (подфреймах)

Образование фреймов формирования токарных операций обработки деталей класса «втулки, диски» производится на основе анализа следующих схем обработки. Первая схема предусматривает применение протягивания как наиболее производительного метода обработки точных центральных отверстий и последующей обработки

наружных поверхностей на центровых нормализованных оправках. Для второй схемы характерно использование одной или нескольких токарных операций при консольном закреплении детали. Третья схема обработки связана с применением специализированных установочно-зажимных оправок и инструментальных наладок. Алгоритм выбора схемы обработки описывается операторной формулой

$$M_{сх} \rightarrow B_1 \cdot \underset{\substack{\downarrow \\ CX^2}}{P_2^{пр}} \cdot \Pi_3 P_4^{\uparrow 7} \cdot CX_5^1; \quad \overset{4\downarrow\uparrow 6}{P_7} CX_8^3,$$

где $P^{пр}_2$ — оператор, проверяющий условия протягивания; Π_3 — поиск в массиве протяжки, удовлетворяющий конкретным условиям; P_4 — оператор, проверяющий результат поиска (выбрана ли протяжка); P_7 — оператор проверки размера партии $n > n_{2доп}$.

Первая схема обработки назначается для деталей с центральным сквозным отверстием, если его размеры и точность удовлетворяют условиям протягивания

$$P_2^{пр} = \begin{cases} Д_{цсо} = Д_1 \div Д_2, \\ T_{цсо} \leqslant 3, \\ L_{дет} \leqslant L_{доп}, \\ l_{цсо} \leqslant l_{доп}, \\ n > n_{1доп}. \end{cases}$$

В тех случаях, когда условия протягивания не удовлетворяются, назначается вторая схема обработки, третья — при отсутствии в массиве применяемости необходимой нормализованной или ранее спроектированной протяжки, когда размер партии достаточен для заказа специального приспособления (специальной оправки).

В зависимости от формы и точности центральных отверстий возможно несколько разновидностей первой схемы обработки (табл. 10).

Таблица 10

Табличный алгоритм дифференциации токарных операций обработки втулок

Условия выбора токарных операций	Виды токарных операций			
	Токарная (*ЦСО*)* в *Э2* Протяжная Токарная (*НП*)**	Токарная (*ОТВ*)*** Протяжная Токарная (*НП*)	Токарная (*ЦСО*) в *Э2* Протяжная Токарная (*НП*) Токарная (*ОТВ*)	Токарная (*ЦСО*) Токарная (*НП*) Токарная (*ОТВ*)
Форма центрального сквозного отверстия (*ЦСО*)	Гладкое	Ступенчатое с одной или двух сторон		
Точность *ЦСО* $T_{цсо} \leqslant 2 \div 3$	Да	Да	Нет	Нет
Точность остальных полуоткрытых отверстий $T_i = 2 \div 3$	—	Нет	Да	Да
Возможность установки детали на оправке	Да	Да	Да	Нет

* *ЦСО* — центральное сквозное отверстие;
** *НП* — наружные поверхности;
*** *ОТВ* — отверстия со стороны торцов детали.

Одна разновидность применяется для деталей с точными центральными сквозными отверстиями и заключается в том, что на основе начерно обработанного отверстия в этапе Э2 в получистовом этапе Э4 производится его протягивание. Затем следует токарная обработка наружных поверхностей (см. 1-й столбец табличного алгоритма).

Вторая разновидность характерна для деталей, у которых центральное отверстие имеет одностороннюю или двухстороннюю ступенчатость и неточные отверстия со стороны торцов. Структура операций обработки основных поверхностей здесь аналогична ранее рассмотренной, только в этапе *Э4* добавляется токарная чистовая обработка отверстий (см. 2-й столбец табличного алгоритма). Третья схема используется при обработке деталей с точными отверстиями со стороны торцов и неточным центральным сквозным отверстием. Если параметры отверстия, кроме точности, дают возможность обработать деталь на оправке, то к составу операций второй разновидности схемы обработки добавляется токарная операция чистовой обработки отверстий со стороны торцов на базе наружных поверхностей с выверкой детали по протянутому отверстию (см. 3-й столбец табличного алгоритма).

Если длина центрального сквозного отверстия недостаточна для установки детали по оправке, то в черновом этапе Э2 производится обработка только центрального отверстия под протягивание. В получистовом этапе *Э4* выполняется протяжная операция, а затем

токарная обработка наружных поверхностей. Точение и шлифование точных расточек со стороны торцов детали производятся после окончательной обработки наружных поверхностей с использованием их в качестве базы.

В серийном производстве при больших размерах выпуска изготавливаются специальные оправки, а детали обрабатываются по обычной схеме: растачиваются начисто отверстия, деталь устанавливается на специальную оправку и производится точение наружных поверхностей.

Третья разновидность схемы обработки состоит в том, что для деталей, размеры центрального отверстия которых превышают максимально допустимые по условиям протягивания, в получистовом этапе обработка центральных отверстий производится точением. Остальные операции такие же, как и в предыдущих случаях (см. 4-й столбец таблицы). Получистовая обработка простых деталей производится в одной совмещенной токарной операции, в которой выполняется точение наружных поверхностей и отверстий.

18.3.3. Дифференциация укрупненных операций в чистовых этапах (подфреймах)

В основу дифференциации укрупненных шлифовальных операций для деталей класса «втулки, диски» положены три схемы обработки. По первой схеме вначале обрабатываются центральные отверстия, а затем наружные поверхности при установке детали на нормализованных оправках:

$$A_{у} \rightarrow A_{в} \cdot A_{ц}.$$

Вторая схема обработки предусматривает шлифование наружных и внутренних поверхностей с каждой стороны детали в отдельных операциях A_1, A_2 и включение между ними плоскошлифовальной операции по обработке крайнего торца детали как базы для операции A_2:

$$A_{у} \rightarrow A_{1} \cdot A_{тор}^{БП} \cdot A_{2}.$$

Эта схема имеет несколько разновидностей в зависимости от типов обрабатываемых поверхностей и их расположения относительно шейки максимального диаметра.

В третьей схеме обработки предполагается использование специализированных установочно-зажимных приспособлений.

Наиболее простой и рациональной является первая схема обработки. Вторая схема требует большего объема работ по подготовке базовых

поверхностей и выверке детали. Поэтому она применяется при небольших размерах партии, когда по конфигурации центрального отверстия невозможны нормализованные оправки. Если партия деталей достаточно велика, то более рационально заказывать специализированные установочно-зажимные приспособления, т. е. применять третью схему обработки.

Алгоритм выбора схемы обработки описывается операторной формулой

$$M_{сх} \rightarrow B_1 \cdot P_2^{\uparrow 7} \cdot \Pi_3 \cdot P_4^{\uparrow 7} \cdot P_5^{\uparrow 7} \cdot CX_6^{1}; \quad \overset{2,4,5}{\downarrow} P_7^{\uparrow 9} \cdot CX_8^{2}; \; \overset{\downarrow 7}{CX}{}_9^{3},$$

где P_2 — оператор проверки размера и точности центрального сквозного отверстия $D_i \leq D_{доп}$, $T \leq 3$; Π_3— поиск нормализованных оправок; P_4, P_5, P_7 — операторы, проверяющие, выбрана ли оправка, возможность установки детали на оправке и величину партии деталей $n_1 \geq n_{доп}$. Взаимодействие операторов осуществляется в следующем порядке. С помощью операторов P_2, Π_3 производится поиск оправки в массиве нормализованных и ранее спроектированных оправок. Если результаты поиска положительны, оператором P_5 производится проверка возможности установки детали на оправке без перекоса.

Решение указанных вопросов базируется на эвристических оценках, полученных обобщением опыта технологического проектирования и изготовления деталей. К числу таких оценок можно отнести отношение длины отверстия к его диаметру $l_{отв}/D_{отв}$ и величину консоли обрабатываемой поверхности относительно торца посадочного отверстия. Если операторы P_2-P_5 выполняются, то назначается первая схема обработки. В противном случае управление передается оператору P_7, который при малом размере партии $n<n_{доп}$ назначает вторую схему обработки, а при большой величине n — третью.

18.4. Образование фрейма автоматизации синтеза управляющих програм

18.4.1. Системная математическая модель перехода

Простой переход является основным структурным элементом операции. Он характеризуется совокупностью упорядоченных во времени основных и вспомогательных приемов, необходимых для обработки с заданными размерами и точностью одной элементарной, типовой или нормализованной поверхности одним простым инструментом. С системных позиций переход определяется функцией f, структурой $S_ф$, $S_в$, $S_п$ и набором параметров

$Z : П_р = \langle f, S_ф, S_п, S_п, Z \rangle$. Для операций, выполняемых на станках с ручным управлением, структура перехода не рассматривается.

Функция основного перехода заключается в преобразовании Ψ обрабатываемой поверхности из одного промежуточного состояния σ_{i-1} в другое σ_i и описывается отображением $\Psi_i : \sigma_{i-1} \rightarrow \sigma_i$. Преобразование Ψ_i определяет вид технологического перехода (точить, фрезеровать, сверлить, шлифовать и т. д.). Состояние σ_i полученное в результате выполнения перехода, характеризуется графом размерных связей обрабатываемой поверхности с базовой W_i *(П, L)* и набором параметров, описывающих форму — *КФ,* размеры — *Р* и физико-механические свойства — *МС* поверхности $Z_i = \{КФ, Р, МС\}$:

$$\sigma_i = \langle (КФ, Р, МС), W_i (П, L) \rangle .$$

Дуги графа W_i (П, L) имеют ориентацию от вершин, принятых за технологические базы к обрабатываемой поверхности (рис. 18.5).

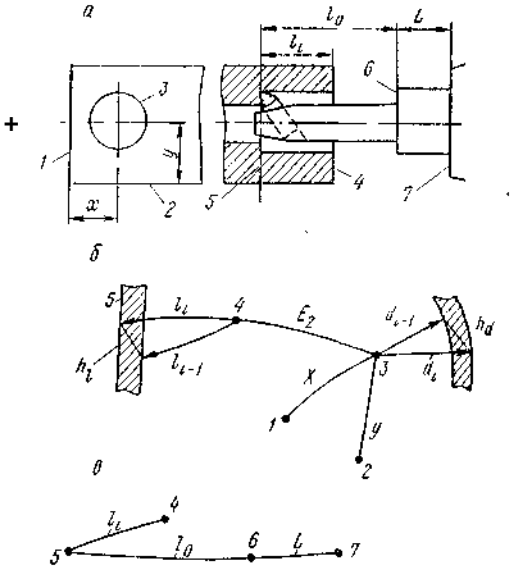


Рис. 18.5. Схема размерных связей перехода: *а*—операционный чертеж; б — графы размерных связей σ_{i-1} и σ_i состояний обрабатываемой поверхности; *в* — граф пространственной структуры перехода

Состояние σ_{i-1} задается характеристиками Z_{i-1} и W_{i-1} (П, L), полученными на предшествующем переходе.

Припуски на обработку h_d и h_z связывают размеры σ_{i-1} и σ_i состояний в контуры $K_d = d_{i-1}, d_{i+1}, h_d$ и $K_z=l_{i-1}, l_i h_i$. При выполнении i — 1 и i переходов от одинаковых баз контуры K_d, K_z получаются трехзвенными. Одно звено соответствует размеру на i — 1 переходе, второе — на i переходе, а припуск выступает как замыкающее звено контура. Если переходы выполняются от различных баз, то количество звеньев в контурах возрастает на величину, равную числу несовмещенных баз.

Функция вспомогательного перехода описывается отображением $\Psi_i^{в} : \sigma_{i-1}^{в} \rightarrow \sigma_i^{в}$, в котором $\Psi^{В}_i$ обозначает вид перехода (установить деталь, сменить инструмент, переместить деталь с одной позиции на другую и т. д.), а $\sigma^{в}_{i-1}$, $\sigma_i^{в}$ — состояния указанных элементов СПИД до и после выполнения перехода. При этом указываются только те параметры элементов, значения которых изменились в результате осуществления перехода.

Состояние детали после перемещения ее с одной позиции на другую характеризуется новым положением в пространстве базовой системы координат детали. Состояние СПИД, возникшее в результате смены инструмента, задается описанием типоразмера нового инструмента и его настрочных размеров.

В техническую характеристику основного перехода входит задание элементов СПИД, участвующих в его выполнении, и параметров процесса обработки поверхности. К таким элементам относятся типоразмеры режущего, вспомогательного и измерительного инструментов *РИ, ВИ, ИИ.* Параметры основного приема — это величина припуска h, число переходов k, скорость V, подача S, глубина резания t, машинное и вспомогательное время $T_м$, $T_в$:

$$Z_{ос} = \{РИ,\ ВИ,\ ИИ,\ h,\ k,\ V,\ S,\ t,\ T_м,\ T_в\}.$$

Вспомогательный переход «установить» характеризуется способом установки детали—*СУ* (вручную, подъемником, автооператором, краном), типоразмером установочно-зажимного приспособления — *ПР,* усилием зажима — $P_З$, погрешностью установки — $\varepsilon_у$, временем установки — $t_у$

$$Z_в = \{СУ,\ ПР,\ P_з,\ \varepsilon_у,\ t_у\}.$$

В операциях, выполняемых на станках с цифровым программным управлением, автоматах и полуавтоматах, **переход разделяется на совокупность приемов и действий.** В связи с этим при анализе технологического процесса необходимо перейти к следующему

уровню расчленения и рассмотреть структуру основных и вспомогательных переходов, функцию и параметры каждого приема. Функциональная структура перехода представляет собой совокупность взаимосвязанных отношений $\sigma_{i-1}\psi_i\sigma_i$ между приемами ψ_i и состояниями σ_i элементов системы СПИД, полученными в результате их выполнения. Указанная совокупность отношений образует граф функциональной структуры перехода $S_ф(\sigma, \psi)$. Его вершинами служат состояния σ_i элементов СПИД, а множество дуг ψ обозначает основные и вспомогательные приемы, посредством которых система СПИД переходит из одного состояния в другое. На рис. 18.6 показаны схема и граф функциональной структуры фрезерного перехода по обработке контура «кармана».

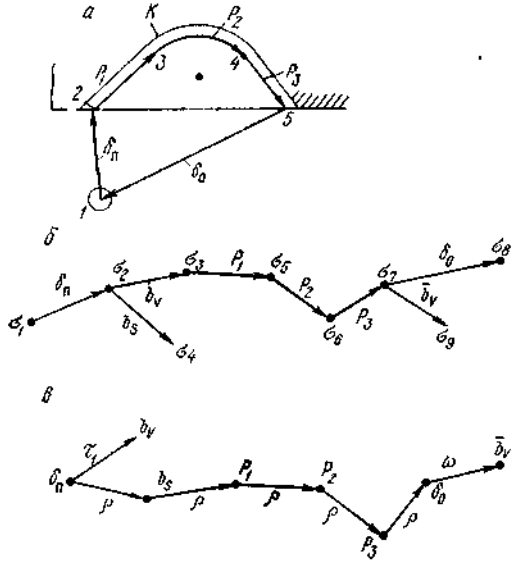


Рис. 18.6. Структура перехода фрезерной обработки контура «к»:
а — схема перехода; б — граф функциональной структуры перехода;
а — граф временной структуры перехода

Граф $S_ф(\sigma, \psi)$ определяет частичную упорядоченность приемов в переходе, которая задается номерами состояний и ориентацией дуг. Однако он не дает точной взаимосвязи приемов во времени. Например, приемы b_v и b_s могут выполняться последовательно или одновременно. Точная взаимосвязь приемов во времени описывается графом временной структуры перехода.

Функциональная структура вспомогательного перехода, например «установить деталь» с помощью автооператора, представляет собой совокупность преобразований детали из неориентированного состояния $\sigma_н$ в ориентированное $\sigma_{ор}$ путем захвата детали *r*, перемещения ее в зону обработки π, совмещения базовых поверхностей детали с базовыми элементами приспособления (базирования δ) и закрепления з:

$$S_ф^у = \sigma_н r_1 \sigma_1 \pi_2 \sigma_2 \delta з \sigma_{0р},$$

где σ_1, σ_2, σ_3—состояния элементов СПИД после захвата детали автооператором, ее перемещения и базирования.

В графе $S_ф(\sigma, \psi)$ каждая его дуга $\sigma_{i-1}\psi_i\sigma_i$ характеризует преобразование элемента СПИД из одного состояниях другое и представляет собой функцию приема. В целом функциональная структура перехода образуется в результате теоретико-множественного объединения элементарных функций отдельных приемов

$$S_ф = f_1 \cup f_2 \cup \ \cdots \ \cup f_m = \bigcup_{i=1}^{m} f_i.$$

При выполнении приема система СПИД проходит ряд последовательных мгновенных состояний *σ(t),* каждое из которых задается набором переменных, характеризующих положение режущего инструмента и обрабатываемой детали, величину подачи в рассматриваемый момент времени. Узловыми будем называть состояния элементов системы СПИД, возникающие после завершения приема. При описании узловых состояний указываются не все параметры элементов СПИД, а только те, значения которых изменились в результате выполнения приема. Например, $\sigma_б$ после быстрого подвода инструмента к детали или $\sigma_р$ после обработки поверхности описывается координатами вершины резца или центра фрезы $\sigma_{б,р}=(X, Y, Z)$ в новом положении. В результате выполнения приемов «включение» или «переключение» скоростей и подач состояние $\sigma_{v,\ s}$ будет задано значениями чисел оборотов и подач.

Временная структура перехода представляет собой совокупность упорядоченных во времени приемов и описывается графом $S_В(\Psi, \Omega)$. В этом графе множеству вершин *Ψ* соответствуют основные и вспомогательные приемы, а множеству дуг *Ωii* — отношения, характеризующие методы совмещения приемов во времени (ρ, ω, τ). Рассмотрим описание временной структуры переходов различных видов.

Структура многопроходного перехода может быть задана посредством графа или соответствующей ему структурной формулой

$$S_{в} = \delta_{п} b_{v} \overline{H_i b_{si} R_i \bar{b}_{si} \delta_{oi} F(i)} \bar{b}_{v} k, \quad i = 1 \div 2,$$

где $б_п$, $б_0$—быстрый подвод и отвод инструмента; b_v, $\bar{b}_v$— включить и выключить скорость резания; b_s, $\bar{b}_s$—включить и выключить подачу; H_i — ЧИСЛО пробных проходов, необходимое для настройки инструмента на размер в i проходе; R, k—обработка и контроль поверхности. Логический оператор $F(i)$ показывает, что группу операторов $H_i b_{si} R_i \bar{b}_{si} \delta_{oi}$ необходимо повторить i раз. Структура переходов, в которых размеры получаются автоматически, описывается формулой

$$S_{в} = \delta_{п} b_{v} b_{s} R \bar{b}_{v} \delta_{o}.$$

Такие переходы служат основой для синтеза операций со сложной структурой, так как в них отсутствуют приемы H_i, k, выполняемые вручную.

Структура $S^{у}_{в}$ вспомогательного перехода «установить деталь» может осуществляться по-разному. В одних случаях все приемы выполняются последовательно, а в других базирование и закрепление детали производятся одновременно:

$$S_1^{у} = з_1 п_2 \delta_3 З_r,$$

$$S_2^{у} = з_1 п_2 (\delta_3 \omega З_r).$$

Аналогичным образом описывается структура других вспомогательных переходов.

На станках с ЧПУ часто реализуются переходы, в которых одним инструментом обрабатывается групаа поверхностей, например сверление нескольких одинаковых отверстий, обработка наружных контуров на токарных и фрезерных станках, фрезерование «колодцев, карманов, окон». Такие переходы будем называть одноинструментальными в отличие от многоинструментальных, выполняемых комбинированным инструментом.

Разновидности структур одноинструментальных переходов (рис. 18.7) образуются за счет разного состава и характера взаимосвязи вспомогательных переходов, необходимых для выполнения следующего простого перехода (перемещение детали с одной позиции на другую r, холостые ходы инструмента m):

$$\psi_1 = a_1,$$

$$\psi_2 = a_1 m_1 a_2, \ \ldots, \ m_{n-1} a_n,$$

$$\psi_3 = a_1 r_1 a_2, \ \ldots, \ r_{n-1} a_n,$$

$$\psi_4 = a_1 (m_1 \gamma r_1) a_2, \ \ldots, \ (m_{n-1} \gamma r_{n-1}) a_n.$$

Перемещение детали с одной позиции на другую	Холостые перемещения инструмента между переходами	
	Отсутствуют $m = 0$	Имеются $m > 1$
Отсутствуют (однопозиционная обработка $i = 1$)	Д, РИ	РИ, m, Д
Многопозиционная обработка $i > 1$	РИ, Д, г	m, РИ, Д, г

Рис. 18.7. Разновидности структур инструментальных переходов

В четвертой схеме инструментальный переход осуществляется с помощью холостых перемещений инструмента и детали, выполняемых последовательно, параллельно или со сдвигом фаз $\gamma=(\rho, \omega, \tau)$. На рис. 18.6 показана схема одноинструментального фрезерного перехода по обработке контура «кармана». Его временная структура описывается выражением

$$S_в = (\delta_п \tau_1 b_V)\, b_з R_1 R_2 R_3\, (\delta_0 \omega \bar{b}_V).$$

Пространственная структура перехода задается графом $S_п(B, L)$ и отражает наладочные размеры элементов инструментальной наладки, предназначенных для осуществления перехода. В графе $S_п(B, L)$ множеству вершин B соответствуют пары взаимосвязанных поверхностей (обрабатываемая поверхность, формообразующий элемент режущего инструмента, торец шпинделя и соприкасающаяся с ним поверхность вспомогательного инструмента и т.д.), а множеству дуг L

— наладочные размеры между ними. На рис. 18.5 показаны элемент инструментальной наладки и граф ее структуры.

Таким образом, системная математическая модель переходов, выполняемых на станках с ЧПУ и станках-автоматах и полуавтоматах, задается совокупностью соотношений, описывающих системные характеристики перехода на двух уровнях расчленения: перехода как целого и на уровне приемов и команд управлении станком

$$f_{\text{пер}} = \psi : \sigma_{i-1} \to \sigma_i,$$

$$S_{\text{пер}} = \{S_{\text{ф}}(C, \Psi),\ S_{\text{в}}(\Psi, \Omega),\ S_{\text{п}}(B, L)\},$$

$$Z_{\text{пер}} = \{Z_1, Z_2, \ldots, Z_k\},$$

$$f_{\text{пр}} = \lambda : \alpha_{j-1} \to \alpha_j,$$

$$Z_{\text{пр}} = \{m, m_2, \ldots, m_k\}.$$

Эта модель является основой для постановки и решения задач автоматизации синтеза траекторий обработки сложных контуров и участков детали на станках с ЧПУ.

18.4.2. Синтез траекторий фрезерной обработки сложных областей на станках с ЧПУ

Широкое распространение и эффективное применение станков с ЧПУ тесно связано с автоматизацией подготовки программ, управляющих этими станками. Известные системы подготовки управляющих программ основаны на предварительной разработке технологом траекторий движения инструмента и последующем применении проблемно-ориентированных языков, позволяющих символически описывать эти траектории и технологические команды, соответствующие отдельным элементам этой траектории. Практическое применение таких систем связано с большим объемом технологических работ, возлагаемых на пользователя.

Иной подход заключается в построении такой системы, в которой ручному описанию подвергаются только геометрия готовой детали и некоторая дополнительная информация технологического и производственного характера; сама же траектория обработки и все технологические команды синтезируются автоматически. Исходными данными для решения указанных задач служат полученные на предыдущем уровне проектирования операционной технологии сведения о типоразмерных станках, приспособлениях, инструментах и их технических характеристиках

$$СТ = \{Z_i^c\},\ i = 1 \div n_1,$$

$$ПР = \{Z_j^n\},\ j = 1 \div n_2,$$

$$ИН = \{Z_k^n\},\ k = 1 \div n_3;$$

функции перехода $\psi : \sigma_{i-1} \to \sigma_i$, включающей задание вида перехода (обработка по эквидистанте, строке), состояния σ_{i-1} и σ_i обрабатываемого контура до и после выполнения перехода

$$\sigma_{i-1} = \langle Z_{i-1}^0,\ W_{i-1}(П,\ L),$$

$$\sigma_i = \langle Z_i^0,\ W_i(П,\ L).$$

Задача заключается в том, чтобы определить функциональную $S_ф(C, \Psi)$ и пространственно-временную структуры сложного перехода по обработке заданного участка детали. Проектирование обработки по эквидистанте осуществляется от окончательного контура к центру области до тех пор, пока не будет построена последняя эквидистантная траектория.

При этом расстояния между последующими эквидистантами могут изменяться в зависимости от технологических особенностей обработки.

Различаются два основных метода обработки по эквидистанте. В первом методе обработка осуществляется от внутренней эквидистанты к внешней вдоль ранее построенных эквидистантных кривых (обработка по внутренним эквдистантам). При этом геометрические особенности контура и соседних эквидистантных кривых могут привести к формированию дополнительных перемещений, дорабатывающих участки поверхности, которые могут остаться необработанными. Во втором методе от самой внутренней эквидистанты строится последовательность внешних друг к другу эквидистант.

При проектировании траектории обработки по эквидистанте автоматически синтезируется математическое описание оставшейся области. Она может состоять из нескольких связных областей, каждая из которых в свою очередь может быть обработана тем или иным способом. Траектории холостых перемещений между обрабатываемыми областями синтезируются автоматически. При необходимости человек может принять участие в ее корректировке. Диалог технолога и ЭВМ осуществляется с помощью графических средств связи и функциональной клавиатуры дисплея.

Практическая реализация системы связана с решением большого комплекса сложных геометрических задач, не имевших ранее

удовлетворительного решения. Наиболее сложные задачи возникают при автоматическом синтезе траекторий обхода обрабатываемых поверхностей, назначении участков входа и выхода инструмента, выборе участков, обрабатываемых по эквидистанте или строке, траекторий холостых перемещений инструмента и т.д.

При построении данной системы в качестве языка программирования был использован геометрически ориентированный алгоритмический язык ФАП-КФ, представляющий собой расширение языка ФОРТРАН геометрическими переменными и операциями. Его применение позволяет существенно сократить сроки разработки системы за счет достаточно развитых геометрических средств языка, покрывающих основные геометрические задачи, возникающие при построении системы. Ниже рассматриваются наиболее характерные геометрические задачи, используемые в системе и связанные с синтезом траекторий обработки плоских областей, по эквидистанте и «по строке».

При обработке сложных областей на станках с ЧПУ приходится строить последовательное семейство эквидистант, покрывающих данную область. Каждая такая эквидистанта представляет собой траекторию движения центра фрезы. Исходные области могут быть многосвязными и несвязными. При этом приходится строить как внутренние, так и внешние эквидистанты. Если область имеет «узкие» участки, то при построении эквидистанты могут появляться самопересечения, которые свидетельствуют о наличии «петель». Это приводит к тому, что эквидистанта распадается на несколько контуров и область, ограниченная ею, может иметь совершенно иную структуру по сравнению с исходной.

Будем различать три разновидности эквидистант: теоретическую, усеченную и распадающуюся. Теоретическая представляет собой геометрическое место концов нормалей длиной δ, восстановленных в каждой точке исходной кривой. Если в некоторой точке кривой нормаль не существует, то на эквидистанте ей соответствует дуга окружности, которая может быть получена при замене этой точки окружностью бесконечно малого радиуса (рис. 18.8). Теоретическая эквидистанта может иметь самопересечения.

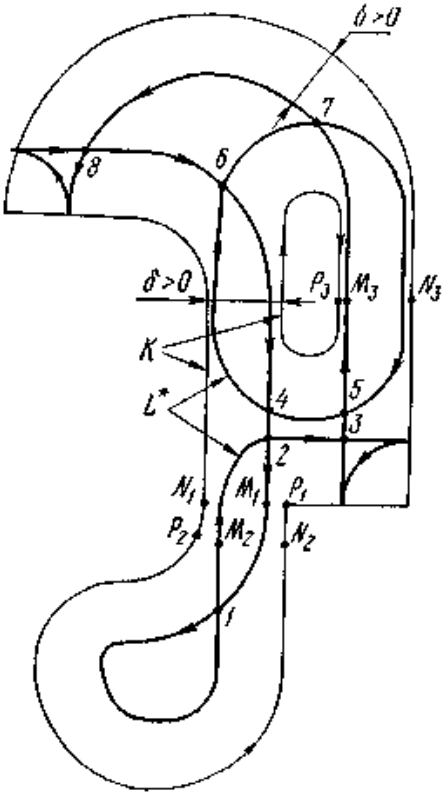


Рис. 18.8. Теоретическая эквидистанта

Усеченная эквидистанта является частью теоретической и отличается от нее тем, что не содержит «петель», которые получаются в результате пересечения элементов (отрезков прямых, дуг окружностей) теоретической эквидистанты, соответствующих смежным участкам исходной кривой. На рис. 18.8 ее можно получить, если убрать «петли» в вершинах *3* и *8*.

Распадающаяся эквидистанта представляет собой границу плоской области, все точки которой удалены от исходной кривой на расстояние, превышающее δ. Если строится внутренняя эквидистанта, то, начиная с некоторого значения δ, распадающаяся эквидистанта может не существовать (рис. 18.9).

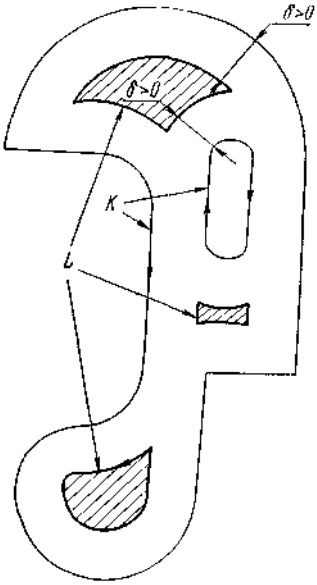


Рис. 18.9. Распадающаяся эквидистанта

При автоматическом проектировании траекторий обработки теоретическая и усеченная эквидистанты используются для построения траектории обработки, а распадающаяся — для построения области, оставшейся необработанной (рис.18.9).

Рассматриваемый ниже метод построения распадающейся эквидистанты можно применять к произвольным связным и несвязным областям. Обозначим через L^+ теоретическую эквидистанту, ориентированную в соответствии с ориентацией кривой *k*, ограничивающей заданную исходную область. Кривую L^+ можно рассматривать как ориентированный граф G^+ (U^+ — множество дуг этого графа), вершинами которого являются точки самопересечения этой кривой, дугами — участки кривой L^+, соединяющие эти вершины. Ориентация дуг графа совпадает с ориентацией соответствующих участков кривой L^+. Если конец нормали при движении вдоль *k* описывает некоторый участок кривой дважды и в разных направлениях, то будем считать, что существует две дуги.

Аналогично границу *L* распадающейся эквидистанты можно рассматривать в виде ориентированного графа *G*, вершинами которого являются точки самопересечения кривой *L*, а дугами — участки

кривой, соединяющие эти вершины. Тогда дуги графа G следует искать среди дуг графа G^+. Поэтому для решения задачи остается указать необходимые и достаточные условия, которым должны удовлетворять дуги графа G^+, принадлежащие графу G.

Отметим, что если среди дуг графа G^+ существуют противоположно ориентированные, то они не принадлежат распадающейся эквидистанте.

Обозначим через $\rho(k, m)$ кратчайшее расстояние от произвольной точки M до ориентированной кривой k. Условимся, что $\rho(k, m)>0$, если M расположена слева от k, и $\rho(k, m)<0$, если справа от k.

Пусть теперь M — произвольная внутренняя точка дуги $U \in U^+$. Тогда, если $\rho(k, m)=\delta$, то дуга U принадлежит распадающейся эквидистанте. В противном случае она ей не принадлежит.

Например, на рис. 18.8 точка M_1, принадлежащая дуге $U_{2,1} \in U^+$, удалена на расстояние δ от соответствующей точки N_1 на исходном контуре K. Однако кратчайшим расстоянием от точки M_1 до контура K является расстояние между точками M_1 и P, отличное от δ. Поэтому дуга $U_{2,1}$ не принадлежит распадающейся зквидистанте. Аналогично точка M_3 удалена от соответствующей ей точки N_3 на величину δ, а кратчайшее расстояние от M_3 до K — это отрезок между точками M_3 и P_3.

Разобьем граф G^+ на простые контуры, не содержащие внутри себя других дуг этого графа. Тогда каждый такой контур либо принадлежит эквидистанте, либо не принадлежит ей. Поэтому, установив принадлежность распадающейся эквидистанте одной дуги, тем самым устанавливаем принадлежность ей всего контура.

Другая характерная геометрическая задача связана с синтезом траектории построчной обработки плоской области. Пусть задана произвольная плоская область P, ограниченная ориентированными кривыми K. Область P назовем однозначной, в заданном направлении V, если любая прямая, параллельная V, пересекает границу области P не более чем в двух точках. Например, на рис. 18.10 область P однозначна в направлении вектора A и неоднозначна в направлении вектора B. Выпуклые области однозначны в любом направлении.

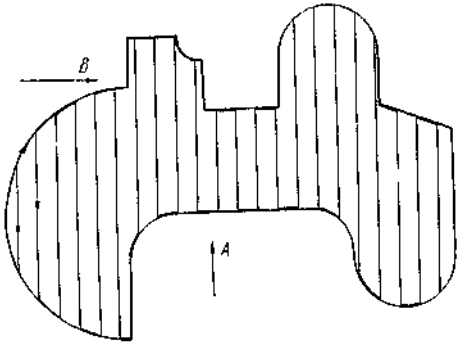


Рис.18.10. Траектория построчной обработки плоской области (вычерчено на ЧГА «ИТЕКАН-2М»)

При автоматическом синтезе траектории построчной обработки плоской области в направлении, в котором она неоднозначна, появляется несколько участков, которые нельзя обработать непрерывным движением фрезы, перемещая ее всегда между границами области. Поэтому возникает многовариантная задача установления траекторий холостых перемещений инструмента.

Таким образом, при синтезе траектории построчной обработки необходимо решать три основных вопроса: 1) разбиение области *P* на однозначные; 2) синтез траектории обработки области, однозначной в направлении обработки; 3) синтез траекторий холостых перемещений инструмента.

В общем случае оптимальная в некоюром смысла траектория обработки может состоять из участков, обрабатываемых разными способами (по эквидистанте, строке и др.).

При проектировании построчной обработки области, однозначной в заданном направлении *A,* начиная с некоторой исходной точки, проводится семейство параллельных прямых, отстоящих друг от друга на расстоянии диаметра фрезы и расположенных по одну сторону от исходной точки. Исходная точка, направление построчной обработки и перпендикулярное ему направление (влево, вправо) могут быть введены с помощью графических средств связи или функциональной клавиатуры дисплея. Затем ищутся точки пересечения этих прямых с контуром. Получаемые таким образом отрезки поочередно ориентируются то в одну, то в другую сторону и соединяются участками контура, заключенными между концами предыдущих отрезков и началами следующих. На рис. 18.10 показана траектория

построчной обработки области, полученная на ЭВМ и вычерченная на ЧГА «ИТЕКАН-2М».

В общем случае задача синтеза управляющей программы является многовариантной и при ее решении в значительной степени используется опыт специалиста. Существует также другой объективный фактор, который существенно затрудняет полное математическое решение этой задачи на ЭВМ. В процессе синтеза траектории обработки принятие тех или иных решений основывается на учете многих геометрических факторов, которые в обычных условиях легко оцениваются зрительно. При автоматическом синтезе траектории обработки выявление этих факторов, как правило, сводится к решению на ЭВМ **позиционных геометрических зада**ч, требующих для своей реализации значительных затрат машинного времени. Решение позиционных геометрических задач возникает при рассмотрении каждого нового варианта обработки поверхности. Поэтому выбор оптимальной траектории обработки поверхности целесообразно осуществлять в человеко-машинном варианте. В этом случае вычислительные и другие геометрические задачи могут быть решены автоматически, а процесс принятия принципиальных решений может быть осуществлен человеком. Естественно, что при такой организации системы важнейшим инструментом, облегчающим взаимодействие человека с ЭВМ, должны явиться графические средства связи.

19. Образование фреймов процессов сборки технических устройств

19.1. Основные конструктивно-технологические свойства технических устройств

Производственная система сборки является сложной иерархической системой, и ее элементы на различных структурных уровнях представляют собой объекты разной сложности — от подсистем, связанных с выполнением основных функций, до первичных элементов системы — технологической оснастки, инструментам, оборудования, рабочих мест и исполнителей. Функционирование производственной системы характеризуется составом и последивательностью технологических операций и получаемыми свойствами объекта производства. По характеру выполняемых функции производственная система разделяется на подсистемы

технологических процессов, оснащения и производственных подразделений.

Технологический процесс сборки включает операции: установки элементов сборочной единицы в требуемое положение относительно друг друга; соединения установленных элементо болтами, заклепками, сварными швами и т. п.; формообразующие, связанные с обеспечением точности и взаимозаменяемости (доводочные операции при установке элементов сборочной единицы обработка разъемов и стыков в разделочных стендах и т. п.) прочие, связанные со специфическим назначением сборочной единицы (операции герметизации, балансировки и т. п.).

Перечисленные технологические операции относятся к группе основных, при выполнении которых изменяются свойства объекта производства. Особую группу составляют операции контроля, назначаемые как для контроля качества изделия, так и для контроля основных технологических операций, оборудования, инструмента, оснастки и т. д.

На содержание автоматизированного образования фрейма процесса сборки влияют свойства производственной системы и конструктивно-технологические свойства изделия структура элементов конструкции и контуров сборочной единицы, пространственная взаимосвязь элементов конструкции, размерные связи и т. д.

Если A — сборный элемент конструкции, то состав деталей, входящих в сборочную единицу, описывается как упорядоченное множество $A = (a_1, a_2, ..., a_n)$. Составы деталей подсборок, входящих в A, записываются как подмножества $A^J_k \subseteq A$; иерархическая структура элементов —подсборок, узлов, деталей — сборочной единицы A описывается в виде графа — дерева $G = (A^J, C)$, где

$A = (A^1_1, A^1_2, ..., A^J_m, a_1, a_2, ..., a_n)$ и

$$C \ni c_{i\,(j)} = \begin{cases} 1, \text{ если } A^J_i \ni A^{J-1}_j; \\ 0 \text{ — в противном случае,} \end{cases} \tag{19.1}$$

или

$$C \ni c_{j\,(i)} = \begin{cases} 1, \text{ если } A^J_i \ni A^{J-1}_j; \\ 0 \text{ — в противном случае.} \end{cases} \tag{19.2}$$

Граф $G = (A^J, C)$ с составом дуг (19.1) характеризует схему членения, а с составом (19.2) —схему сборки изделия A. Например, состав деталей, входящих в узел (рис. 19.1) гитары токарно-винторезного станка, описывается множеством $A = (a_1, a_2, ..., a_n)$; граф $G = (A^J, C)$ одной из возможных схем членения этого узла показан на рис. 19.2, *а* и граф схемы сборки — на рис.19.2, *б*.

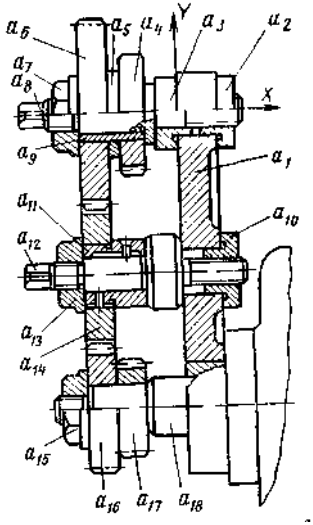


Рис. 19.1. Узел гитары токарно-винторезного станка:
A_1— кронштейн; a_2, a_3, a_9, a_{10}, a_{11} — втулки; a_4 — шестерня привода; a_5 — распорное кольцо; a_6, a_{14}, a_{16}, a_{17} — сменные шестерни; a_7, a_{13}, a_{15} — гайки; a_8, a_{12} — пальцы; a_{18} — вал фартука станка

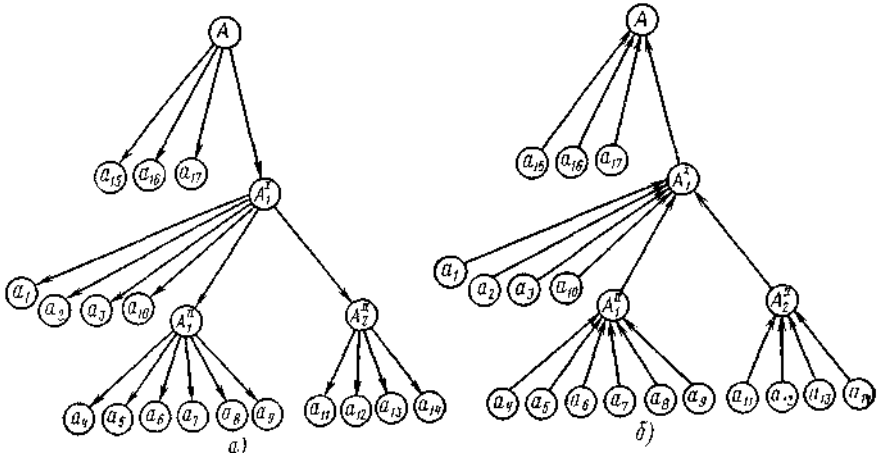


Рис. 19.2. Схемы членения (а) и сборки (б) узла гитары токарно-винторезного станка

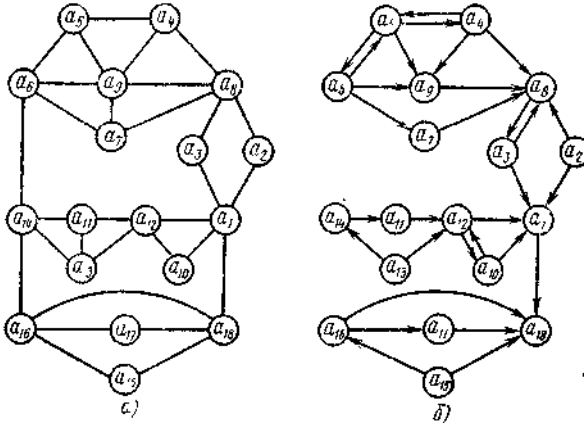
Отношение иерархической подчиненности отражает один из видов бинарных отношений между элементами изделия; кроме того, могут существовать бинарные отношения других видов описываемые булевыми матрицами (15.4)—(15.5) или соответствующими графами. Наиболее важны при сборке такие бинарные отношения:

1) элементы a_i, a_j сопряжены друг с другом (отношение сопряженности описывается матрицей или графом сопряжений — рис. 19.3, *а);*

2) элементы a_i, a_j соединены друг с другом (отношение описывается матрицей или графом соединений);

3) элементы a_i, a_j влияют на возможные перемещения друг друга (отношение описывается матрицей или графом пространственной взаимосвязи);

4) элемент a_i является базовым для a_j (отношение описывается матрицей или графом базирования — рис. 19.3, *б).*



I j

Рис. 19.3. Граф сопряжений *(а)* и граф базирования (б) узла гитары токарно-винторезного станка

На содержание фрейма технологического процесса сборки существенное влияние оказывает не только состав контуров сборочной единицы, но и составы элементов конструкции, свойства которых входят в состав контуров *F (A).* Поэтому контур F_j *(A)* следует рассматривать как теоретический контур, а состав элементов конструкции

$$F_j(A) = (a_{j1}, a_{j2}, \ldots, a_{jr}) \subseteq A,$$

свойства (контуры) которых влияют на существование F_j *(A),* следует рассматривать как конструктивное тело контура $F_j(A)$. Элементы $a_{j_k} \in F(A)$ называются звеньями тела контура; при этом звенья, соответствующие строкам матрицы (15.3), в которых $c_{i(j)}=1$, называются исполнительными звеньями, а звенья, определяемые через элементы $c_{i(j)} = 1$ j-го столбца матрицы (15.2), называются внутренними звеньями тела контура $F_j(A)$.

Применительно к геометрическим контурам F_j *(A)* конструктивное тело F_j *(A)* представляет собой совокупность элементов конструкции, поверхности которых входят в контур F_j *(A),* и элементов конструкции, свойства которых влияют на свойства геометрического контура F_j *(A).* Части элементов конструкции, поверхности которых входят в состав поверхностей F_j, будут исполнительными звеньями, а все остальные — внутренними звеньями F_j *(A).* Например, в конструкции вала с шестерней (см. рис. 15.4, б) исполнительным звеном тела контура F_1 зубчатого зацепления будет шестерня, а внутренними звеньями — все остальные детали (вал, шпонка, накладка и т. д.).

Под пространственной взаимосвязью элементов изделия понимают характер их взаимного расположения, наличие механических связей и характер возможных движений одного относительно другого или какой-либо системы отсчета. Пространственная взаимосвязь определяет возможные варианты последовательности установки элементов изделия при сборке, возможность применения тех или иных способов базирования и компенсации погрешностей, подходы к зонам выполнения операций и т. д. Описание пространственной взаимосвязи осуществляется методами системы ИСТРА на различных уровнях абстрагирования.

Взаимное расположение элементов изделия A задают на сборочном чертеже. Способ задания расположения элемента $a_i \in A$ может быть либо неявным — через установочные размеры и размеры сопряжений элемента a_i с другими элементами A, либо каноническим — заданием линейных и угловых координат начала собственной системы координат элемента a_i в системе координат изделия A.

Описание характера движения элемента изделия разделяют на описание траектории движения и описание состава и взаимосвязи возможных движений в процессе сборки. На количественном уровне описание траектории движения делают методами аналитической геометрии. Весьма удобным методом, хорошо приспособленным для выполнения расчетов на ЭВМ, является описание траектории движения с помощью матриц перемещения $\|D^\alpha\|$. Матрица $\|D^\alpha\|$ описывает движение как векторную величину — возможное

перемещение D^{α}, имеющее модуль d^{α}. На логическом уровне возможное перемещение D^{α} рассматривается как логическая переменная

$$D^{\alpha} = \begin{cases} 1, \text{ если } d^{\alpha} > \Delta^{\alpha}; \\ 0, \text{ если } d^{\alpha} \leqslant \Delta^{\alpha}, \end{cases}$$

где Δ^{α} — величина вектора D^{α}, заданная при решении конкретной задачи.

В декартовой системе координат существует двенадцать различных векторов возможных перемещений, коллинеарных осям координат: шесть поступательных

$$D^{+\bar{x}}, D^{-\bar{x}}, D^{+\bar{y}}, D^{-\bar{y}}, D^{+\bar{z}}, D^{-\bar{z}}$$

и шесть вращательных

$$D^{+\overset{0}{x}}, D^{-\overset{0}{x}}, D^{+\overset{0}{y}}, D^{-\overset{0}{y}}, D^{+\overset{0}{z}}, D^{-\overset{0}{z}}.$$

Общий характер возможных движений a_i относительно a_j описывается уравнением $D_{i(j)}$, включающим состав возможных перемещений и описание их взаимосвязи с помощью логических операций:

$$D_{i(j)} = \begin{cases} D^{\alpha} \wedge D^{\beta} \text{ — если } D^{\alpha} \text{ и } D^{\beta} \text{ могут быть реализованы только одновременно;} \\ D^{\alpha} \vee D^{\beta} \text{ — если они могут быть реализованы либо одновременно, либо поочередно в любой последовательности;} \\ D^{\alpha} \nabla D^{\beta} \text{ — если они могут быть реализованы только поочередно в любой последовательности.} \end{cases}$$

Для логической суммы и логического произведения одинаковых возможных перемещений a_i относительно a_j имеют силу законы идемпотентности

$$\bigvee_{i=1}^{n} D_i^{\alpha} = D_r^{\alpha}, \quad d_r^{\alpha} = \sum_{i=1}^{n} d_i^{\alpha};$$

$$\bigwedge_{i=1}^{n} D_i^{\alpha} = D_r^{\alpha}, \quad d_r^{\alpha} = d_{i\,\min}^{\alpha}.$$

В связанной системе тел возможные перемещения a_i относительно a_j определяются с учетом влияния остальных тел. Совокупность тел будет связанной системой, если граф сопряжений этих тел связен. В такой системе содержание уравнения $D_{i(j)}$ зависит от структуры графа сопряжений и определяется по формуле

$$D_{i\,(j)} = \bigwedge_{J=1}^{N} (D_{i\,(j)})_J = \bigwedge_{J=1}^{N} \left(\bigvee_{i}^{i+1=j} D^{\alpha}_{i\,(i+1)} \right)_J,$$

где N — общее количество простых цепей между a_i и a_j в графе сопряжений.

Если граф сопряжений — простая цепь ($J = 1$), то в уравнении $D_{i(j)}$

$$D^{\alpha}_{i\,(j)} = \bigvee_{i}^{i+1=j} D^{\alpha}_{i\,(j)}, \quad d^{\alpha}_{i\,(j)} = \sum_{i}^{i+1=j} d^{\alpha}_{i\,(i+1)}; \quad (19.3)$$

если граф сопряжений имеет более сложную структуру ($J > 1$), то

$$D^{\alpha}_{i\,(j)} = \bigwedge_{J=1}^{N} (D^{\alpha}_{i\,(j)})_J, \quad d^{\alpha}_{i\,(j)} = (d^{\alpha}_{i\,(j)})_{J_{\min}},$$

где $(D^{\alpha}_{i\,(j)})_J$ вычисляется по формуле (19.3), как $D^{\alpha}_{i(j)}$.

В процессе сборки каждый элемент a_i изделия переходит из свободного в связанное состояние. Этот переход осуществляется путем реализации некоторых возможных перемещений a_i относительно установленных ранее элементов изделия. Установка a_i при сборке возможна, если установленные ранее элементы не препятствуют ей. Состав элементов, препятствующих установке, описывается уравнением

$$\overline{\overline{D}}(a_i) =$$

$$= \begin{cases} a_j \vee \cdots \vee a_k & \text{— если } D_{i\,(j)} = \cdots = D_{i\,(k)} = 0, \text{ т. е. любой из элементов } a_j, \ldots, a_k \text{ препятствует установке } a_i; \\ a_j \wedge \cdots \wedge a_k & \text{— если } D_{i\,(j)} \wedge \cdots \wedge D_{i\,(k)} = 0, \text{ т. е. элементы } a_j, \ldots, a_k \text{ препятствуют установке } a_i \text{ только все вместе взятые;} \\ 0 & \text{— если установке } a_i \text{ не препятствуют никакие элементы.} \end{cases}$$

В состав уравнений $D_{i\,(j)}, \ldots, D_{i\,(k)}$, влияющих на содержание уравнения $\overline{\overline{D}}(a_i)$, включаются освобождающиеся возможные перемещения, реализация которых приводит к утрате связи с сопрягаемыми элементами $a_j, \ldots, a_k$. Состав уравнений $\overline{\overline{D}}(a_i)$ можно определить и непосредственно по чертежу изделия, анализируя характер возможных движений элементов. Например, для деталей a_4, a_5, a_6, a_7, a_9 (см. рис. 19.1) уравнения $\overline{\overline{D}}(a_i)$ (имеют вид

$$\overline{\overline{D}}(a_4) = (a_5 \vee a_6 \vee a_7) \wedge a_8;\ \overline{\overline{D}}(a_5) = (a_6 \vee a_7) \wedge [a_8 \vee (a_4 \wedge a_9)];$$

$$\overline{\overline{D}}(a_6) = a_7 \wedge [a_8 \vee (a_9 \wedge (a_4 \vee a_5))];\ \overline{\overline{D}}(a_7) = 0;$$

$$\overline{\overline{D}}(a_9) = a_7 \wedge a_8.$$

Содержание уравнений $D_{i(j)}$ и $\overline{\overline{D}}(a_i)$ определяется характером расположения элементов a_i, a_j, ...,a_k в пространстве. Поскольку эти элементы — материальные объекты, в одном и том же месте пространства могут находиться точки лишь какого-либо одного объекта, т. е. множества точек этих объектов не пересекаются. Это свойство записывается в виде условия непересечения геометрических фигур на плоскости и в пространстве.

Обозначим форму объекта a_i контуром F^p (a_i), а положение относительно внешней системы координат — контуром $F^U(a_i)$. Для количественного описания контуров формы и положения a_i обычно используется аппарат аналитической геометрии. Однако при технологическом проектировании сборки использование такого аппарата часто нерационально из-за большого объема данных, необходимых для описания элементов изделия, оборудования, инструмента и оснастки. В этом случае целесообразно пользоваться упрощенными, но более экономными методами, например, прямоугольной моделью формы и расположения объектов. В этой модели объект a_i представляется как прямоугольный параллелепипед, грани которого параллельны плоскостям координат (рис. 19.4).

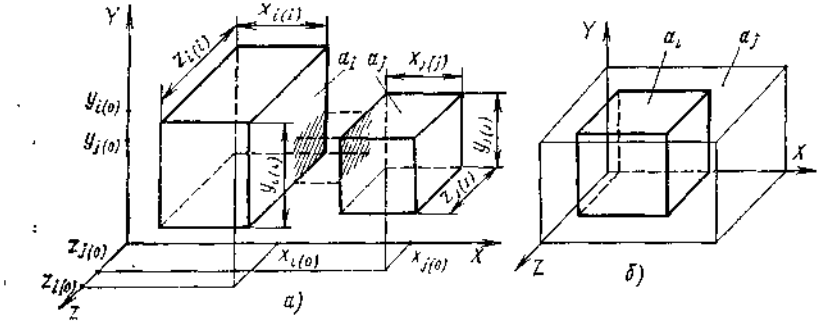


Рис. 19.4. Прямоугольные модели пространственной взаимосвязи объектов

Конутр формы F^P_i (a_i) при этом определяется тремя параметрами: $x_{i\,(i)}$, $y_{i\,(i)}$, $z_{i\,(i)}$, а контур положения $F^U(a_i)$ —тремя линейными координатами $x_{i(0)}$, $y_{i(0)}$, $z_{i(0)}$.

Если точность описания с помощью одного параллелепипеда недостаточна, то он описывается набором нескольких параллелепипедов, неподвижно связанных друг с другом. Условие непересечекия параллелепипедов a_i и a_j описывается выражением

$$\left.\begin{array}{c} a_i \cap a_j \neq \varnothing, \text{ если} \\ \exists\alpha\,(\alpha_{i\,(0)} > \alpha_{j\,(0)} + \alpha_{j\,(j)}) \nabla (\alpha_{j\,(0)} > \alpha_{i\,(0)} + \alpha_{i\,(i)}) = 1), \\ \alpha = x, y, z. \end{array}\right\} \quad (19.4)$$

Условие пересечения проекций объектов a_i и a_j на ось O_α ($\alpha = x, y, z$) имеет вид

$$(\alpha_{i\,(0)} < \alpha_{j\,(0)} < \alpha_{i\,(0)} + \alpha_{i\,(i)}) \nabla (\alpha_{j\,(0)} < \alpha_{i\,(0)} < \alpha_{j\,(0)} + \alpha_{j\,(j)}) = 1, \quad (19.5)$$

а условие пересечения проекций на плоскость $O_{\alpha\beta}$ ($\alpha, \beta = x, y, z$) имеет вид

$$\left.\begin{array}{c} (\alpha_{i\,(0)} < \alpha_{j\,(0)} < \alpha_{i\,(0)} + \alpha_{i\,(i)}) \nabla (\alpha_{j\,(0)} < \alpha_{i\,(0)} < \alpha_{j\,(0)} + \alpha_{j\,(j)}) = 1 \\ (\beta_{i\,(0)} < \beta_{j\,(0)} < \beta_{i\,(0)} + \beta_{i\,(i)}) \nabla (\beta_{j\,(0)} < \beta_{i\,(0)} < \beta_{j\,(0)} + \beta_{j\,(j)}) = 1. \end{array}\right\} \quad (19.6)$$

Если, например, выполняется условие

$$x_{j\,(0)} > x_{i\,(0)} + x_{i\,(i)} \quad (19.7)$$

и условие (19.6) относительно плоскости *OYZ* (рис. 19.4, *а),* то в направлении $+X$ существует ограниченное возможное перемещение

$$D_{i\,(j)}^{+\bar{x}} = 1, \; d_{i\,(j)}^{+\bar{x}} = x_{j\,(0)} - (x_{i\,(0)} + x_{i\,(i)}); \quad (19.8)$$

в случае

$$x_{j\,(0)} = x_{i\,(0)} + x_{i\,(i)} \quad (19.9)$$

возможное перемещение $D_{i\,(j)}^{+\bar{x}} = 0$, и объекты a_i и a_j сопряжены друг с другом в плоскости, параллельной *OYZ*, а в графе сопряжений $G = (A, C)$ существуют дуги $c_{i(j)} = c_{j(i)} = 1$. Аналогично вычисляются остальные возможные перемещения.

Использование отношений вида (19.6)—(19.9) позволяет описывать и условия, при которых a_i может разместиться в определенной зоне. Например, размещение a_i внутри a_j (рис. 19.4, б) возможно при условии

$$((x_{i\,(0)} \geqslant x_{j\,(0)}) \wedge (x_{i\,(i)} \leqslant x_{j\,(j)})) \wedge ((y_{i\,(0)} \geqslant y_{j\,(0)}) \wedge$$
$$\wedge (y_{i\,(i)} \leqslant y_{j\,(j)})) \wedge ((z_{i\,(0)} \geqslant z_{j\,(0)}) \wedge (z_{i\,(i)} \leqslant z_{j\,(j)})) = 1.$$

Наличие сопряжения a_i с a_j, а также действующих на них механических сил является причиной существования механических связей между этими объектами. Наличие этих связей описывается графом сопряжений G = *(A,* C). При сборке между элементами изделия существуют в основном голономные неудерживающие механические связи, поэтому характер связей описывают с помощью единичных баз — векторных величин, параллельных осям базовой системы координат. В декартовой системе координат возможны двенадцать баз: шесть поступательных $B^{+\bar{x}}, B^{-\bar{x}}, B^{+\bar{y}}, B^{-\bar{y}}, B^{+\bar{z}}, B^{-\bar{z}}$, характеризующих связи, препятствующие поступательным движениям параллельно осям координат, и шесть вращательных $B^{+\overset{0}{x}}, B^{-\overset{0}{x}}, B^{+\overset{0}{y}}, B^{-\overset{0}{y}}, B^{+\overset{0}{z}}, B^{-\overset{0}{z}}$, характеризующих связи, которые препятствуют поворотам вокруг осей, паралельных осям координат.

Состав единичных баз определяется на основе анализа сил, действующих на базируемое тело, в виде системы скользящих векторов. Представляя единичную базу как булеву переменную

$$B^{\alpha} = \begin{cases} 1, \text{ если связь существует;} \\ 0, \text{ если связь не существует,} \end{cases} \qquad (19.10)$$

состав механических связей элемента a_i с элементом a_j можно описать булевым вектором базирования $B_{i(j)}$, компонентами которого будут истинные значения булевых переменных (19.10), соответствующих $B^{+\bar{x}}, B^{-\bar{x}}, \ldots, B^{+\overset{0}{z}}, B^{-\overset{0}{z}}$. Если элемент a_i связан с несколькими элементами a_j, где j = 1, 2, ..., m, то вектор базирования $B_{i\{m\}}$ этого элемента относительно всех вместе взятых элементов a_j вычисляется с помощью логического сложения булевых векторов $B_{i(j)}$, где j= 1, 2, ..., m:

$$B_{i\{m\}} = B_{i(1)} \vee B_{i(2)} \vee \cdots \vee B_{i(m)} = \bigvee_{j=1}^{m} B_{i(j)} . \qquad (19.11)$$

При вычислении $B_{i\{m\}}$ число элементов a_j равно числу дуг $c_{i(j)} = 1$ в *i-й* строке матрицы смежностей графа сопряжений элементов изделия.

Например, для распорного кольца a_5 (см. рис. 19.1) по формуле (19.11) получим

$$
\begin{array}{lcccccccccccc}
 & B^{+\bar{x}} & B^{-\bar{x}} & B^{+\bar{y}} & B^{-\bar{y}} & B^{+\bar{z}} & B^{-\bar{z}} & B^{+\mathring{x}} & B^{-\mathring{x}} & B^{+\mathring{y}} & B^{-\mathring{y}} & B^{+\mathring{z}} & B^{-\mathring{z}} \\
B_{5\,(4)} = (1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0) \\
B_{5\,(6)} = (0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0) \\
B_{5\,(9)} = (0, & 0, & 1, & 1, & 1, & 1, & 0, & 0, & 1, & 1, & 1, & 1) \\
\hline
B_{5\,\{m\}} = (1, & 1, & 1, & 1, & 1, & 1, & 0, & 0, & 1, & 1, & 1, & 1)
\end{array}
$$

Одной из важнейших задач при образованиифрейма технологического процесса сборки является определение требуемых составов сборочных баз, обеспечивающих определенность базирования устанавливаемых элементов изделия. Для решения этой задачи значение $B^{\alpha} = 1$ в формуле (19.10) принимается только в том случае, когда существует механическая связь и обеспечивается требуемая точность базирования. Возможные составы сборочных баз элемента a_i описывают выражением

$$
B(a_i) = \begin{cases}
a_j \wedge \cdots \wedge a_m \text{ — если } B_{i\,(0)} = \bigvee B_{i\,(k)},\ B_{i\,(k)} \neq B_{i\,(0)}; \\
a_j \vee \cdots \vee a_m \text{ — если } B_{i\,(0)} = B_{i\,(j)} = \cdots = B_{i\,(m)}; \\
\qquad a_i \text{ — если элемент } a_i \text{ не нуждается в сборочной базе;} \\
\qquad 0 \text{ — если определенность базирования } a_i \text{ не обеспечена.}
\end{cases}
$$

Здесь $B_{i(0)}$ — вектор базирования, соответствующий случаю, когда обеспечена определенность базирования. Например, для деталей $a_4, a_5, \ldots, a_9$ (см. рис. 19.1) уравнения составов сборочных баз будут

$$B(a_4) = a_9 \wedge (a_5 \vee a_8); \quad B(a_5) = a_9 \wedge (a_4 \vee a_6);$$

$$B(a_6) = a_9 \wedge (a_5 \vee a_7); \quad B(a_7) = B(a_8) = B(a_9) = a_8.$$

В конструкции изделия сопряжение любых элементов осуществляется по контурам, имеющим строго определенную форму. Поэтому каждое сопряжение обеспечивает вполне определенный характер возможных перемещений сопрягаемых элементов относительно друг друга. По составу и виду этих перемещений конструкции разделяются на классы подвижности. Класс подвижности определяется степенью базирования, т. е. числом единичных баз, ограничивающих возможные перемещения сопрягаемых элементов. По характеру взаимосвязи перемещений, входящих в уравнение возможных перемещений, классы подвижности подразделяются на дизъюнктивные и конъюнктивные.

В уравнении возможных перемещений дизъюнктивного класса подвижности имеются только возможные перемещения, параллельные

осям координат, связанные неразделительной или разделительной дизъюнкцией. В табл. 19.1 приведены все возможные дизъюнктивные классы подвижности различных родов, которые могут существовать при сопряжении элемента конструкции с другим элементом или группой неподвижных относительно друг друга элементов.

Таблица 19. 1

Дизъюнктивные классы подвижности различных родов

B_j \ $П_i$	$П_0$	$П_I$	$П_{II}$ а	$П_{II}$ б	$П_{III}$ а			$П_{III}$ б	$П_{IV}$ а			$П_{IV}$ б			$П_V$				$П_{VI}$
B_0	●	●	●	●	⊂⊃			●	⊂⊃			⊂⊃			⊂⊃				●
B_I			●		●	●	●		⊂⊃		●	●	⊂⊃		⊂⊃		●	●	●
B_{II} а			●		●	⊂⊃			⊂⊃		●	⊂⊃			⊂⊃		⊂⊃		●
B_{II} б			●		⊂⊃		●		●	⊂⊃		●	●	●	●	⊂⊃		●	●
B_{III} а			●		●	●	●		●	●	●	●	⊂⊃		●	●	●	●	●
B_{III} б									⊂⊃						⊂⊃			●	●
B_{IV} а			●		⊂⊃				●	⊂⊃		⊂⊃			●	⊂⊃			●
B_{IV} б									⊂⊃		●				●	●	⊂⊃		●
B_V									⊂⊃		●				⊂⊃		●	●	●
B_{VI}									⊂⊃						⊂⊃				●

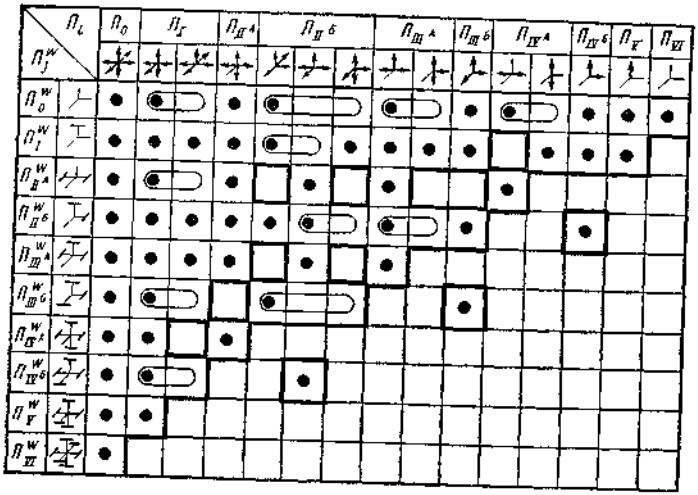
Например, дизъюнктивный класс подвижности шестерни относительно втулки a_9 без учета других деталей (см. рис. 19.1) равен $П_{IVA} \vee B_{IVA}$, а с учетом всех деталей узла — $П_{VI} \vee B_{IVA}$.

Если уравнение возможных перемещений сопряженных элементов не может быть записано в виде совокупности перемещений, связанных только дизъюнкцией, а обязательно содержит логические произведения возможных перемещений, то такое сопряжение будет относиться к конъюнктивному классу подвижности. Например, класс подвижности гайки a_7 в узле, показанном на рис. 19.1, равен $П_V \wedge B_V$.

При решении некоторых задач проектирования сборки, например при оценке характера доступа в зону выполнения операции, при выборе оборудования, инструмента и т. п , необходимо учитывать ограничения на поступательные возможные перемещения. Классы подвижности с учетом ограничений по поступательным возможным перемещениям приведены в табл. 19.2.

Таблица 19.2

Классы подвижности с учетом ограничений



Автоматизация образования фреймов технологических процессов сборки с применением ЭВМ осуществляется в рамках подсистемы сборочных работ АСТПП. Основными элементами математического обеспечения этой подсистемы являются математическая модель производственной системы сборки; математическая модель собираемого изделия, алгоритмы проектирования технологических процессов и оснащения сборочных работ. Создание этих элементов предшествует собственно проектированию технологических процессов сборки с применением ЭВМ. Процесс проектирования заключается в моделировании на ЭВМ поведения модели производственной системы сборки с учетом информации о конструктивно-технологических свойствах изделия, вводимой в ЭВМ в виде математической модели собираемого изделия.

Производственная система предприятия представляется состоящей из производственных систем низшего уровня, связанных с определенными видами работ. Одной из таких систем низшего уровня и является производственная система сборки. В свою очередь, сама производственная система сборки имеет иерархическую структуру, включающую элементы различной сложности. Основными подсистемами производственной системы сборки являются подсистемы технологических процессов сборки, производственных подразделений сборки и оснащения сборки.

Производственные и технологические возможности и свойства производственной системы сборки описываются комплексом взаимосвязанных табличных, сетевых и перестановочных математических

моделей, используемых при автоматизации проектирования технологических процессов и оснащения сборки. Эти модели разрабатываются с различной степенью полноты и подробности описания. Полнота описания определяется тем, охватывает ли модель все возможные элементы и процессы производственной системы (например, на уровне отрасли промышленности), или только часть этих элементов и процессов (на уровне системы производства изделий одного класса, на уровне конкретного предприятия, производственного участка и т. п.). Подробность описания определяется тем, какие технологические процессы будут моделироваться в данной производственной системе — укрупненные, маршрутные, на уровне операций и т. д.

Исходными данными для разработки математических моделей производственной системы сборки являются технические описания оборудования, инструмента, оснастки, РТМ, производственные инструкции и т. п. Модель производственной системы на конкретном предприятии разрабатывается один раз и в дальнейшем только корректируется в соответствии с необходимыми изменениями, связанными с изменением состава и свойств элементов производственной системы.

Образование фреймов технологического процесса сборки с применением ЭВМ разделяется на два этапа: 1) разработка фрейма принципиальной схемы сборки; 2) разработка фреймов операционного технологического процесса сборки. На первом этапе производятся: выбор схемы технологического членения изделия и графа-схемы сборки; выбор схемы базирования и конструктивной схемы сборочной оснастки; выбор последовательности установки элементов сборочной единицы и других основных операций сборки. На втором этапе производятся: выбор состава и последовательности выполнения операций соединения, контроля и т. д. при заданной последовательности установки элементов сборочной единицы; выбор состава инструмента, оборудования и оснастки; определение состава и квалификации исполнителей; расчет технико-экономических показателей и выбор оптимального варианта технологического процесса сборки; формирование карт технологического процесса сборки.

При решении всех задач широко используются фреймы расчетных моделей собираемого изделия. Расчетная модель является аналогом моделируемого объекта, включающим минимум информации, необходимой и достаточной для решения конкретной задачи с учетом требуемой точность решения. Количество элементов расчетной модели, как правило, намного меньше, чем у моделируемого объекта,

так как при разработке расчетной модели рассматриваются только свойства, существенно влияющие на решение данной задачи.

Фрейм расчетной модели сборочной единицы A разрабатывается на основе анализа конструктивно-технологических свойств элементов конструкции, таких, как состав функциональных и технологических контуров, характер пространственной взаимосвязи, базовые свойства, принадлежность к тем или иным структурным классам и т. д. Для образования фрейма расчетной модели A строится булева матрица (15.3) описания свойства элементов конструкции $a_i \in A$. Построение расчетной модели A по матрице (15.3) сводится к минимизации числа векторов-строек матрицы путем замены строк, обладающих одинаковыми значениями компонент $c_{i(j)}$, одной вектор-строкой. После проведения минимизации каждая оставшаяся строка будет соответствовать элементу расчетной модели A.

Для сложных изделий — двигателей, сложных редукторов и т. п. — разрабатывается несколько расчетных моделей, каждая из которых служит для решения конкретной задачи.

19.2. Образование фрейма принципиальной схемы сборки

Любая принципиальная схема сборки возможна только при условии, что при такой схеме возможен хотя бы один технологический процесс сборки, обеспечивающий требуемые качества изделия. Поэтому при образовании фрейма принципиальной схемы сборки необходимо проектировать и возможные варианты укрупненных технологических процессов, отражающие состав и последовательность выполнения основных операций и основные технико-зкономические показатели сборки. Если состав всех основных технологических операций определяется составом функциональных и технологических контуров изделия, подлежащих реализации, то последовательность выполнения всех операций сборки зависит главным образом от последовательности установки элементов изделия. Так, операции соединения могут выполняться только после установки соединяемых элементов, операции контроля — после выполнения соответствующих операций установки и соединения и т. д. Поэтому для получения оптимального технологического процесса сборки необходимо прежде всего анализировать различные варианты установки элементов изделия.

Процесс установки элементов сборочной единицы A представляется как упорядоченная последовательность T_m из элементов $a_i \in A$.

Основными факторами, влияющими на последовательность установки, являются условия базирования и доступа к месту установки при сборке. Условие базирования при установке a_i выполняется, если среди установленных ранее элементов есть такие, которые образуют хотя бы один из возможных составов сборочной базы вида $B_p(a_i) = a_1 \wedge \ldots \wedge a_k$. Условие доступа выполняется, если среди установленных ранее элементов нет таких, которые образуют состав элементов, препятствующих установке a_i, вида $\overline{\overline{D}}_q(a_i) = a_1 \wedge \ldots \wedge a_k$. Очевидно, эти условия являются конкретной формой проявления условий вида (15.64)—(15.65). Формально эти условия можно представить так: установка всех элементов изделия $A=\{a_1, a_2, \ldots, a_n\}$ в последовательности T_m возможна, если выполняется условие

$$\forall a_i \in A\,[\exists B_p(a_i)\,(B_p(a_i) \subseteq A_i^0)] \quad (19.12)$$

и условие

$$\forall a_i \in A\,[\forall \overline{\overline{D}}_q(a_i)\,(\overline{\overline{D}}_q(a_i) \not\subset A_i^0)], \quad (19.13)$$

где A^0_i — совокупность элементов изделия, установленных до установки a_i.

Последовательность установки T_m возможна только в том случае, когда условия (19.12) и (19.13) выполняются одновременно, в противном случае последовательность T_m не может быть реализована. Например, из приведенных выше уравнений $B(a_i)$ составов сборочных баз и уравнений $\overline{\overline{D}}(a_i)$ элементов, препятствующих установке a_i, для деталей $a_4, a_5, \ldots, a_9$ (см. рис. 19.1) следует, что последовательности установки

$$T'_k = a_3, a_7, a_9, a_4, a_5, a_6;$$

$$T''_k = a_8, a_9, a_6, a_5, a_4, a_7$$

возможны по условию базирования (19.12), однако не удовлетворяют условию доступа (19.13) и поэтому не могут быть реализованы. С другой стороны, последовательности

$$T'_n = a_8, a_4, a_9, \overline{a_3}, a_6, \overline{a_7};$$

$$T''_n = a_8, a_4, a_5, a_6, \overline{a_9}, a_7$$

возможны по условию доступа (19.13), но не удовлетворяют условию базирования (19.12) и также не могут быть реализованы. Единственным вариантом установки, одновременно удовлетворяющим обоим условиям, является последовательность $T_m = a_8, a_9, a_4, a_5, a_6, a_7$, так как здесь условия базирования и доступа выполняются для всех деталей на каждом этапе установки. Таким образом, для проектирования процесса установки может быть использована перестановочная модель класса $S_3(T)$.

Матрица контуров (15.3) этой модели включает в себя главным образом геометрические контуры $F_i \in F(A)$, каждый из которых состоит из двух специфических частей: контура формы F^P_i, характеризующего форму соответствующих поверхностей, и контура положения F^U_i, характеризующего положение этих поверхностей в системе координат сборочной единицы. Взаимосвязь этих контуров описывается конъюнкцией

$$\bar{F}_i = F^P_i \wedge F^U_i.$$

Очевидно, если $a_j \in A$, то

$$\bar{F}^P_i(A) = \bigwedge_{j}^{r} (F^P_i(a_j) \wedge F^U_i(a_j)).$$

Здесь $F^P_i(A)$ — контур формы $F_i(A)$ сборочной единицы; $F^P_i(a_j)$ — контур формы контура $F_i(a_j)$ элемента $a \in A$; $F_i^U(a_j)$ — контур положения $F_i(a_j)$ в системе координат сборочной единицы; r — общее количество замыкающих звеньев тела $F_i(A)$ контура $F_i(A)$.

Условия базирования и доступа влияют не только на последовательность установки элементов изделия, но и на возможность выполнения всех других операций сборки, связанных с применением какого-либо инструмента или оснастки, если между ними и элементами изделия в процессе выполнения операции должна существовать механическая связь. Действительно, инструмент (сверло, зенкер, электрод, фреза и г. п.) должен быть определенным образом ориентирован относительно изделия в зоне обработки; это условие аналогично условию базирования при установке элементов изделия. Очевидно, обработка возможна только в том случае, если существует доступ рабочих органов оборудования с инструментом в зону обработки; это условие аналогично условию доступа к месту установки элементов изделия. Указанные соображения справедливы и для элементов оснастки. Таким образом, пространственная взаимосвязь элементов изделия, оснастки, инструмента и оборудования в процессе сборки является одним из главных факторов, влияющих на содержание и последовательность выполнения сборочных операций.

Состав и последовательность выполнения операций соединения зависят в основном от состава контуров соединений. Контуры соединений элементов конструкции изделий имеют весьма сложную структуру, зависящую от вида соединения (табл. 19.3).

Таблица 19.3

Структура контуров соединения элементов конструкции

Контуры вида соединений	Контуры пакета						Контуры соединительных элементов																				
	Состав пакета			Сопрягаемые поверхности			Контур места соединительного элемента						Контуры собственно соединительных элементов														
	Деталь	Деталь	●●●	Поверхность	Поверхность	●●●	Соединяемая поверхность	Отверстие в пакете	Гнездо в пакете	Кромка под сварку	●●●		Тело болта	Резьба болта	Тело гайки	Резьба гайки	Тело шайбы	Тело винта	Резьба винта	Тело заклепки	Замыкающая головка	Сварная точка	Сварной шов	Припой	Клеевая пленка	●●●
F_1	●	●		●	●			●	●				●	●	●	●	●									
F_2	●	●		●	●			●	●									●	●							
F_3	●	●		●	●			●	●											●	●					
F_4	●	●		●	●		●															●				
F_5	●	●		●	●		●																●			
F_6	●	●		●	●		●			●													●			
F_7	●	●		●	●		●																	●		
F_8	●	●		●	●		●																		●	
F_9	●	●		●	●		●	●	●				●	●	●	●	●	●	●						●	
F_{10}	●	●		●	●		●	●	●											●	●				●	
F_{11}	●	●		●	●		●															●	●		●	

Контуры вида соединений (индексы при *F)* 1 — болтовое, 2 — винтовое, 3 — заклепочное, 4 — сварное точечное, 5 — сварное роликовой сваркой, 6 — сварное дуговой сваркой, 7 — паяное, 8 — клеевое, 9 — клее-резьбовое, 10 — клее-заклепочное, 11 — клее-сварное.

Основными частями всякого контура соединения являются:

контур пакета соединяемых элементов, характеризующий свойства элементов конструкции в месте соединения: состав пакета, толщину пакета, форму сопрягаемых поверхностей, марку материала и т. п.;

контур соединительного шва, характеризующий форму шва, количество рядов соединительных элементов, отверстия и гнезда под болты, винты или заклепки, кромки под сварку, поверхности для нанесения клея или припоя и т. п.;

контур соединительного элемента, характеризующий свойства болта, винта, заклепки, сварной точки и т. п.

Структура связей соединений в конструкции изделия описывается графом сопряжений соединительных элементов с другими элементами конструкции. Контуры соединений описываются аналогично другим элементам конструкции изделия.

Состав операций выполнения соединения определяется видом соединения и отличается большим разнообразием и разнородностью контуров, реализуемых в процессе производства. Технологический процесс соединения вполне четко разделяется на группы упорядоченных операций, для математического моделирования которых используются табличные и сетевые модели операций, объединяемые в многоуровневую модель операций соединения. На

верхнем уровне моделируются состав и взаимосвязь основных операций при различных видах соединений (рис. 19.5), а на последующих уровнях моделируется более подробный состав операций в зависимости от особенностей и параметров контуров соединений (рис. 19.6).

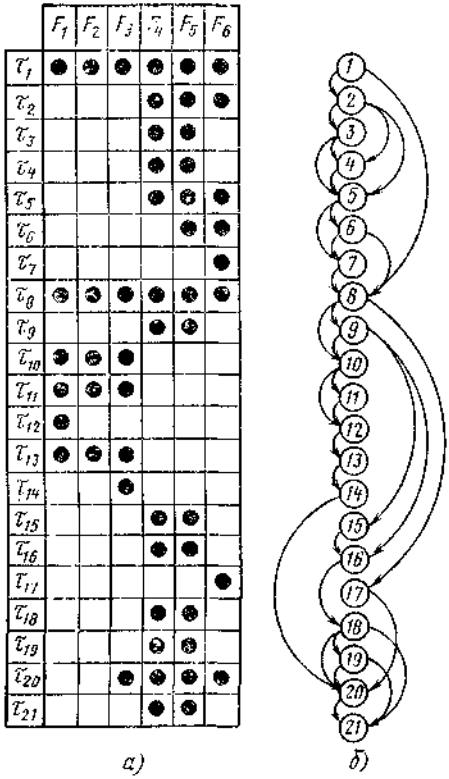


Рис. 19.5. Обобщенная сетевая модель операторов выполнения процессов соединения:

а - конъюнктивная матрица контуров; *б* — граф взаимосвязи операторов; контуры F_i : F_1 — болтовое соединение 7 —9-го квалнтета; F_2 — болтовое соединение 11 —13-го квалитета; F_3 — заклепочное соединение; F_4 — сварное соединение для сталей; F_5 — сварное соединение для алюминиевых сплавов; F_6 — клеевое соединение; операторы τ_i: τ_1 — подготовка деталей под сборку; τ_2 — предварительная установка деталей; τ_3 — подготовка кромок; τ_4 — механическая очистка поверхностей; τ_5 — химическая очистка поверхностей; τ_6 — нанесение покрытия; τ_7 — нанесение клея; τ_8 —

окончательная установка деталей; τ_9 — прихватка; τ_{10} — сверление отверстий; τ_{11} — образование гнезда под потайную головку заклепки; τ_{12} — разделка отверстий; τ_{13} — установка болтов, винтов и заклепок; τ_{14} - расклепывание заклепок; τ_{15} — правка после прихватки; τ_{16} — образование сварного шва; τ_{17} — образование клеевого шва; τ_{18} — правка после сварки; τ_{19} — термообработка сварного узла; τ_{20} — исправление дефектов соединения; τ_{21} — нанесение защитных покрытий

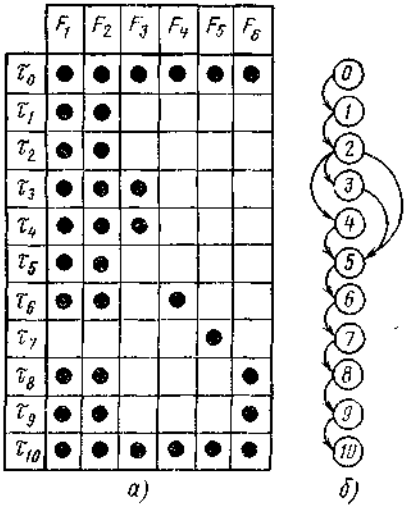


Рис. 19.6. Сетевая модель операторов выполнения болтовых соединений:

а — матрица контуров; б — граф взаимосвязи операторов; контуры. F_1 — отверстие $d \leq 8$ мм; F_2 — отверстие $d \leq 16$ мм; F_3, — отверстие седьмого квалитета; F_4 — контур положения болта; F_5 -- контур положения шайбы; F_6 - контур положения гайки; операторы: τ_0 — исходный оператор; τ_1 — сверление; τ_2 — рассверливание; τ_3 — развертывание; τ_4 — протягивание; τ_5 — снятие фаски; τ_6 — установка болта; τ_7 - установки шайбы; τ_8 — установка гайки; τ_9, — кернение; τ_{10} — конечный оператор.

Графы взаимосвязи операций моделей низшего уровня являются результатом развертки графов взаимосвязи операций моделей высшего уровня.

Для моделирования состава и взаимосвязи элементов материальной системы производства в процессах соединения исноль-зуются табличные и сетевые модели. Сетевые модели применяются в тех случаях, когда операции соединения реализуют одинаковые контуры, но различаются по составу применяемого оборудования, инструмента и оснастки (рис. 19.7).

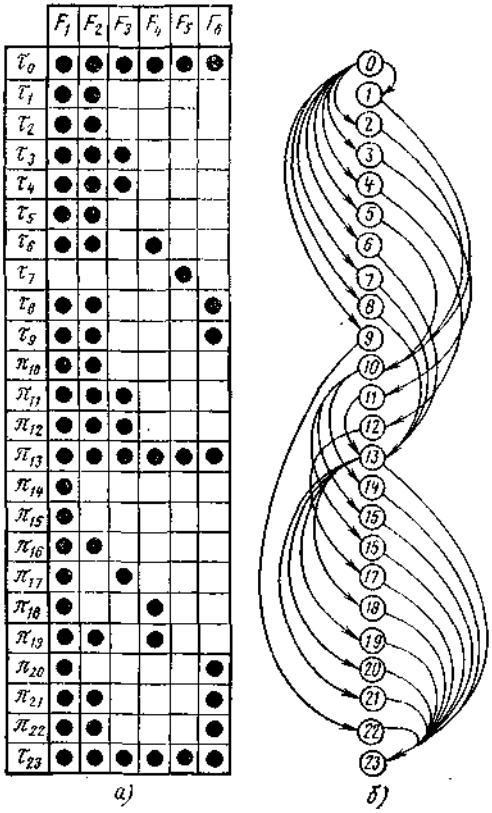


Рис. 19.7. Сетевая модель производственной системы выполнения болтовых соединений:

а — матрица контуров; *б* — граф смежности операторов и элементов производственной системы; контуры; F_1 — отверстие $d \leq 8$ мм; F_2 — отверстие $d \leq 16$ мм; F_3 — отверстие седьмого квалитета; F_4 — контур положения болта; F_5 — контур положения шайбы; F_6 — контур положения гайки; операторы; τ_0 — исходный оператор; τ_1 — сверление;

τ_2 — рассверливание; τ_3 — развертывание; τ_4 — протягивание; τ_5 — снятие фаски; τ_6— установка болта; τ_7 — установка шайбы; τ_8 — установка гайки; τ_9 — кернение; τ_{23} — конечный оператор; элементы производственной системы; π_{10} — сверло; π_{11} — развертка; π_{12} — протяжка; π_{13} — слесарный инструмент; π_{14} — пневмодрель СМ21-25; π_{15} и π_{16} — сверлильно-зенковальные устройства СЗВУ-А1 и СЗВУ-А2; π_{17} —- протяжное устройство ПУ-8; π_{18} и π_{19} — пневмоинструменты ПИВ-10 и ПИВ-16 для вставки болтов; π_{20} и π_{21} — пневмо-гайковерты ПВ21-180 и РПГ41-16, π_{22} — пневмокернер КП-09М

Сетевая модель такого вида является моделью производственной системы S_5 *(P),* описывающей состав и взаимосвязь технологических операторов τ_k и соответствующего им оборудования, инструмента и оснастки π_k. Такая модель используется только для выбора элементов технологической системы, а последовательность реализации операций определяется по моделям типа показанных на рис. 19.5 и 19.6.

Условия доступа и базирования инструмента и элементов оборудования в зоне соединений описываются соотношениями вида (19.12) и (19.13). Уравнения $\overline{\overline{D}}\ (a_i)$ определяются с учетом взаимного расположения и соотношения величин параметров элементов конструкции изделия, оборудования и инструмента. Состав этих параметров зависит от класса подвижности элементов в рабочей зоне с учетом ограничений по поступательным возможным перемещениям (см. табл. 19.2). Для вычисления условий доступа с применением ЭВМ целесообразно использовать прямоугольную модель формы и расположения элементов конструкции изделия, оборудования и инструмента.

Между отдельными этапами технологического процесса соединения и другими операциями сборки существуют определенные отношения порядка. Каждый из этих этапов может состоять из большого числа операций, включая вспомогательные и контрольные операции. Как правило, взаимосвязь операций сборки по возможной последовательности их выполнения даже при укрупненном описании весьма сложна. Например, на рис. 19.8 показан редуктор, а на рис. 19.9 — граф взаимосвязи операторов при сборке этого редуктора.

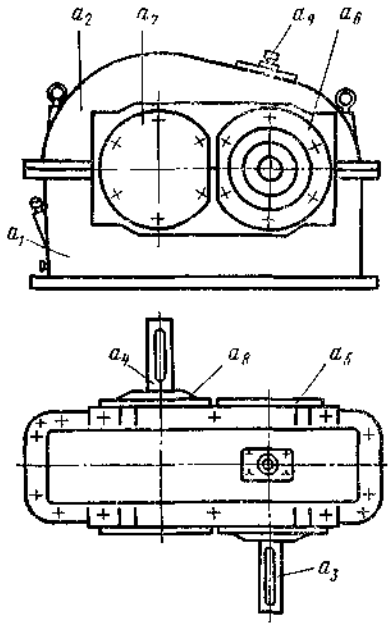


Рис. 19.8. Редуктор цилиндрический: a_1 — корпус редуктора; a_2 — крышка редуктора; a_3 — вал входной; a_4 — вал выходной; a_5, a_7 — крышки глухие; a_8 - крышки сквозные; a_9 - отдушина

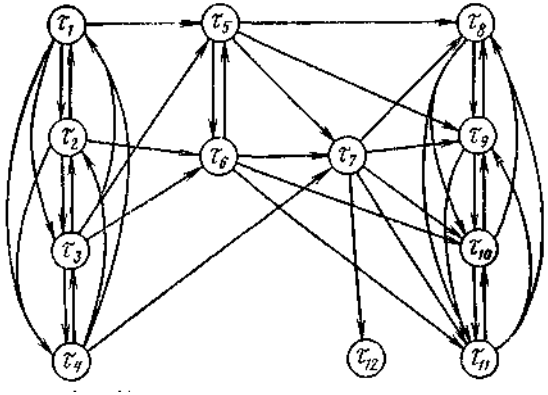


Рис. 19.9. Граф взаимосвязи операторов при сборке редуктора: $\tau_г$ — собрать узел входного вала; τ_2 — собрать узел вы-

ходного вала; τ_3 - собрать узел корпуса редуктора; τ_4 — собрать узел крышки редуктора; τ_5 — установить входной вал; τ_6 — установить выходной вал; τ_7 — собрать крышку редуктора с корпусом; τ_8 — собрать узел глухой крышки входного вала; τ_9 — собрать узел сквозной крышки входного вала; τ_{10} — собрать узел глухой крышки выходного вала; τ_{11} — собрать узел сквозной крышки выходного вала; $\tau_{12,}$ — собрать узел отдушины

Поэтому при автоматизированном проектировапин технологических процессов сборки одного и того же изделия используются, в зависимости от степени детализации процесса, различные модели S_l *(T):* модели операторов установки, соединения, доводки, регулировки и т. д.

Проектирование технологического процесса сборки выполняется поэтапно, путем последовательного решения задач различного уровня, начиная с определения состава этапов технологического процесса и заканчивая выбором оптимальных вариантов отдельных операций. Для этого создается многоуровневая математическая модель выполнения сборки, включающая: перестановочные модели для определения последовательности установки элементов изделия; табличные или сетевые модели этапов соединения и других операций с учетом последовательности операции установки элементов изделия; табличные или сетевые модели операций на различных этапах процесса соединения.

Взаимосвязь этих моделей обеспечивается через взаимосвязь контуров, а также путем включения смежных операций разных этапов сборки в модели обоих этапов. Основную роль в увязке различных этапов сборки и их моделей в единое целое выполняет - последовательность установки элементов сборочной единицы. Модель операторов установки чаще всего является перестановочной моделью класса S_3 *(T)* или S_9*(T).* Основными контурами, реализуемыми в процессе установки, являются контуры положения и — для нежестких элементов — контуры формы элементов сборочной единицы. Поэтому при определении возможных последовательностей установки элементов на первом этапе проектирования рассматривается упрощенная модель, состоящая из описания множества устанавливаемых элементов сборочной единицы и наборов условий базирования и доступа.

Последовательность выполнения операций установки элементов $a_i \in A$ можно представить как перестановку из n символов устанавливаемых элементов, удовлетворяющую условиям вида (19.12), (19.13). Возможные последовательности T_m установки могут также

определяться по графу-дереву вариантов установки, которое строится следующим образом.

1. Выбираются элементы $a_i \in A$, для которых выполняется условие $B_p\ (a_i) = a_i$; каждый такой элемент может устанавливаться первым — этот элемент принимается за корневую вершину дерева вариантов установки.

2. Рассматриваются все остальные, кроме корневой веашнны, элементы $a_j \in A$; те из них, для которых при наличии корневой вершины a_i выполняются условия базирования и доступа, могут устанавливаться вторыми. Корневая вершина a_i соединяется дугами с каждой вершиной a_j, которая может быть второй в последовательности T_m.

3. Рассматриваются все остальные, кроме корневой a_i и второй a_j вершины, элементы $a_k \in A$; те из них, для которых при наличии a_i и a_j выполняются условия базирования и доступа, могут устанавливаться третьими. Вершина a_j соединяется дугами с каждой вершиной a_k, которая может быть третьей в последовательности T_m. Затем рассматривается следующая вторая вершина дерева и т. д.

Построение дерева вариантов установки заканчивается, когда получены все висячие вершины, путь до которых от корневой вершины включает все элементы $a_i \in A$. Возможная последовательность установки T_m определяется как путь от корневой до висячей вершины; очередность установки элемента $a_i \in A$ определяется порядковым номером вершины a_i в пути, считая от корневой вершины дерева вариантов установки. Например, на рис.19.10 показано дерево вариантов установки элелгентов редуктора (см. рис.19.8).

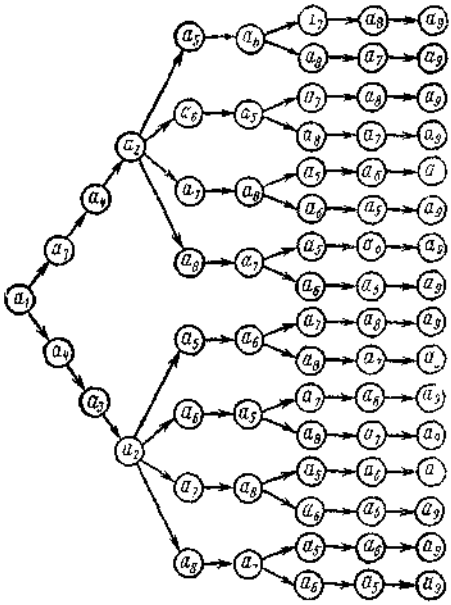


Рис.19.10 Дерево вариантов установки элементов редуктора

После определения возможных последовательностей установки проектируются процессы других этапов сборки изделия. При этом используются взаимосвязанные по составу контуров модели этапов сборки, описывающие эти этапы с различной степенью полноты и детализации. Например, на рис. 19.11 приведены дизъюнктивные модели этапа сборки крышки a_2 с корпусом a_1 редуктора (см. рис. 19. 8).

		F_1	F_2	F_3	F_4	F_5	F_6	F_7
Покрыть плоскости разъема герметиком	τ^{I}_{71}	●						
Установить крышку на корпусе	τ^{I}_{72}		●					
Соединить крышку с корпусом болтами	τ^{I}_{73}			●	●	●	●	
Ввернуть отжимные болты	τ^{I}_{74}							●

а)

		F_1	F_2	F_3	F_4	F_5	F_6	F_7
Покрыть плоскости разъема герметиком	τ^{II}_{71}	●						
Установить крышку на корпусе	τ^{II}_{72}		●					
Вставить болты	τ^{II}_{73}			●				
Наживить гайки вручную	τ^{II}_{74}				●			
Затянуть гайки вручную	τ^{II}_{75}					●		
Затянуть гайки гайковертом	τ^{II}_{76}					●		
Наживить и затянуть гайки гайковертом	τ^{II}_{77}				●	●		
Законтрить гайки	τ^{II}_{78}						●	
Ввернуть отжимные болты	τ^{II}_{79}							●

б)

в)

Рис. 19.11. Дизъюнктивные модели операторов сборки крышки с корпусом редуктора:
а — табличная модель операторов; б - матрица контуров сетевой модели операторои, *в* - граф взаимосвязи операторов сетевой модели

Модель на рис 19.11, *а* является табличной моделью класса S_1 *(T),* а модель на рис. 19.11, *б, в* — сетевой моделью класса S_2*(T).* Определение последовательности выполнения всех операций технологического процесса сборки осуществляется следующим образом: определяется последовательность установки элементов конструкции изделия; определяется состав операций соединения, доработки, контроля и т. д.; операции соединения, контроля и т. п. вводятся между операциями установки в соответствии с условиями их реализации.

Возможность выбора средств механизации и автоматизации операций, например соединения, непосредственно зависит не только от вида соединения, но и от геометрических свойств поверхностей, на которых расположены соединительные элементы, а также формы самих соединительных швов. Эти свойства контура соединения определяют требуемые кинематические свойства рабочих органов оборудования для выполнения операций соединения — форму траекторий движения

рабочих органов, область существования траекторий, состав и взаимосвязь возможных перемещений и т. д.

Область траекторий представляет собой линию, поверхность или объемную область пространства. Для характеристики области траекторий могут использоваться свойства образующего и направляющего элементов этой области (рис.19.12), отражающие форму этих элементов и уравнение D_i возможных перемещений точки относительно этих элементов.

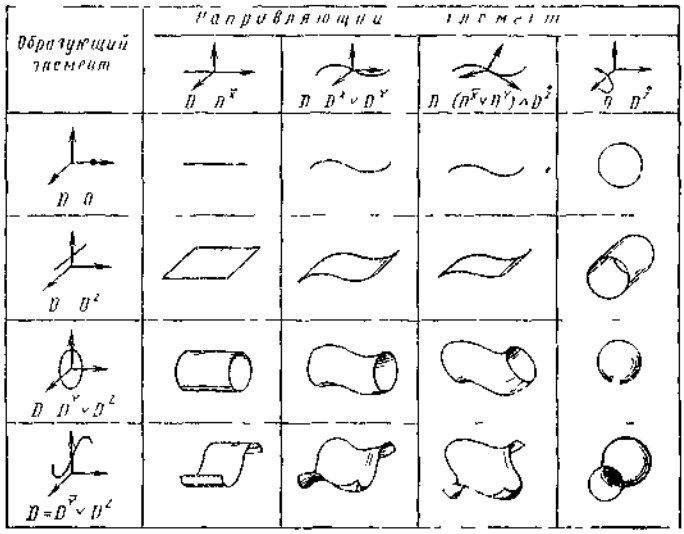


Рис. 19.12. Области траектории при различных образующих и направляющих элементах движения

Уравнение D_i, дополненное при необходимости матрицами возможных перемещений, входящих в это уравнение, достаточно полно характеризует кинематические связи элементов оборудования, инструмента и оснастки с требуемыми свойствами траекторий и областей траекторий при выполнении операций соединения.

Рассмотренная структура технологического процесса сборки относится к одной сборочной единице без учета сборки входящих в нее сборочных единиц низшего порядка. Технологический процесс сборки всего изделия с учетом процесса сборки входящих в него сборочных единиц низшего порядка имеет более сложную структуру и обычно описывается набором технологических процессов сборки всех

сборочных единиц низшего порядка и сборки изделия в целом, а взаимосвязь всех этапов сборки описывается схемой сборки изделия.

Схема сборки определяется по схеме технологического членения изделия на сборочные единицы. При этом для сборного элемента любого порядка

$$A_k^J = \{A_1^{J+1},\ A_2^{J+1}, \ldots,\ A_{n_k}^{J+1}\}$$

должно выполняться условие

$$\left.\begin{aligned} &\bigcup_{i=1}^{n_k} A_i^{J+1} = A_k^J; \\ &\forall i, j \in n_k, i \neq j\,(A_i^{J+1} \cap A_j^{J+1} \neq \varnothing). \end{aligned}\right\} \quad (19.14)$$

Однако состав сборочной единицы A^J_i удовлетворяет не только условию (19.14), но и технологическим требованиям, при выполнении которых возможна сборка A^J_i. Следовательно, формированию схемы технологического членения должно предшествовать определение возможных составов сборочных единиц низших порядков, каждый из которых удовлетворяет технологическим требованиям.

Необходимыми условиями существования сборочной единицы A^J_i являются: наличие в составе элементов сборочной единицы заданных звеньев тел контуров $F_i\ (A)$ в соответствии с техническими условиями на сборку агрегата; возможность сборки, т. е. установки и соединения каждого элемента $a_k \in A^J_i$ с другими элементами сборочной единицы.

Первое условие отражает правила технологического членения изделия с учетом функционального назначения его элементов, а второе условие — ограничения на сборку с учетом пространственной взаимосвязи, базирования и возможности соединения элементов сборочной единицы.

Любая сборочная единица $A^J_j = \{ a_{j_1}, a_{j_2}, \ldots, a_{j_r} \}$, являющаяся частью изделия A, представляет собой сочетание из n по r элементов $a_i \in A$, где $r = 2, 3, \ldots, n$. Поэтому вначале определяют все возможные сочетания элементов собираемого изделия и проверяют возможность их сборки. Если элементы, входящие в сочетание, могут быть собраны с соблюдением условий (19.12) и (19.13), то это сочетание может быть сборочной единицей.

Схема сборки показывает состав и взаимосвязь сборочных единиц, а иногда и сборочной оснастки и участков, соответствующих определенным этапам сборки. Схема сборки разрабатывается на основе схемы технологического членения изделия и используется для решения организационных задач сборки, определения цикла сборки изделия и т. д. Если схема сборки показывает только состав и взаимосвязь сборочных единиц, то она описывается графом G = (A^J, C)

с составом дуг (19.2) (см. рис. 19.2). Если схема сборки включает и элементы производственной системы, то полная схема сборки разрабатывается на основе графа G = (A^J, C) схемы технологического членения. Сборочная единица A^J_i, входящая в состав A, собирается на определенном рабочем месте (участке) Π^J_i. Полная схема сборки, включающая состав сборочных единиц $A^J_i \in A$, оснащенных сборочными приспособлениями и оборудованием рабочих мест $\Pi^J_i \in \Pi$, и структуру связей между элементами множеств A и П, описывается в виде графа

$$G = (\mathbf{A}, C);\ A^J_t \in \mathbf{A};\ \Pi^J_t \in \Pi \subset \mathbf{A}.$$

Каждой вершине A^J_i инцидентны два вида дуг — заходящая дуга $c_{i(i-)} = (\Pi^J_i, A^J_i)$ и исходящая дуга $c_{i(j)} = (A^J_i, \Pi^{J-1}_j)$, направленная к рабочему месту Π^{J-1}_j сборки элемента конструкции A^J_{j-1} высшего порядка. В каждую вершину Π^{J-1}_j заходит количество дуг вида $c_{i(j)}$, равное числу входящих в сборочную единицу A^J_{j-1} элементов A^J_i низшего порядка. Следовательно, если известен граф $G = (A^J, C)$ схемы технологического членения, то граф полной схемы сборки можно получить введением в A вершин П Э Π^J_i , заменой дуг $c_{i(j)} = (A^J_i, A^J_{j-1})$ дугами вида $c_{i(j)} =(A^J_i, \Pi^{J-1}_j)$ и добавлением дуг вида $c_{i(i)} = (\Pi^J_i, A^J_i)$.

В состав сборочных баз элементов конструкции сборочной единицы входят базовые элементы (базы) двух типов: собственные базы, являющиеся элементами конструкции сборочной единицы; несобственные базы, являющиеся базовыми элементами сборочной оснастки. Поскольку в конкретном технологическом процессе сборки для каждого элемента изделия может быть применен единственный состав сборочной базы, при наличии нескольких составов необходимо выбирать наилучший из них. Любое уравнение $B(a_i)$ можно, используя законы коммутативности, ассоциативности, дистрибутивности и идемпотентности логических функций, привести к виду

$$B(a_i) = B_1(a_i) \vee B_2(a_i) \vee \cdots \vee B_m(a_i), \quad (19.15)$$

где m — общее количество различных составов сборочных баз вида $B_p(a_i) = a_j \wedge \ldots \wedge a_k$.

Полный набор возможных составов сборочных баз всех элементов $a_i \in A$ определяется как набор всех различных покрытий булевой матрицы

$$[A \times B] = \begin{array}{c} \begin{array}{cccc} B_1 & B_2 & \ldots & B_m \end{array} \\ \begin{bmatrix} B_1(a_1) & B_2(a_1) & \ldots & B_m(a_1) \\ B_1(a_2) & B_2(a_2) & \ldots & B_m(a_2) \\ \ldots & \ldots & \ldots & \ldots \\ B_1(a_n) & B_2(a_n) & \ldots & B_m(a_n) \end{bmatrix} \end{array} \begin{array}{c} \\ a_1 \\ a_2 \\ \ldots \\ a_n, \end{array} \qquad (19.16)$$

где m — максимальное количество различных составов сборочных баз в уравнениях вида (19.15) элементов $a_i \in A$.

Каждое покрытие матрицы (19.16), удовлетворяющее условию

$$B_k(A) = B_{k_1}(a_1) \wedge B_{k_2}(a_2) \wedge \cdots \wedge B_{k_n}(a_n) = 1,$$

определяет возможную схему базирования при сборке A, если существует хотя бы одна такая последовательность установки всех $a_i \in A$, реализация которой возможна. Схема базирования отражает составы сборочных баз и взаимосвязь базируемых элементов изделия с элементами сборочных баз. Наглядное изображение этой взаимосвязи осуществляется с помощью графа базирования, в котором множество вершин есть множество элементов изделия и базовых элементов сборочной оснастки и дуги $c_{i(j)} \in C$ равны

$$c_{i(j)} = \begin{cases} 1, \text{ если } a_j \in B_k(a_i); \\ 0 \text{ — в остальных случаях.} \end{cases} \qquad (19.17)$$

Граф базирования получается путем введения в граф сопряжений элементов изделия вершин, соответствующих базовым элементам оснастки, и заменой дуг графа сопряжений дугами вида (19.17). Например, на рис. 19.3, *б* показан граф базирования деталей узла рис. 19.1; в этом графе элементы сборочной оснастки отсутствуют, так как базирование всех деталей обеспечивается по месту сопряжения с другими элементами конструкции узла.

Для каждой возможной схемы базирования строятся сборочные размерные цепи и рассчитывается точность сборки. Схемы базирования, не обеспечивающие требуемой точности сборки, исключаются из дальнейшего рассмотрения; схемы базирования, обеспечивающие точность сборки не ниже требуемой, включаются в состав расчетных схем базирования, рассматриваемых в дальнейшем при выборе оптимальных вариантов сборки.

Выбор схемы сборочной оснастки осуществляется следующим образом. Сборочное приспособление, рассматриваемое как единая механическая система, состоит из двух основных подсистем — базовых элементов и несущих элементов. Элементы приспособления, поверхности которых входят в состав установочных баз, называются базовыми элементами, а совокупность этих элементов образует

систему базовых элементов (БЭС) сборочного приспособления. Базовые, кондукторные и другие элементы могут размещаться в каркасе приспособления, представляющем собой систему несущих элементов (НЭС), объединяющую все элементы сборочного приспособления в единую механическую систему.

Структура контуров сборочного приспособления определяется его связью с контурами собираемого изделия. Взаимосвязь элементов БЭС и собираемого изделия реализуется через взаимосвязь, контуров изделия, сопрягаемых с элементами БЭС, и контуров установочных баз элементов БЭС. Поэтому, если булев вектор $F\ (A)_{k-1}$ характеризует состав контуров изделия перед k-м этапом сборки, то взаимосвязь изделия и сборочной оснастки характеризуется составом булева вектора

$$\bar{F}(A)_{k-1} = (\bar{F}_1, \bar{F}_2, \ldots, \bar{F}_n), \quad (19.18)$$

где $\bar{F}\ (A)_{k-1}$ — множество всех контуров элементов изделия, по которым они могут быть дополнены элементами БЭС оснастки. Классификационными признаками БЭС являются свойства, отражающие взаимосвязь элементов этой системы с изделием и системой несущих элементов (НЭС) приспособления. Для конкретного сборочного приспособления $(B_{СП})$ состав контуров установочных баз характеризуется булевым вектором

$$F_{БЭС}\,(B_{СП})_k = (\bar{F}_1, \bar{F}_2, \ldots, \bar{F}_n), \quad (19.19)$$

компоненты $\bar{F}_i = 1$ которого образуют подмножество компонент $\bar{F}_i = 1$ вектора (19.18). Поэтому основным классификационным признаком системы БЭС является состав вектора (19.19), т. е. состав контуров, сопрягаемых с контурами собираемого изделия.

Состав базовых элементов $B_{БЭС}$ является множеством исполнительных звеньев оснастки, реализующих контуры (19.19). Состав всех возможных конструктивно различных элементов $b_i \in B_{БЭС}$ представляется в виде булевой матрицы

$$\|c_{i\ (j)}\|_{B_{БЭС},\ \bar{F}\ (A)} = [B_{БЭС} \times \bar{F}\ (A)]. \quad (19.20)$$

Элементы БЭС, определяемые по матрице (19.20), включаются в составы сборочных баз уравнения (19.15).

Полный состав контуров $F\ (b_i)$ базового элемента включает рабочие контуры $\{\bar{F}_1, \bar{F}_2, \ldots, \bar{F}_n\}$, а также контуры $\{F_{n+1}, F_{n+2}, \ldots F_N\}$ сопряжения b_i с элементами НЭС, контуры, обусловленные особенностями конструкции b_i, и т. п. Поэтому состав и свойства всех

конструктивно различных элементов $b_i \in B_{БЭС}$ описываются булевой матрицей контуров

$$[B_{БЭС} \times F(B)] =$$

$$= \begin{array}{c} \begin{array}{cccccccc} \overline{F}_1 & \cdots & \overline{F}_n & \cdots & F_{n+1} & \cdots & F_N \end{array} \\ \begin{bmatrix} c_1(1) & \cdots & c_1(n) & \cdots & c_1(n+1) & \cdots & c_1(N) \\ c_2(1) & \cdots & c_2(n) & \cdots & c_2(n+1) & \cdots & c_2(N) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_k(1) & \cdots & c_k(n) & \cdots & c_k(n+1) & \cdots & c_k(N) \end{bmatrix} \end{array} \begin{array}{c} b_1 \\ b_2 \\ \cdot \\ b_k \end{array} \tag{19.21}$$

Основными функциональными контурами элементов $b_i \in B_{БЭС}$ являются контуры, определяющие положение базовых элементов сборочного приспособления, и контуры, обеспечивающие функционирование приспособления как единой системы. Состав и свойства элементов $B_{НЭС}$ описываются булевыми матрицами контуров, аналогичными (19.20)—(19.21).

Математическая модель S (*П*) сборочного приспособления включает в себя описание состава элементов $B_{сп}$, состава и взаимосвязи контуров элементов приспособления, а также графы сопряжений элементов приспособления, графы базирования элементов при монтаже приспособления и т. п. Способы математического моделирования сборочных приспособлений аналогичны моделированию изделия и технологических процессов. Например, для выбора состава базовых элементов конструкции приспособления на основе матрицы контуров (19.21) используется сочетательная модель класса S^0_3 (П) либо сетевая модель класса S_5 (*П*), граф $G = (B_{БЭС}, C)$ которой служит для формирования возможных наборов базовых элементов приспособления. Поэтому при выборе схемы сборочного приспособления используются те же алгоритмы, что и при решении других задач технологического проектирования.

К основным технико-экономическим показателям сборки относятся трудоемкость, технологическая себестоимость и цикл сборки. Факторами, влияющими на трудоемкость и технологическую себестоимость сборки при различных вариантах технологического процесса, являются составы сборочных баз, последовательность установки элементов конструкции и варианты выполнения остальных операций сборки. Информация о способах базирования элементов $a_i \in A$ вместе с информацией о конструктивно-технологических свойствах элементов a_i вполне достаточна для определения трудоемкости и технологической себестоимости сборки с применением ЭВМ.

Оценку вариантов сборки по циклу изготовления агрегата осуществить значительно труднее, чем оценку варианта сборки по трудоемкости или технологической себестоимости. Это объясняется тем, что цикл сборки зависит в основном от структуры схемы сборки, так как эта структура определяет возможную очередность выполнения сборочных работ. Поэтому для определения цикла используется технологическая сеть, отражающая последовательность выполнения технологических операций. Если технологическая сеть построена, то для определения цикла можно использовать методы сетевого планирования и управления, поскольку минимальный цикл можно представить как критическое время выполнения всего комплекса работ. Для использования алгоритмов сетевого планирования схема сборки превращается в сеть добавлением фиктивной исходной вершины и дуг, соединяющих эту вершину со всеми висячими вершинами графа-схемы сборки, за исключением конечной вершины.

19.3. Образование фрейма операционного технологического процесса сборки

При проектировании технологического процесса сборки на ЭВМ вначале по сетевой модели производственной системы (например вида, показанного на рис. 19.7) определяется состав возможных операций, оснастки, инструмента и оборудования, необходимого для реализации контуров сборочной единицы. Затем проверяется условие существования операций (хотя бы в единственном варианте) для реализации каждого контура. Если это условие не выполняется, то состав контуров, которые не могут быть реализованы в данной производственной системе, выдается на печать как результат диагностирования недостающих свойств производственной системы. Чтобы сборка изделия стала возможной, производственная система должна быть дополнена недостающими операциями, оборудованием, инструментом и оснасткой, позволяющими реализовать все контуры изделия.

Далее, для возможных операций технологических процессов осуществляется нормирование трудоемкости, расчет технологической себестоимости и, при необходимости, других технико-экономических показателей. Возможен случай, когда в памяти ЭВМ не окажется некоторых нормативов, необходимых для нормирования возможных операций технологических процессов. Тогда состав контуров изделия и относящихся к ним операций, не обеспеченных нормативными

данными, выдается на печать как результат диагностирования недостающих нормативных данных в производственной системе сборки.

Если свойства производственной системы достаточны для сборки данного изделия, ЭВМ автоматически переходит к следующим этапамо бразования фреймов: выбору оптимального состава операций, формированию технологического процесса сборки в соответствии с заданной последовательностью выполнения операций и расчету цикла сборки. В качестве критериев оптимальности могут назначаться трудоемкость, технологическая себестоимость или цикл сборки. Оптимальный технологический процесс выдается на печать в виде маршрутных карт технологического процесса сборочных работ. Одновременно печатаются исходные данные об изделии и материалы по результатам промежуточных вычислений: допустимые варианты технологических процессов сборки; оценка допустимых вариантов по трудоемкости и технологической себестоимости сборки; содержание и технико-экономические параметры операций оптимального технологического процесса сборки; последовательность операций; расчет цикла сборки.

Литература

1. В.Ф.Асмус. Проблема интуиции в философии и математике. «Мысль», М.1965.

2. Дж. Барвайс. Введение в логику первого порядка. Справочная книга по математической логике. Ч.1. «Наука», М.1982. Пер. с англ..: Handbook of mathematical logic. J. Barwise (Ed). North-Holland P.C. 1977.

3. Дж.Булос, Р.Джеффри. Вычислимость и логика. М. «Мир» 1994. Пер. с английского: George S. Boolos, Richard C. Jeffrey. Computability and logic. Cambridge University press, 1989.

4. Е.А.Беляев, В.Я.Перминов. Философские и методологические проблемы математики. Изд-во Московского университета, 1981.

5. М.Бунге. Интуиция и наука. «Прогресс», М. 1967. Пер. с английского: M.Bunge. Intuition and Science. New York, 1962.

6. Н.Бурбаки. Начала математики. Ч.1, кн.1. Теория множеств. Мир. М. 1965. Пер. с французского.: Elements de Mathematique par N.Bourbaki. Livre 1. Theorie des ensembles. Troisieme edition, 1958.

7. Е.Вигнер. Непостижимая эффективность математики в естественных науках. УФН, т.94, вып.3, 1968, 535 – 546. Пер. с англ.: E.Wigner. The Unreasonable Effectiveness of Mathematics in the Natural Sciences, Comm. Pure and Appl. Math. 131, 1 (1960).

8. Р.Декарт. Правила для руководства ума. Избранные произведения. М.1950. Пер. с французского: Descartes R. Oeuvres, t. X. Paris, 1908.

9. М.Клайн. Математика. Утрата определённости. М.”Мир”, 1984. Пер. с англ.: Morris Kline. MATHEMATICS. The Loss of Certainty. N-Y, Oxford University Press, 1980.

10. С.К.Клини. Введение в метаматематику. ИЛ М. 1957. Пер.с англ. : Introduction tu metamathematics by Stephen Cole Kleene. 1952. D.van

Nostrand Company, inc. New York , Toronto.

11. М.Кац, С.Улам. Математика и логика. Ретроспектива и перспективы. «Мир», М. 1971. Пер. с англ.: Mathematics and Logic. Retrospect and Prospects. Mark Kac and Stanislaw M. Ulam. N.-Y. Washington. London. 1968.

12. А.Е. Кононюк. Общая теория познания и созидания. Кн.1. Киев: «Освіта України», 2013. 648 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

13. А.Е. Кононюк. Общая теория познания и созидания. Кн.2, ч.1. Киев: «Освіта України», 2013. 544 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

14. А.Е. Кононюк. Общая теория познания и созидания. Кн.2, ч.2. Киев: «Освіта України», 2013. 644 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

15. А.Е. Кононюк. Информациология. Общая теория информации. Кн.1. Киев: «Освіта України», 2011. 476 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

16. А.Е. Кононюк. Информациология. Общая теория информации. Кн.2. Киев: «Освіта України», 2011. 476 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

17. А.Е. Кононюк. Информациология. Общая теория информации. Кн.3. Киев: «Освіта України», 2011. 412 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

18. А.Е. Кононюк. Информациология. Общая теория информации. Кн.4. Киев: «Освіта України», 2011. 488 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

19. А.Е. Кононюк. Общая теория понятий. Кн.1. Киев: «Освіта України», 2014. 514с.

20. А.Е. Кононюк. Общая теория понятий. Кн.2. Киев: «Освіта України», 2014. 544с.

21. А.Е. Кононюк. Общая теория понятий. Кн.3. Киев: «Освіта України», 2014. 614с.

22. А.Е. Кононюк. Системология. Общая теория систем. Кн.1. Киев: «Освіта України», 2012. 564с. *ecat.diit.edu.ua:81/ft/index_ru.html*

23. А.Е. Кононюк. Системология. Общая теория систем. Кн.2. Ч.1. Киев: «Освіта України», 2014. 558с.
ecat.diit.edu.ua:81/ft/index_ru.html
24. А.Е. Кононюк. Системология. Общая теория систем. Кн.2. Ч.2. Киев: «Освіта України», 2014. 658с.
ecat.diit.edu.ua:81/ft/index_ru.html
25. А.Е. Кононюк. Системология. Общая теория систем. Кн.2. Ч.1. Киев: «Освіта України», 2014. 558с.
ecat.diit.edu.ua:81/ft/index_ru.html
26. А.Е. Кононюк. Общая теория распознавания. Кн.1. Киев: «Освіта України», 2012. 584 с. *ecat.diit.edu.ua:81/ft/index_ru.html*
27. А.Е. Кононюк. Общая теория распознавания. Кн.2. Киев: «Освіта України», 2012. 588 с. *ecat.diit.edu.ua:81/ft/index_ru.html*
28. А.Е. Кононюк. Консалтология. Общая теория консалтинга. Кн.1. Киев: «Освіта України», 2013. 448 с.
ecat.diit.edu.ua:81/ft/index_ru.html
29. А.Е. Кононюк. Консалтология. Общая теория консалтинга. Кн.2. Киев: «Освіта України», 2013. 412 с.
ecat.diit.edu.ua:81/ft/index_ru.html
30. А.Е. Кононюк. Консалтология. Общая теория консалтинга. Кн.3. Киев: «Освіта України», 2013. 520 с.
ecat.diit.edu.ua:81/ft/index_ru.html
31. А.Е. Кононюк. Консалтология. Общая теория консалтинга. Кн.4. Киев: «Освіта України», 2013. 508 с.
ecat.diit.edu.ua:81/ft/index_ru.html
32. А.Е. Кононюк. Дискретно-непрерывная математика. Начала. Кн.1. Киев: «Освіта України», 2012. 652с.
ecat.diit.edu.ua:81/ft/index_ru.html
33. А.Е. Кононюк. Дискретно-непрерывная математика. Множества. Кн.2. Ч.1. Киев: «Освіта України», 2012. 452с.
ecat.diit.edu.ua:81/ft/index_ru.html
34. А.Е. Кононюк. Дискретно-непрерывная математика. Множества. Кн.2. Ч.2. Киев: «Освіта України», 2013. 536 с.
ecat.diit.edu.ua:81/ft/index_ru.html
35. А.Е. Кононюк. Дискретно-непрерывная математика. Отношения. Кн.3. Ч. 1. Киев: «Освіта України», 2013. 552с.
ecat.diit.edu.ua:81/ft/index_ru.html

36. А.Е. Кононюк. Дискретно-непрерывная математика. Отношения. Кн.3. Ч. 2. Киев: «Освіта України», 2013. 548 с.
ecat.diit.edu.ua:81/ft/index_ru.html
37. А.Е. Кононюк. Дискретно-непрерывная математика. Алгебры. Кн.4. Ч.1. Киев: «Освіта України», 2011. 452с.
ecat.diit.edu.ua:81/ft/index_ru.html
38. А.Е. Кононюк. Дискретно-непрерывная математика. Алгебры. Кн.4. Ч.2. Киев: «Освіта України», 2011. 668 с.
ecat.diit.edu.ua:81/ft/index_ru.html

39. А.Е. Кононюк. Дискретно-непрерывная математика. Алгебры. Кн.4. Ч.3. Киев: «Освіта України», 2015. 488 с.
http://lib.sumdu.edu.ua/library/DocDescription?doc_id=640902

40. А.Е. Кононюк. Дискретно-непрерывная математика. Алгебры. Кн.4. Ч.4. Киев: «Освіта України», 2015. 548 с.

41. А.Е. Кононюк. Дискретно-непрерывная математика. Алгебры. Кн.4. Ч.5. Киев: «Освіта України», 2015. 528 с.

42. А.Е. Кононюк. Дискретно-непрерывная математика. Алгебры. Кн.4. Ч.6. Киев: «Освіта України», 2015. 608 с.

43. А.Е. Кононюк. Дискретно-непрерывная математика. Матрицы. Кн.5. Ч.1. Киев: «Освіта України», 2013. 612 с.
ecat.diit.edu.ua:81/ft/index_ru.html

44. А.Е. Кононюк. Дискретно-непрерывная математика. Матрицы. Кн.5. Ч.2. Киев: «Освіта України», 2013. 500 с.
ecat.diit.edu.ua:81/ft/index_ru.html

45. А.Е. Кононюк. Дискретно-непрерывная математика. Матрицы. Кн.5. Ч.3. Киев: «Освіта України», 2013. 520 с.
ecat.diit.edu.ua:81/ft/index_ru.html
46. А.Е. Кононюк. Дискретно-непрерывная математика. Матрицы. Кн.5. Ч.4. Киев: «Освіта України», 2013. 508 с.
ecat.diit.edu.ua:81/ft/index_ru.html

47. А.Е. Кононюк. Дискретно-непрерывная математика. Матрицы. Кн.5. Ч.5. Киев: «Освіта України», 2013. 672 с.
ecat.diit.edu.ua:81/ft/index_ru.html
48. А.Е. Кононюк. Дискретно-непрерывная математика. Поверхности. Кн.6. Ч.1. Киев: «Освіта України», 2012. 652с.
ecat.diit.edu.ua:81/ft/index_ru.html
49. А.Е. Кононюк. Дискретно-непрерывная математика. Графы. Кн.7. Ч.1 Киев: «Освіта України», 2014. 652с.
ecat.diit.edu.ua:81/ft/index_ru.html
50. А.Е. Кононюк. Дискретно-непрерывная математика. Графы. Кн.7. Ч.2 Киев: «Освіта України», 2014. 552с.
ecat.diit.edu.ua:81/ft/index_ru.html
51. А.Е. Кононюк. Дискретно-непрерывная математика. Графы. Кн.7. Ч.3 Киев: «Освіта України», 2015. 512с.
ecat.diit.edu.ua:81/ft/index_ru.html
52. А.Е. Кононюк. Дискретно-непрерывная математика. Графы. Кн.7. Ч.4 Киев: «Освіта України», 2015. 552с.
ecat.diit.edu.ua:81/ft/index_ru.html

53. А.Е. Кононюк. Дискретно-непрерывная математика. Графы. Кн.7. Ч.5 Киев: «Освіта України», 2015. 660с.

54. А.Е. Кононюк. Обобщенная теория моделирования. Кн.1. Ч.1 Киев: «Освіта України», 2012. 602с.
ecat.diit.edu.ua:81/ft/index_ru.html
55. А.Е. Кононюк. Обобщенная теория моделирования. Кн.1. Ч.2 Киев: «Освіта України», 2012. 708с. *ecat.diit.edu.ua:81/ft/index_ru.html*
56. А.Е. Кононюк. Обобщенная теория моделирования. Кн.1. Ч.3 Киев: «Освіта України», 2012. 568с. *ecat.diit.edu.ua:81/ft/index_ru.html*
57. А.Е. Кононюк. Обобщенная теория моделирования. Кн.2. Киев: «Освіта України», 2012. 548с. *ecat.diit.edu.ua:81/ft/index_ru.html*
58. А.Е. Кононюк. Обобщенная теория моделирования. Кн.3. Ч.1 Киев: «Освіта України», 2012. 636с. *ecat.diit.edu.ua:81/ft/index_ru.html*
59. А.Е. Кононюк. Обобщенная теория моделирования. Кн.3. Ч.2 Киев: «Освіта України», 2012. 448с. *ecat.diit.edu.ua:81/ft/index_ru.html*
60. А.Е. Кононюк. Обобщенная теория моделирования. Кн.3. Ч.3 Киев: «Освіта України», 2013. 588с. *ecat.diit.edu.ua:81/ft/index_ru.html*
61. А.Е. Кононюк. Основы теории оптимизации. Кн.1. Киев: «Освіта України», 2011. 602с. *ecat.diit.edu.ua:81/ft/index_ru.html*

62. А.Е. Кононюк. Основы теории оптимизации. Кн.2. Ч.1. Киев: «Освіта України», 2011. 552с. *ecat.diit.edu.ua:81/ft/index_ru.html*
63. А.Е. Кононюк. Основы теории оптимизации. Кн.2. Ч.2. Киев: «Освіта України», 2011. 616с. *ecat.diit.edu.ua:81/ft/index_ru.html*
64. А.Е. Кононюк. Основы теории оптимизации. Кн.2. Ч.3. Киев: «Освіта України», 2012. 456с. *ecat.diit.edu.ua:81/ft/index_ru.html*
65. А.Е. Кононюк. Основы теории оптимизации. Кн.2. Ч.4. Киев: «Освіта України», 2012. 512с. *ecat.diit.edu.ua:81/ft/index_ru.html*
66. А.Е. Кононюк. Основы научных исследований. Кн.1. Киев: «Освіта України», 2011. 508с. *ecat.diit.edu.ua:81/ft/index_ru.html*
67. А.Е. Кононюк. Основы научных исследований. Кн.2. Киев: «Освіта України», 2011. 452с. *ecat.diit.edu.ua:81/ft/index_ru.html*
68. А.Е. Кононюк. Основы научных исследований. Кн.3. Киев: «Освіта України», 2011. 456с. *ecat.diit.edu.ua:81/ft/index_ru.html*
69. А.Е. Кононюк. Основы научных исследований. Кн.4. Киев: «Освіта України», 2011. 456с. *ecat.diit.edu.ua:81/ft/index_ru.html*

70. А.Е. Кононюк. Общая теория коммуникаций. Кн.1. Киев: «Освіта України», 2014. 488с.

71. А.Е. Кононюк. Нейроні мережі і генетичні алгоритми. Киев: «Корнійчук», 2010. 448с. *ecat.diit.edu.ua:81/ft/index_ru.html*

72. Кононюк А. Е. Обобщенная теория познания и созидания. [В 2 кн.] Кн. 1 : Начала / А. Е. Кононюк. — Киев : Освіта України, 2013. *ecat.diit.edu.ua:81/ft/index_ru.html*

73. Кононюк А. Е. Обобщенная теория познания и созидания. [В 2 кн.] Кн. 2 : Теория познания. Ч. 1 / А. Е. Кононюк. — Киев : Освіта України, 2013 *ecat.diit.edu.ua:81/ft/index_ru.html*

74. Кононюк А. Ю. Вища математика. (Модульна технологія навчання) : навчальний посібник : в 2 кн. / А. Ю. Кононюк. — Київ : КНТ, 2009 — Кн. 1. — 2009. — 702 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

75. Кононюк А. Ю. Вища математика. (Модульна технологія навчання) : навчальний посібник : в 2 кн. / А. Ю. Кононюк. — Київ :

КНТ, 2009 Кн. 2. — 2009. — 790 с. *ecat.diit.edu.ua:81/ft/index_ru.html*

76. А.Е. Кононюк. Дискретно-непрерывная математика. Поверхности. Кн.6. Ч.2. Киев: «Освіта України», 2012. 652с. http://www.dut.edu.ua/ua/lib/127/category/96/view/1297

77. А.Е. Кононюк. Дискретно-непрерывная математика. Пространства. Кн.8. Ч.1. Киев: «Освіта України», 2016. 748 с. http://www.dut.edu.ua/ua/lib/1/category/96/view/1439

78. А.Е. Кононюк. Дискретно-непрерывная математика. Пространства. Кн.8. Ч.2. Киев: «Освіта України», 2016. 480с. http://lib.sumdu.edu.ua/library/DocDescription?doc_id=640775

79. А.Е. Кононюк. Истины и информация (фундаментальная теория представления истин и информации). К.1. Киев: «Освіта України», 2016. 568с.

80. А.Е. Кононюк. Истины и информация (фундаментальная теория представления истин и информации). К.2. Киев: «Освіта України», 2016. 558с.

81. А.Е. Кононюк. Истины и информация (фундаментальная теория представления истин и информации). К.3. Киев: «Освіта України», 2016. 588с.

82. А.Е. Кононюк. Истины и информация (фундаментальная теория представления истин и информации). К.4. Киев: «Освіта України», 2016. 552с

83. А.Е. Кононюк. Истины и информация (фундаментальная теория представления истин и информации). К.5. Киев: «Освіта України», 2016. 836 с

84. А.Е. Кононюк. Истины и информация (фундаментальная теория представления истин и информации). К.6. Киев: «Освіта України», 2016. 576 с