

Н. Е. КОБРИНСКИЙ и Б. А. ТРАХТЕНБРОТ

ВВЕДЕНИЕ В ТЕОРИЮ КОНЕЧНЫХ АВТОМАТОВ



ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

МОСКВА 1962

АННОТАЦИЯ

В последние годы повсеместно и с большой интенсивностью ведутся работы по созданию и применению различных автоматических систем дискретного действия для переработки информации. Они лежат в основе быстродействующих цифровых вычислительных машин, автоматических устройств для управления объектами и систем, моделирующих деятельность живого организма (так называемых роботов). Производство кибернетических автоматов растет быстрыми темпами, непрерывно расширяются области их применения.

Книга посвящена изложению общих вопросов теории автоматов дискретного действия, имеющих конечную память, которые называются конечными автоматами. Изложению теории предпосылаются краткие сведения по алгебре логики и логике предикатов, необходимые для решения задач анализа и синтеза автоматов. Описываются физические элементы: электронно-ламповые, полупроводниковые и магнитные, осуществляющие простейшие логические операции и являющиеся «ячейками» автомата. Приводятся общие методы анализа конечных автоматов.

Большое внимание уделено синтезу конечных автоматов. Излагаются общие методы синтеза, основанные на графических и аналитических способах задания операторов. Описываются различные практические приемы синтеза, которые иллюстрируются большим числом примеров.

Книга предназначена для студентов, аспирантов и научных работников, специализирующихся в теории кибернетических систем, и инженеров, занимающихся разработкой цифровых автоматических устройств—вычислительных и управляющих.

Кобринский Натан Ефимович и Трахтенброт Борис Авраамович.

Введение в теорию конечных автоматов.

М., Физматгиз, 1962 г., 404 стр. с илл.

Редактор Бирюков Б. В.

Техн. редактор Мурашова Н. Я.

Корректор Халанская А. Д.

Сдано в набор 6/IX 1961 г. Подписано к печати 9/III 1962 г.
Бумага 84×1081/32. Физ. печ. л. 12,625. Условн. печ. л. 20,71. Уч.-изд.
л. 19,66. Тираж 15 000 экз. Т-00973. Цена книги 1 р. 18 к. Заказ № 1246.

Государственное издательство физико-математической литературы.
Москва, В-71, Ленинский проспект, 15.

Московская типография № 5 Мосгорсовнархоза
Москва, Трехпрудный пер., 9.

ОГЛАВЛЕНИЕ

Предисловие	5
Введение	9
Глава I. Элементы логики	23
§ 1. Общие замечания	23
§ 2. Элементы алгебры логики	27
§ 3. Минимизация формул алгебры логики	45
§ 4. Разложение функций алгебры логики	59
§ 5. Элементы логики предикатов	67
Глава II. Операторы. Логические сети	84
§ 1. Общие замечания	84
§ 2. Детерминированность операторов	89
§ 3. Ограниченно-детерминированные операторы (опе- раторы с конечным весом)	96
§ 4. Элементы (элементарные автоматы) и схемы	106
§ 5. Логические сети	115
§ 6. Реализуемость ограниченно-детерминированных опе- раторов. Задачи анализа и синтеза автоматов	124
Глава III. Операторы физических элементов	130
§ 1. Вводные замечания	130
§ 2. Электронно-ламповые и полупроводниковые элементы	131
§ 3. Триггерные элементы	154
§ 4. Ферромагнитные элементы	168
Глава IV. Анализ автоматов	190
§ 1. Общие замечания	190
§ 2. Достижимость	194
§ 3. Различимость	199
§ 4. Периодичность	204
§ 5. Анализ схемы автомата	210
Глава V. Способы задания операторов	219
§ 1. Общие замечания	219
§ 2. Фрагменты деревьев	224
§ 3. Матрицы перехода	238

§ 4. Еще о языке логики суждений	245
§ 5. Язык логики предикатов	254
§ 6. О выразительности логического языка	275
Глава VI. Практические приемы синтеза автоматов . . .	281
§ 1. Общие замечания	281
§ 2. Синтез одновыходной логической сети	284
§ 3. Синтез многовыходной логической сети	318
§ 4. Синтез логической сети «мышь в лабиринте» . . .	334
Глава VII. Асимптотическая оценка сложности логических сетей	347
§ 1. Основные понятия. Формулировка предельной теоремы	347
§ 2. Синтез сети, реализующей функцию алгебры логики; верхняя оценка сложности сети	356
§ 3. Операторы, допускающие простую реализацию . . .	368
§ 4. Выбор кода	382
§ 5. Синтез сетей с памятью	389
Добавление	395
Литература	399
Предметный указатель	403



ПРЕДИСЛОВИЕ

В последние годы повсеместно и с большой интенсивностью ведутся работы по созданию и применению различных автоматических систем дискретного действия для переработки информации. Такие системы лежат в основе цифровых вычислительных машин, автоматических устройств для управления объектами по заданному алгоритму и агрегатов, моделирующих деятельность живого организма (так называемых роботов). Эти автоматы реализуются в виде самостоятельных устройств специализированного назначения или в виде блоков, входящих в системы регулирования и управления, и строятся на различных элементах—электромеханических реле, электронно-ламповых, полупроводниковых, магнитных и др. Производство различных типов автоматов дискретного действия увеличивается быстрыми темпами. Области их применения непрерывно расширяются. Вместе с тем все более важное значение приобретает разработка теории систем дискретного действия.

Теория систем дискретного действия охватывает широкий круг вопросов, связанных с исследованием возможности автоматизации тех или иных процессов переработки информации, повышением производительности и надежности работы автоматов и созданием эффективных методов их анализа и синтеза. Разработке этих проблем уделяется в настоящее время большое внимание как в Советском Союзе, так и за рубежом. В связи с этим в отечественной и зарубежной периодической литературе публикуется большое число статей по различным вопросам теории автоматов. Однако работ монографического характера по указанной проблематике пока крайне мало. Между тем потребность в литературе, знакомящей широкий круг

читателей с современным состоянием теории автоматов дискретного действия, беспрерывно возрастает.

Указанные причины побудили авторов сделать попытку систематически изложить основные вопросы теории так называемых конечных автоматов и при этом четко разграничить различные аспекты этой теории.

В первой главе книги в краткой форме сообщаются сведения по математической логике, необходимые для последующего изложения вопросов анализа и синтеза логической сети. Сюда включены основные понятия и некоторые теоремы логики суждений (алгебры логики) и наиболее распространенные приемы минимизации формул алгебры логики. Авторы сочли также целесообразным включить в эту главу элементы логики предикатов, используемой в качестве формального языка для описания операторов, реализуемых автоматами.

Во второй главе рассматриваются операторы переработки дискретной информации и общие вопросы построения логической сети с помощью наборов функциональных элементов сети, а также формулируется и обосновывается утверждение о физической реализуемости конечного автомата.

В третьей главе описываются операторы физических элементов логической сети. Авторы ограничились рассмотрением бесконтактных элементов, получивших наиболее широкое практическое применение в дискретных автоматах, — именно, электронно-ламповых, полупроводниковых и ферромагнитных элементов. Основное содержание главы посвящено описанию логических свойств элементов. Физические стороны работы элементов освещаются лишь в той мере, в какой это необходимо для объяснения их функционирования в качестве ячеек логической сети.

Четвертая глава посвящена анализу автоматов. Излагаемая процедура позволяет установить, обеспечивает ли рассматриваемая схема детерминированную переработку информации или нет, и в случае утвердительного ответа дает канонические уравнения оператора, реализуемого схемой. В связи со спецификой кодирования информации в схеме часто возникает избыточность памяти, и при анализе схемы иногда возникает необходимость выявления этой избыточности и ее устранения. Эти вопросы также рассматриваются в четвертой главе. Наконец, здесь же

исследуются некоторые общие свойства операторов и, в частности, свойства, связанные с переработкой периодических входных последовательностей.

Последние три главы книги посвящены проблеме синтеза автоматов. В пятой главе рассматриваются вопросы так называемого абстрактного синтеза, заключающегося в переходе от первоначальной записи условий работы проектируемого автомата к стандартному заданию (посредством канонических уравнений) автоматного оператора этого автомата. Здесь же рассматривается возможность использования языка логики предикатов для описания синтезируемого оператора.

Переход от канонических уравнений оператора к схеме, построенной из элементов заданного типа и реализующей этот оператор, относится уже к структурному синтезу автоматов, который исследуется в последующих главах. В шестой главе рассматриваются приемы синтеза логических сетей с применением различных элементов, наиболее широко используемые на практике. Здесь же разбирается большое количество примеров синтеза схем, в том числе и синтез схемы «мышь в лабиринте».

Основное содержание седьмой главы посвящено исследованию эффективных методов синтеза оптимальных логических сетей, не сводящихся к перебору множества сетей. Эти методы базируются на асимптотических оценках функции «стоимости» логической сети. Здесь же рассматриваются некоторые практически важные операторы, допускающие существенно более простую реализацию, чем та, которая установлена предельной теоремой для почти всех операторов. Наконец, исследуется возможность улучшения синтеза путем надлежащего кодирования информации.

При написании книги авторами были использованы материалы курсов лекций по математической логике и теории вычислительных машин, неоднократно читанных студентам Пензенского политехнического и Пензенского педагогического институтов. Для понимания текста от читателей не требуется специальной математической подготовки. Сведения, сообщаемые в курсах математики высших технических учебных заведений, готовящих специалистов по автоматике, телемеханике и вычисли-

тельной технике, достаточны для понимания настоящей книги.

Естественно, что ориентация на широкий круг читателей, а также объем книги определили отбор материала и методику изложения. В частности, пришлось отказаться от изложения ряда результатов (в том числе и собственных), относящихся к проблематике, рассматриваемой в главах V и VII, и представляющих, по-видимому, интерес для более узкого круга математиков. Вместе с тем для этой категории читателей принятое нами изложение местами может представляться чрезмерно пространным. Мы не смогли также достаточно отразить в книге и результаты, полученные в самое последнее время (основной текст книги был написан в 1958 г.). По всем этим причинам эту книгу можно рассматривать лишь как введение в бурно развивающуюся теорию дискретных автоматов, что и отражено в ее названии.

Введение, главы первая (кроме § 5), третья и шестая написаны Н. Е. Кобринским, главы вторая, четвертая, пятая и седьмая, а также § 5 главы первой — Б. А. Трахтенбротом.

При подготовке книги к печати нам оказали большую помощь С. В. Яблонский, О. Б. Лупанов, М. Л. Цетлин и Л. М. Шехтман, давшие ряд полезных советов и любезно предоставившие в наше распоряжение материалы своих работ по вопросам, затрагиваемым в настоящей книге. С. В. Яблонский, Ю. И. Журавлев, Н. В. Белякин и А. И. Барышев внимательно просмотрели рукопись и сделали ценные замечания по содержанию и методике изложения. Весьма тщательное и квалифицированное редактирование рукописи выполнил Б. В. Бирюков. Считаю своим приятным долгом выразить указанным лицам признательность за оказанную нам помощь.

*Н. Е. Кобринский
Б. А. Трахтенброт*

ВВЕДЕНИЕ

1. В кибернетике автоматами обычно называют разнообразные устройства, предназначенные для переработки информации. Эти устройства имеют конечное число входов, воспринимающих информацию, и конечное число выходов для выдачи обработанной информации. В автоматах дискретного действия, являющихся предметом рассмотрения в настоящей книге, воспринимаемая и выдаваемая информация изображается символами, множество которых конечно (символами, взятыми из конечного алфавита). Физическими носителями символов являются квантованные сигналы.

Последовательности сигналов поступают в квантованной шкале времени на входы автомата. С некоторым запаздыванием Δt , обусловленным физическими свойствами работы автомата, они вызывают на его выходах последовательности ответных сигналов. Целесообразный характер действий автомата достигается тем, что надлежащей его организацией обеспечивается определенная зависимость между возбуждением его входов и возбуждением его выходов (между внешним воздействием и реакцией), которая задается предписанным алгоритмом переработки информации.

В дальнейшем мы будем рассматривать автоматы, реализующие только так называемые *детерминистские* процессы переработки информации. В этом случае последовательности символов, вырабатываемые на выходах автомата, однозначно определяются входными последовательностями. Такие автоматы можно назвать *детерминистскими**) в отличие от *вероятностных* автоматов,

*) Этот термин следует отличать от термина «детерминированный», который будет употребляться в другом смысле.

вырабатывающих случайные последовательности (при фиксированных входных последовательностях). В вероятностных автоматах появление того или иного символа имеет ту или иную вероятность. Имеет смысл сразу же указать три возможных типа автомата, качественно отличающихся друг от друга в функциональном отношении.

В устройствах первого типа набор выходных сигналов, вырабатываемый в момент времени $t + \Delta t$, зависит только от набора входных сигналов, поданных в момент времени t , и не зависит от сигналов, поступивших на входы автомата в предшествующее время. Интервал Δt — время реакции автомата — остается одинаковым для всех t при любых допустимых наборах входных сигналов. Это означает, что возбуждение одних и тех же входов автомата, в какой бы момент времени оно ни произошло, всегда вызывает возбуждение соответствующих им одних и тех же выходов. Такое однозначное и неизменное во времени соответствие между наборами входных и выходных сигналов обуславливается неизменностью внутреннего состояния автомата и независимостью этого состояния от внешнего воздей-

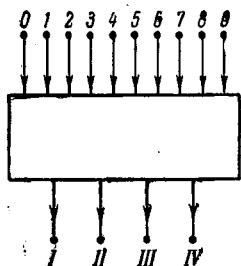


Рис. 0.1

ствия. Устройства описанного типа называют иногда *автоматами без памяти*. Примером подобного устройства может служить шифратор для кодирования десятичных цифр четырехзначными кодовыми группами (рис. 0.1). Зависимость между возбуждениями входов и выходов шифратора имеет вид, указанный в таблице 0.1.

В автоматах второго типа набор выходных сигналов, вырабатываемый в некоторый квантованный момент времени, зависит не только от сигналов, поданных в тот же момент (здесь, как и в дальнейшем изложении, удобно полагать, что $\Delta t = 0$), но и от сигналов, поступивших ранее. Эти предшествующие внешние воздействия (или их фрагменты) фиксируются в автомате путем изменения его внутреннего состояния. Реакция такого детерминистского автомата однозначно определяется поступившим на-

бором входных сигналов и его внутренним состоянием в данный момент времени. Этими же факторами однозначно определяется и то состояние, в которое автомат переходит.

Представляется очевидным, что, поскольку любое физически реализуемое устройство может состоять лишь из конечного числа элементов, оно может пребывать в к о н е ч н о м числе функционально различных состояний,

Таблица 0.1

Вход	I	II	III	IV
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

0—выход невозбужден, 1—выход возбужден.

тракуемом как объем (внутренней) памяти. Таким образом, мы естественно приходим к понятию *конечного автомата* как объекта, имеющего конечное число внутренних состояний и конечное число входов, работа которого носит детерминистский характер в указанном выше смысле.

Простым примером конечного автомата является устройство, преобразующее последовательность единиц, поступающих в квантованной шкале времени на его вход, в некоторую периодическую последовательность, например, вида 100 100 100... Очевидно, что в этом случае автомат должен обладать тремя различными внутренними состояниями.

Если конечный автомат снабдить внешней памятью и постулировать допустимость ее неограниченного расширения, то такая система будет принадлежать к автоматам третьего типа, называемым *машинами Тьюринга*. А. Тьюринг показал, что с помощью автомата этого типа может быть реализован любой алгоритм

переработки информации. Изучение машин Тьюринга представляет большой интерес с точки зрения теории алгоритмов.

В теории автоматов наибольшее внимание уделяется изучению конечных автоматов, поскольку они составляют основу практически реализуемых систем для обработки дискретной информации. Автоматы первого типа могут рассматриваться как частный случай конечных автоматов.

2. Физическая реализация любого автомата обычно осуществляется путем соединения (коммутации) некоторого набора *элементарных компонент*, или *элементов*.

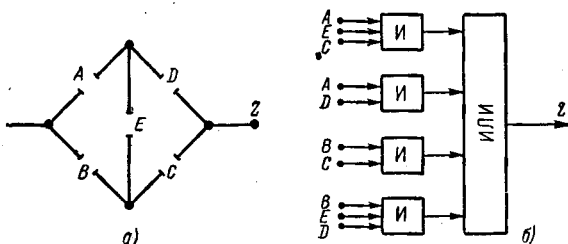


Рис. 0.2.

Понятие *элемента* почти всегда является в известной мере условным, и поэтому целесообразно уточнить, что именно мы будем считать элементами автомата. Это некоторые достаточно простые объекты, каждый из которых, взятый в отдельности, реализует ту или иную функцию. Таким функциональным элементом является, например, пентод, реализующий функцию Шеффера, или двойной катодный повторитель, осуществляющий логическое сложение сигналов, поступающих на его сетки. Работа автомата, построенного из таких элементов, определяется «суммарным действием» каждого из них. Отметим вместе с тем, что элементами можно считать и такие, например, объекты, как сопротивления, конденсаторы, контакты и т.д., которые в указанном выше смысле не являются функциональными. Однако путем их целесообразного соединения могут быть получены автоматы любой сложности, осуществляющие предписанные функциональные действия.

Эти положения лучше всего проиллюстрировать на примерах. На рис. 0.2 показаны две схемы дискретного

действия, реализующие сложные зависимости между возбуждениями входов A, B, C, D, E и выходом Z . Схема на рис. 0.2, *а* построена на релейно-контактных элементах, а на рис. 0.2, *б* — на функциональных. В обеих схемах выход Z возбуждается при подаче на входы по крайней мере одного из следующих наборов сигналов: $\{A, D\}$, $\{B, C\}$, $\{A, E, C\}$, $\{B, E, D\}$.

Отметим сразу же, что при коммутации функциональных элементов с целью получения сложных зависимостей должны выполняться некоторые условия, обеспечивающие

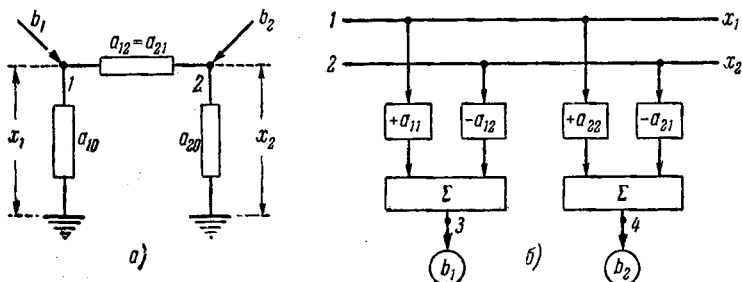


Рис. 0.3.

правильную (в определенном смысле) работу схемы. Так, например, иногда не допускается подключение каждого из входов элемента более чем к одному выходу или образование обратных связей (петель), не содержащих элементов, обеспечивающих временную задержку сигнала, и т. д.

Интересно отметить, что аналогичные ситуации возникают и при построении *автоматов непрерывного действия* *). На рис. 0.3 показаны два типа таких устройств для решения системы из двух линейных алгебраических уравнений:

$$\begin{aligned} a_{11}x_1 - a_{12}x_2 &= b_1, \\ -a_{21}x_1 + a_{22}x_2 &= b_2, \end{aligned}$$

с симметричной матрицей коэффициентов. Устройство, представленное на рис. 0.3, *а*, реализовано с помощью «нефункциональных» элементов — активных проводимостей,

*) См. Н. Е. Кобринский, Математические машины непрерывного действия, Гостехиздат, М., 1954.

величина которых пропорциональна коэффициентам уравнений. Схема на рис. 0.3, б построена на функциональных элементах — коэффициентных и суммирующих. Здесь при коммутации функциональных элементов для обеспечения правильной работы схемы должны соблюдаться условия, аналогичные указанным выше.

3. Схемы конечных автоматов, построенные из функциональных элементов с соблюдением определенных условий их коммутации, мы называем, пользуясь терминологией Беркса и Райта*), *логическими сетями*. Именно такие схемы являются предметом рассмотрения в настоящей книге.

Ограничивая ее содержание рамками конечных автоматов указанного вида, мы исходим из следующих соображений. Исследования и результаты, относящиеся к основным вопросам теории автоматов, в частности к классификации и свойствам операторов переработки информации, способам их задания, оптимизации схемы автомата, носят общий характер и применимы к конечным автоматам, построенным на функциональных и нефункциональных элементах. Поэтому логические сети могут служить удобным модельным объектом для изучения общих закономерностей, возникающих при анализе и синтезе любых конечных автоматов. Вместе с тем вопросы теории релейно-контактных схем достаточно широко освещены в отечественной литературе. В развитии теории этих систем важную роль сыграли исследования, начатые около двадцати лет назад В. И. Шестаковым и К. Шенноном и несколько позднее М. А. Гавриловым. Этими авторами написан ряд обзорных статей и монографий по теории контактных схем. Разработка теории релейно-контактных систем активно продолжается и в настоящее время; решению тех или иных вопросов этой теории посвящены многочисленные работы упомянутых ученых, а также работы Г. Н. Поварова, А. Г. Лунца, С. В. Яблонского, В. Н. Рогинского, А. В. Кузнецова, О. Б. Лупанова, Б. А. Трахтенброта, А. Свободы, Г. Мойсила и др. В последующем,

*) A. W. Burks, G. B. Wright, Theory of logical nets. Proceedings of IRE, v. 41, № 10, 1953.

в связи с развитием электронной автоматик и, в частности, быстродействующих вычислительных машин, появились работы, использующие и развивающие методы теории релейно-контактных схем применительно к логическим сетям, реализуемым на электронных, ферромагнитных и других функциональных элементах. Здесь следует в первую очередь указать работы Г. А й к е н а, Д. Х а ф ф м е н а, Ф. М у р р е я, М. Л. Ц е т л и н а. Целью большинства указанных работ является создание эффективных способов анализа и синтеза систем дискретного действия, базирующихся на методах алгебры логики.

В 1953 г. появилась статья А. Б е р к с а и Д. Р а й т «Теория логических сетей»^{*)}). В ней рассматривается понятие эффективного оператора преобразования информации, реализуемого логической сетью, и исследуются некоторые общие вопросы структуры и организации логических сетей с конечной памятью. Ряд статей, посвященных общим вопросам теории автоматов с конечной памятью, построенных из функциональных элементов, и в частности вопросам анализа, синтеза и надежности их работы, опубликован в сборнике «Автоматы», изданном в 1956 г. под редакцией К. Ш е н н о н а и Д. М а к к а р т и (в том же году он был переиздан у нас на русском языке)^{**)}. В 1956—1960 гг. много статей по различным проблемам теории автоматов появилось в отечественной и зарубежной периодической литературе^{***)}.

^{*)} См. сноску на стр. 14.

^{**)} «Автоматы». Сб. статей под ред. К. Э. Шеннона и Дж. Маккарти, перев. с англ. под ред. А. А. Ляпунова, ИЛ, М., 1956.

^{***)} Таковы, например, работы: A. W. Burks, H. Wang, The logic of automata, Univ. of Michigan, Ann. Arbor, Mich. Eng. Res. Inst., 1956; A. W. Burks, J. M. Copi, The logical design of an idealized generalpurpose computer, J. Franklin Inst., 1956, v. 261, № 3, 4; Gr. C. Moisil, Intrebuinta area logicilor trivalente în teoria mecanismelor automate. Comunicarill, Acad. R. P. R., 1956, t. 6, № 2, 3; Д. Д. Ауфенкамп, Ф. Е. Хон, Анализ последовательностных машин. Периодический сб. перев. иностр. статей «Математика», № 3 : 3, 1959; Д. Д. Ауфенкамп, Анализ последовательностных машин. Периодический сб. перев. иностр. статей «Математика», № 3 : 6, 1959; А. Ш. Блох, О задачах, решаемых последовательностными машинами, Сб. «Проблемы кибернетики», вып. 3, Физматгиз, М., 1960; М. А. Айзерман, Л. А. Гусев,

4. Под *анализом автомата* мы понимаем определение характера преобразования информации, т. е. определение оператора, реализуемого данным автоматом. Проблемы анализа автоматов включают также классификацию операторов и выделение класса операторов, реализуемых логическими сетями, и изучение некоторых важных общих свойств этих операторов. Сюда же относятся и такие практически важные вопросы, как установление алгоритма функционирования конечного автомата, схема и структура которого заданы, выяснение правильности его организации, экономичности его схемы и объема памяти.

При изложении указанных вопросов мы придерживаемся, как это принято в кибернетике, двух аспектов в изучении реальных физических систем для переработки информации; один аспект заключается в рассмотрении способа функционирования автомата *), другой — в рассмотрении его внутренней конструкции **).

При изучении способа функционирования автомата последний, пользуясь кибернетической терминологией, рассматривается как «черный ящик», т. е. как устройство, внутренняя конструкция которого неизвестна. Внимание исследователя в этом случае направлено на изучение характера переработки информации, осуществляемой автоматом, и на выяснение функциональной зависимости между информацией, поступающей на входы устройства и вырабатываемой на его выходах. Сведения, которые могут быть получены при таком «внешнем» наблюдении за «поведением» автомата, носят, естественно, более ограниченный характер, чем те, которые могут

Л. И. Розоноэр, И. М. Смирнова, А. А. Таль, Конечные автоматы. I. «Автоматика и телемеханика», т. XXI, № 2, 1960. Конечные автоматы. II. «Автоматика и телемеханика», т. XXI, № 3, 1960; Они же, О методах реализации конечного автомата, тактность которого определяется изменением состояний входа. «Автоматика и телемеханика», т. XXI, № 12, 1960; Ю. Я. Базилевский, Вопросы теории временных логических функций, Сб. «Вопросы теории математических машин». Физматгиз, М., 1958; В. М. Глушков, Синтез конечных автоматов. Укр. матем. журнал, т. XII, № 2, 1960.

*) Такой подход характерен для так называемой абстрактной теории автоматов.

**) Этот подход характерен для так называемой структурной теории автоматов.

быть получены при анализе его внутренней конструкции. Тем не менее на их основе могут быть выведены некоторые заключения о внутренней структуре автомата и обнаружены некоторые свойства операторов, полезные при решении задач синтеза.

Такого рода результаты были получены независимо друг от друга Э. Ф. Муром*) и Б. А. Трахтенбротом**). Ими установлена связь между объемом памяти автомата и длиной эксперимента (т. е. длиной входной последовательности символов), с помощью которого могут быть обнаружены различные внутренние состояния автомата.

Изучение внутренней структуры автомата позволяет установить принципы его построения и найти взаимосвязь между его конструкцией и функцией, выполняемой автоматом. Это означает, что на основе анализа функциональных свойств элементов и их коммутации в схеме автомата может быть дано формальное описание оператора, реализуемого автоматом.

5. Здесь уместно несколько подробнее коснуться вопроса о способах описания операторов переработки информации. Ясно, что этот вопрос возникает не только в связи с проблемой анализа, но — в еще большей степени — в связи с проблемой синтеза логической сети. Представляется очевидным, что при синтезе автомата прежде всего необходимо каким-то способом описать тот оператор, который должен быть реализован автоматом.

Стандартное описание операторов без памяти осуществляют таблицы, подобные таблице 0.1 (стр. 11.). Если же нужно построить логическую сеть, реализующую оператор с памятью, то обычно полагают, что этот оператор описан посредством канонических уравнений.

Дело, однако, в том, что зачастую то преобразование информации, которое подлежит реализации логической сетью, сообщается в виде предписания, совсем не похожего на ту или иную из упомянутых выше форм, которые можно называть стандартными. Тогда возникает задача перехода

*) Э. Ф. Мур, Умозрительные эксперименты с последовательными машинами, Сб. «Автоматы», ИЛ, М., 1956.

**) Б. А. Трахтенброт, Об операторах, реализуемых в логических сетях, ДАН СССР, т. 112, № 6, 1957.

от первоначального описания оператора к принятой стандартной форме. В тех случаях, когда заведомо известно, что реализуемый оператор является оператором без памяти, описание его в такой, например, стандартной форме, как таблица, не вызывает особых трудностей. Оно сводится (как это имело место в рассмотренном выше примере с шифратором) к указанию для каждого из возможных наборов входных символов соответствующего выходного набора.

Положение существенно усложняется при описании операторов с памятью. В некоторых простейших случаях и здесь оказывается возможным непосредственно указать для любой входной последовательности соответствующую выходную последовательность путем построения ряда таблиц*). Однако в большинстве случаев получение стандартного описания оператора с памятью является задачей далеко не тривиальной; решение этой задачи порою сопряжено с весьма значительными трудностями. Предположим, например, что реализации подлежит оператор поразрядного сложения пары двоичных чисел (многоразрядных). Предписание, сделанное в такой форме, не является стандартным. Перечислить всевозможные комбинации входных последовательностей и указать для них соответствующие выходные последовательности тоже практически невозможно. Для получения стандартного описания нужно провести арифметический анализ процесса суммирования, в результате которого выясняется программа его реализации, основанная на правилах образования суммы одnorазрядных двоичных чисел и учета переноса единицы из низшего разряда в высший. По этой программе может быть дано описание этого оператора в той или иной стандартной форме, например в виде одной таблицы или в виде эквивалентных таблице канонических уравнений оператора.

Приведем еще один пример, который покажет, что для получения стандартного описания оператора может оказаться необходимым достаточно сложный логический ана-

*) Так называемых таблиц включения. См. М. А. Гаврилов, Теория релейно-контактных схем, Изд. АН СССР, М., — Л., 1950.

лиз реализуемого процесса. Предположим, что входными символами являются буквы русского алфавита, разбитые на группы по следующим трем признакам: буква является гласной или согласной, буква является прописной или строчной, буква является типично русской или совпадает по начертанию с какой-либо буквой латинского алфавита. Выходной алфавит двоичный: 0 и 1. Первоначальное описание оператора переработки информации дано в следующем виде: в момент времени t выходной символ должен быть 1, если существовал такой момент $\sigma < t$, когда входным символом была прописная буква, и в каждый момент времени τ в интервале $\sigma \div t$, в котором поданная буква была гласной, был такой момент времени λ в интервале $\sigma \div \tau$, когда входной была типично русская буква. Указанная форма задания оператора не является стандартной. Однако на основе логического анализа описанного процесса требуемый оператор может быть сформулирован в виде формулы исчисления предикатов (см. § 1 и § 5 гл. V). В зависимости от конкретных условий синтеза соответствующей логической сети эти формулы либо непосредственно используются как стандартная форма, либо путем тождественных преобразований приводятся к другим стандартным формам.

Разработке и исследованию различных способов задания операторов с памятью посвящен ряд работ советских и зарубежных ученых, в частности работы С. К. К л и н и и В. М. Г л у ш к о в а (язык регулярных функций), Ю. Т. М е д в е д е в а, М. А. Г а в р и л о в а (таблицы включения), М. Л. Ц е т л и н а (матрицы перехода), Ю. Я. Б а з и л е в с к о г о (язык временных функций), Б. А. Т р а х т е н б р о т а (язык исчисления предикатов), Э. М у р а (диаграммы), Д. Х а ф ф м е н а и др.

6. Представление оператора переработки информации в той или иной стандартной форме позволяет непосредственно перейти к синтезу соответствующей логической сети. Задача синтеза логической сети охватывает круг вопросов, связанных с построением логической сети из указанного набора функциональных элементов, с выяснением физической реализуемости тех или иных классов операторов, с установлением правил построения логических сетей и, наконец, с определением некоторых общих

закономерностей, возникающих при синтезе сетей и оценках их сложности.

7. Как указывалось выше, в системах дискретного действия материальным носителем информации являются квантованные сигналы. В современной дискретной технике для этих целей используются сигналы различной физической природы, причем обычно имеют дело с двумя фиксированными значениями сигналов, которым сопоставляют символы двоичного алфавита (чаще всего 0 и 1). Физические элементы, применяемые в качестве функциональных элементов логической сети, также приспособлены к работе с «двоичными» сигналами; их входные и выходные каналы способны пребывать в двух функционально различных состояниях. В связи с этим при синтезе логической сети входные и выходные символы кодируются в двоичном коде и при решении задач синтеза широко пользуются символическим аппаратом двузначной логики.

8. При синтезе автомата и его практической реализации приходится обычно принимать во внимание различные соображения. Некоторые из них диктуются конкретными условиями проектирования и эксплуатации устройства (такими, например, как размеры устройства, его стоимость, внешние условия работы и т. п.), другие же носят более общий характер и поэтому поддаются достаточно общей математической трактовке; к последним относятся вопросы экономичного построения (минимизации) логической сети и функциональной надежности ее действия.

Большинство из упомянутых выше работ в области синтеза логических сетей посвящено разработке различных практических приемов синтеза с целью получения по возможности более экономной сети (имеющей минимальное число элементов). Вместе с тем в ряде работ исследовались вопросы синтеза схем дискретного действия, связанные с установлением общих закономерностей, возникающих при их реализации, и получением асимптотических оценок числа необходимых элементов. Применительно к схемной реализации операторов без памяти эти исследования были начаты К. Шенноном*) и получили

*) C. E. Shannon, The synthesis of two-terminal switching circuits, BSTJ, v. 28, № 1, 1949, p. 56—98.

блестящее завершение в работах О. Б. Лупанова *). В дальнейшем родственные закономерности были установлены С. В. Яблонским **) и другими авторами для ряда классов операторов, допускающих сравнительно простую схемную реализацию, а Б. А. Трахтенбротом ***)—для операторов с памятью.

Важное теоретическое и практическое значение приобретает в настоящее время проблема функциональной надежности логической сети, требующая изыскания методов синтеза, обеспечивающих заданный уровень вероятности правильной работы автомата при заданной надежности его элементов. Эта задача — и соответствующие методы ее решения — носит статистический характер; она впервые была сформулирована и исследована в работах Дж. Неймана ****).

9. В теории логических сетей вопросы анализа, синтеза и функциональной надежности излагаются при известной *идеализации* условий работы реальных устройств. Полагают, что сигналы, несущие информацию, имеют дискретные, фиксированные уровни; мощность сигналов при этом не учитывается. Непрерывно текущее время заменяется дискретно-изменяющимся; при этом считают, что изменение уровней сигналов происходит в дискретных фиксированных интервалах времени. Ряд допущений принимается при рассмотрении физических элементов сети. Так, пренебрегают неизбежной инерцией элементов, если возникающее в связи с этим запаздывание в их срабатывании мало сравнительно с тактом работы логической сети. Если это запаздывание соизмеримо с тактом, то оно реализуется в логической сети отдельным элементом задержки. По мере повышения быстродействия автоматов и

*) См. О. Б. Лупанов, Об одном методе синтеза схем, Изв. высш. учеб. заведений, Радиофизика, № 1, 1958, стр. 120—140.

**) См. С. В. Яблонский, Об алгоритмических трудностях синтеза минимальных контактных схем. Сб. «Проблемы кибернетики», вып. 2, Физматгиз, М., 1959, стр. 75—122.

***) См. Б. А. Трахтенброт, Асимптотическая оценка сложности логических сетей с памятью, ДАН СССР, т. 127, № 2, 1959, стр. 281—284.

****) См. Дж. Нейман, Вероятностная логика и синтез надежных организмов из ненадежных компонент, Сб. «Автоматы», ИЛ, М., 1956.

усложнения реализуемых ими процессов влияние указанных факторов на функционирование логической сети, и особенно на надежность ее работы, становится все более существенным. В связи с этим важное значение приобретает разработка вопросов теории «реальных» логических сетей, условия работы которых приближаются к действительным условиям работы реальных автоматических устройств. Некоторые попытки в этом направлении предприняты в работе М. Л. Цетлина и Л. М. Шехтмана *). Большой интерес для разработки этой проблематики представляют проводящиеся С. В. Яблонским исследования по применению аппарата многозначных логик для анализа и синтеза электронных схем.

В настоящей книге охвачен круг вопросов, относящихся лишь к некоторым из упомянутых выше аспектов теории конечных автоматов.

*) М. Л. Цетлин и Л. М. Шехтман, Двухтактные ферротранзисторные схемы и алгебраический метод их синтеза, Сб. «Проблемы кибернетики», № 2, Физматгиз, М., 1959, стр. 139—179.

ГЛАВА I

ЭЛЕМЕНТЫ ЛОГИКИ

§ 1. Общие замечания *)

Как указывалось во введении, материальным носителем информации в системах дискретного действия обычно являются квантованные сигналы, имеющие два фиксированных значения. Физическая реализация сигнала может быть различной; он может быть представлен импульсом напряжения, следующим по рассматриваемому каналу, или уровнем напряжения в этом канале. Возможна реализация сигнала посредством перемещения звена механической цепи и т. п. В соответствии с этим каналы и элементы логической сети приспособлены к восприятию и передаче двух уровней сигнала.

Двум попарно различным значениям сигнала, несущего дискретную информацию, сопоставляются попарно различные символы, чаще всего обозначаемые 0 и 1. Обычно символ 1 обозначает наличие сигнала, символ 0 — отсутствие сигнала. Теми же символами пользуются для обозначения и двух функционально различных состояний каналов сети. Промежуточные состояния канала, возникающие при его переходе из одного расчетного состояния в другое, при рассмотрении идеализированных условий работы логической сети во внимание не принимаются.

В связи с изложенными обстоятельствами при физической реализации процесса переработки дискретной

*) При изложении § 1, 2 и 3 мы пользовались в основном работой С. В. Яблонского «Функциональные построения в k -значной логике» (Труды Матем. ин-та им. В. А. Стеклова, т. 51, Изд. АН СССР, М., 1958, стр. 5—142).

информации возникает необходимость ее двоичного кодирования, т. е. представления букв употребляемых алфавитов наборами двоичных символов. При этом попарно различным буквам алфавита должны сопоставляться попарно различные наборы двоичных знаков. В таблице 0.1 (см. стр. 11) десятичным цифрам сопоставлены наборы из четырех двоичных знаков, удовлетворяющие указанному условию. Наборы двоичных знаков $\langle x_1 x_2 \dots x_m \rangle$ ($x_j = 0, 1$), используемые для кодирования букв заданного алфавита, часто называют *кодowymi группами*. Очевидно, что при m двоичных знаках число различных наборов равно 2^m (где m — целое положительное число). Так, например, при $m=3$ мы имеем следующие восемь наборов: $\langle 000 \rangle$, $\langle 001 \rangle$, $\langle 010 \rangle$, $\langle 011 \rangle$, $\langle 100 \rangle$, $\langle 101 \rangle$, $\langle 110 \rangle$ и $\langle 111 \rangle$.

Мы будем рассматривать множество S всех функций $f(x_1, x_2, \dots, x_m)$ таких, что их значения равны 0 или 1 и определены для всевозможных наборов двоичных аргументов. Эти функции называют *функциями алгебры логики*. Для того чтобы задать функцию алгебры логики от m переменных, нужно указать ее значения для каждого из 2^m наборов значений аргументов, например, с помощью таблицы вида I.1.

Таблица I.1

x_1	x_2	x_3	...	x_{m-1}	x_m	$f(x_1, x_2, \dots, x_{m-1}, x_m)$
0	0	0	...	0	0	$f(0, 0, 0, \dots, 0, 0)$
0	0	0	...	0	1	$f(0, 0, 0, \dots, 0, 1)$
...
...
...
1	1	1	...	1	0	$f(1, 1, 1, \dots, 1, 0)$
1	1	1	...	1	1	$f(1, 1, 1, \dots, 1, 1)$

Для удобства наборы аргументов расположены в таблице так, что они соответствуют числам $0, 1, 2, 3, \dots, 2^m - 1$, записанным в двоичной системе счисления.

В приведенной таблице каждому набору аргументов может быть сопоставлено два значения функции: 0 или 1. Отсюда следует, что число всех (различных) функций алгебры логики, зависящих от m аргументов, равно $2^{(2^m)}$.

Здесь полезно подчеркнуть, что число функций алгебры логики очень быстро возрастает с увеличением m . Вместе с тем при большом числе аргументов (уже при $m > 5$) таблица вида I.1, задающая функцию, весьма усложняется. В этих случаях целесообразно пользоваться для задания функции алгебры логики таблицами другого вида (см. ниже, § 3).

В алгебре логики символы 1 и 0 отождествляются, соответственно, с понятиями истинного суждения и ложного суждения, а функциональная таблица вида I.1 — с таблицей, определяющей зависимость истинности и ложности сложного суждения $f(x_1, x_2, \dots, x_m)$ от истинности и ложности составляющих суждений x_i (где $i = 1, 2, \dots, m$).

В связи со сказанным выше целесообразно выделить из множества функций алгебры логики ряд «элементарных» функций, имеющих простую логическую интерпретацию и обычно используемых в качестве логических связей (операций) для построения более сложных суждений. К таким элементарным функциям относятся:

I. \bar{x} — дополнение к x , или отрицание x ; читается «не x ». Суждение \bar{x} ложно (равно нулю), если суждение x истинно (равно 1); суждение \bar{x} истинно, если суждение x ложно.

II. $x_1 \cdot x_2$ (другое обозначение: $x_1 \& x_2$ *) — конъюнкция, или логическое умножение; читается « x_1 и x_2 ». Эта функция выражает суждение, которое истинно в том и только в том случае, когда как суждение x_1 , так и суждение x_2 истинны.

III. $x_1 \vee x_2$ — дизъюнкция, или логическое сложение; читается « x_1 или x_2 ». Эта функция выражает суждение, которое истинно в том и только в том случае, если по крайней мере одно из двух суждений x_1 и x_2 является истинным.

IV. $x_1 \rightarrow x_2$ — импликация; читается «если x_1 , то x_2 » или «из x_1 следует x_2 ». Эта функция выражает суждение, которое ложно в том и только в том случае, когда x_1 истинно, а x_2 ложно.

*) В дальнейшем для указания этой операции мы будем пользоваться как одним, так и другим обозначением. Точку в большинстве случаев будем опускать.

V. $x_1 \equiv x_2$ — эквиваленция; читается « x_1 эквивалентно x_2 », « x_1 равнозначно x_2 », « x_1 , если и только если x_2 ». Эта функция выражает суждение, которое тогда и только тогда истинно, когда суждения x_1 и x_2 или оба истинны, или оба ложны.

Ниже приведены таблицы, соответствующие этим «элементарным» функциям.

Т а б л и ц а I.2

x	\bar{x}
0	1
1	0

Т а б л и ц а I.3

x_1	x_2	$x_1 \cdot x_2$	$x_1 \vee x_2$	$x_2 \rightarrow x_1$	$x_1 \equiv x_2$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Указанные выше функции позволяют ввести язык формул для построения сложных суждений, выражаемых в разговорном языке посредством соединения простых суждений с помощью слов «или», «и», «если..., то...» и т. п., и введения в суждения частицы «не», поскольку, как будет показано ниже, любая сложная функция алгебры логики может быть представлена в виде суперпозиции перечисленных «элементарных» функций.

Пример. Напишем в виде формулы следующее сложное суждение (см. введение, стр. 12, рис. 0.2): «выход Z возбуждается, если и только если возбуждены входы A и D , или входы B и C , или входы A , E и C , или входы B , E и D », которое можно считать составленным из простых суждений вида «возбужден вход A », «возбужден вход B » и т. д. и слов «или», «и» и «если и только если».

Пользуясь «элементарными» функциями, наше суждение можно представить в следующем виде:

$$Z \equiv (A \cdot D \vee B \cdot C \vee A \cdot E \cdot C \vee B \cdot E \cdot D).$$

Функциональные построения, связанные с операциями над суждениями, являются предметом изучения *логики суждений*, или *алгебры логики*, являющейся одним из разделов *математической логики*.

Однако язык логики суждений недостаточен для формального описания более сложных логических связей, выражающих, например, соотношения между субъектом и предикатом суждения. Во введении было дано словесное описание оператора, перерабатывающего последовательность букв русского алфавита в последовательность двоичных знаков по некоторому предписанному алгоритму. Для формального описания такого оператора в терминах математической логики возникает необходимость привлечения языка *логики предикатов*.

В этой главе даются краткие сведения по алгебре логики и логике предикатов, необходимые для изложения вопросов анализа и синтеза логических сетей. Во втором и третьем параграфах излагаются основы алгебры логики и приводятся некоторые распространенные способы упрощения формул алгебры логики. Мы сочли целесообразным посвятить отдельный параграф (§ 4) рассмотрению общей задачи разложения функции алгебры логики, решение которой дано О. Б. Лупановым*). Эти результаты используются при изложении вопросов оптимизации сети. Наконец, в последнем параграфе даются краткие сведения по логике предикатов.

§ 2. Элементы алгебры логики

Изложение элементов алгебры логики естественно начать с определения формулы логики суждений (формулы алгебры логики). Это определение носит индуктивный характер и имеет следующий вид:

1. Малые латинские буквы (включая буквы с индексами): $p, q, r, \dots, x, y, z, p_1, q_2, \dots, x_1, x_2, \dots, z_1, z_2, \dots$,

*) О. Б. Лупанов, Об одном методе синтеза схем. Изв. высш. учебн. заведений, Радиофизика, № 1, 1958, стр. 120—140.

(называемые *переменными для суждений* или *пропозициональными переменными*), а также константы 0 и 1 (константы ложного и истинного суждений) суть формулы.

2. (а) Если некоторое выражение \mathcal{A} есть формула, то $(\overline{\mathcal{A}})$ тоже формула.

(б) Если некоторые выражения \mathcal{A} и \mathcal{B} суть формулы, то выражения

$$(\mathcal{A} \vee \mathcal{B}),$$

$$(\mathcal{A} \& \mathcal{B}),$$

$$(\mathcal{A} \rightarrow \mathcal{B}),$$

$$(\mathcal{A} \equiv \mathcal{B})$$

тоже формулы.

3. Никакие другие выражения, кроме тех, которые являются формулами на основании пунктов 1 и 2, не суть формулы.

Пример. Выражение

$$(((x \& y) \rightarrow (x \vee y)) \vee z)$$

является формулой. Это видно из того, что x, y — формулы (пункт 1 определения формулы), в силу чего $(x \& y)$ и $(x \vee y)$ — тоже формулы (пункт 2(б) определения формулы), что обуславливает то, что и $((x \& y) \rightarrow (x \vee y))$ является формулой (пункт 2(б)); а раз $((x \& y) \rightarrow (x \vee y))$ — формула и z (в соответствии с пунктами 1) — формула, то и $((x \& y) \rightarrow (x \vee y)) \vee z$ тоже формула.

В связи с приведенным определением формулы логики суждений сделаем следующие замечания. Большие готические буквы употребляются для разговора о формулах, но сами формулами не являются. Вместо знака «&» мы будем часто употреблять точку «·» (как отмечено в § 1, точка может и опускаться); таким образом, если \mathcal{A} и \mathcal{B} — формулы логики суждений, то $(\mathcal{A} \cdot \mathcal{B})$ — тоже формула логики суждений. Договоримся, что в одной и той же формуле знаки «&» и «·» не должны встречаться.

В целях экономии скобок в формулах примем следующие соглашения. Внешние скобки в формуле могут опускаться. Если формула имеет вид отрицания, т. е. вид $\overline{\mathcal{A}}$, то внешние скобки подформулы \mathcal{A} могут опускаться. Наряду с круглыми скобками при написании формул могут использоваться квадратные и фигурные скобки.

Чтобы еще более сократить число скобок в формулах, условимся, что знаки конъюнкции «&» или дизъюнкции « \vee » связывают более тесно, чем знаки импликации или эквиваленции, так что если имеются, например, формулы

$$((x \& y) \rightarrow z), \quad (p \equiv (q \vee \bar{r})), \quad ((x \& y) \rightarrow (x \vee y)),$$

то их можно записать в виде

$$(x \& y \rightarrow z), \quad (p \equiv q \vee \bar{r}), \quad (x \& y \rightarrow x \vee y),$$

или, опуская внешние скобки:

$$x \& y \rightarrow z, \quad p \equiv q \vee \bar{r}, \quad x \& y \rightarrow x \vee y.$$

Условимся, наконец, что если в качестве знака конъюнкции используется точка, то конъюнкция связывает более тесно, чем дизъюнкция. В силу этого соглашения формулы

$$(x \& \bar{x}) \vee z, \quad (x \& y) \vee (x \& z (\bar{x} \vee y))$$

мы можем переписать в виде

$$\bar{x} \vee z, \quad xy \vee xz (\bar{x} \vee y).$$

Каждая формула алгебры логики задает некоторую функцию алгебры логики. При этом различные формулы могут задавать одну и ту же функцию. Две формулы \mathfrak{A} и \mathfrak{B} называются *эквивалентными*, если они задают одну и ту же функцию алгебры логики. Чтобы проверить, являются ли формулы \mathfrak{A} и \mathfrak{B} эквивалентными, достаточно построить таблицы, соответствующие функциям, задаваемым каждой из этих формул (такие таблицы принципиально всегда могут быть построены, коль скоро нам известен вид формул); если распределение знаков 0 и 1 в последних колонках обеих таблиц совпадает, то формулы \mathfrak{A} и \mathfrak{B} эквивалентны. Для указания эквивалентности формул мы будем использовать знак «экв.». Запись « \mathfrak{A} экв. \mathfrak{B} » означает, что \mathfrak{A} и \mathfrak{B} эквивалентны. Вместо знака «экв.» мы будем часто употреблять обычный знак равенства.

П р и м е р ы. Доказать эквивалентность формул

$$(x_1 \& x_2) \& x_3 \text{ и } x_1 \& (x_2 \& x_3).$$

Строим таблицы, соответствующие обоим формулам (табл. I.4; эта таблица содержит фактически две таблицы:

одну — для формулы $(x_1 \& x_2) \& x_3$ и другую — для формулы $x_1 \& (x_2 \& x_3)$.

Таблица I.4

x_1	x_2	x_3	$x_1 \& x_2$	$(x_1 \& x_2) \& x_3$	$x_2 \& x_3$	$x_1 \& (x_2 \& x_3)$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

Поскольку в таблице I.4 распределение знаков «0» и «1» в колонках для $(x_1 \& x_2) \& x_3$ и $x_1 \& (x_2 \& x_3)$ совпадает, эти формулы эквивалентны. Таким образом,

$$(x_1 \& x_2) \& x_3 \text{ экв. } x_1 \& (x_2 \& x_3)$$

или, что означает то же самое,

$$(x_1 \& x_2) \& x_3 = x_1 \& (x_2 \& x_3).$$

Проверим теперь, являются ли эквивалентными следующие формулы:

$$\bar{x} \vee y \text{ и } \overline{x \& y}.$$

Строим соответствующие им таблицы (см. табл. I.5).

Таблица I.5

x	y	\bar{x}	$\bar{x} \vee y$	$x \& y$	$\overline{x \& y}$
0	0	1	1	0	1
0	1	1	1	0	1
1	0	0	0	0	1
1	1	0	1	1	0

Так как в таблице I.5 в колонках для $\overline{x \vee y}$ и $\overline{x \& y}$ распределение знаков «0» и «1» различно, эти формулы не эквивалентны.

Применение табличного представления функций алгебры логики *) позволяет установить следующие законы, относящиеся к свойствам элементарных логических операций.

1. Операции конъюнкции, дизъюнкции и эквиваленции обладают свойством *ассоциативности*, т. е.:

$$\begin{aligned}(x_1 \& x_2) \& x_3 &= x_1 \& (x_2 \& x_3), \\ (x_1 \vee x_2) \vee x_3 &= x_1 \vee (x_2 \vee x_3), \\ (x_1 \equiv x_2) \equiv x_3 &= x_1 \equiv (x_2 \equiv x_3).\end{aligned}$$

Это дает право писать формулы, содержащие только знаки конъюнкции (соответственно, только знаки дизъюнкции, только знаки эквиваленции) без скобок:

$$\begin{aligned}(x_1 \& x_2) \& x_3 &= x_1 \& (x_2 \& x_3) = x_1 \& x_2 \& x_3, \\ (x_1 \vee x_2) \vee x_3 &= x_1 \vee (x_2 \vee x_3) = x_1 \vee x_2 \vee x_3, \\ (x_1 \equiv x_2) \equiv x_3 &= x_1 \equiv (x_2 \equiv x_3) = x_1 \equiv x_2 \equiv x_3.\end{aligned}$$

2. Операции конъюнкции, дизъюнкции и эквиваленции обладают свойством *коммутативности*:

$$\begin{aligned}x_1 \& x_2 &= x_2 \& x_1, \\ x_1 \vee x_2 &= x_2 \vee x_1, \\ x_1 \equiv x_2 &= x_2 \equiv x_1.\end{aligned}$$

3. Для конъюнкции и дизъюнкции имеют силу, далее, следующие законы:

$$x \& x = x, \quad x \vee x = x.$$

Законы, выражаемые этими соотношениями, называются

*) Говоря о функциях алгебры логики, мы будем часто пользоваться выражениями вида $f(x, y)$ («функция f алгебры логики от переменных x и y »), $g(x_1, x_2, \dots, x_m)$ («функция g алгебры логики от переменных x_1, x_2, \dots, x_m ») и т. п. Когда будет говориться о некоторой функции алгебры логики, ее можно представлять себе заданной таблично или определенной некоторой формулой логики суждений.

законами *идемпотентности* для конъюнкции и дизъюнкции.

$$4. \quad \begin{aligned} (x_1 \vee x_2) \& x_3 &= (x_1 \& x_3) \vee (x_2 \& x_3), \\ (x_1 \& x_2) \vee x_3 &= (x_1 \vee x_3) \& (x_2 \vee x_3). \end{aligned}$$

Первое из этих равенств выражает закон *дистрибутивности* (распределительности) конъюнкции относительно дизъюнкции. второе — закон *дистрибутивности* дизъюнкции относительно конъюнкции. Используя точку « \cdot » вместо знака « $\&$ », мы можем переписать соотношения 4 в виде

$$\begin{aligned} (x_1 \vee x_2) \cdot x_3 &= x_1 \cdot x_3 \vee x_2 \cdot x_3, \\ x_1 \cdot x_2 \vee x_3 &= (x_1 \vee x_3) \cdot (x_2 \vee x_3). \end{aligned}$$

5. Операции конъюнкции и дизъюнкции обладают также следующими свойствами:

$$\begin{aligned} x \& 0 &= 0, \\ x \& 1 &= x, \\ x \vee 0 &= x, \\ x \vee 1 &= 1. \end{aligned}$$

Напомним, что 0 логически интерпретируется как ложное суждение, а 1 — как истинное суждение. Поэтому эти соотношения можно истолковывать (соответственно) как «конъюнкция x с ложным суждением порождает ложное суждение», «конъюнкция x с истинным суждением равнозначна x (т. е. конъюнктивное добавление 1 ничего не меняет)», «дизъюнкция x и ложного суждения равнозначна x (т. е. дизъюнктивное присоединение ложного суждения ничего не меняет)», «дизъюнкция x и истинного суждения порождает истинное суждение».

6. Наконец, имеют место соотношения:

$$\begin{aligned} x \& \bar{x} &= 0, \\ x \vee \bar{x} &= 1, \\ \bar{\bar{x}} &= x. \end{aligned}$$

Первое из этих равенств выражает логическое противоречие: «суждение x в конъюнкции со своим отрицанием

\bar{x} образуют ложное суждение $x \& \bar{x}$. Второе равенство выражает так называемый закон исключенного третьего и может быть прочитано следующим образом: «дизъюнкция суждения x и его отрицания \bar{x} (т. е. суждение « x или не x ») образует истинное суждение». Третье равенство выражает равнозначность двойного отрицания (т. е. отрицания отрицания) данного суждения самому этому суждению.

7. Введем для аргумента x_j и его отрицания обозначение $x_j^{\sigma_j}$, где σ_j равно 0 или 1, причем условимся, что

$$x^0 = \bar{x}, \quad x^1 = x.$$

Произвольная конъюнкция

$$x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_h^{\sigma_h}$$

равна единице, если и только если $x_1 = \sigma_1, x_2 = \sigma_2, \dots, x_h = \sigma_h$. Во всех остальных случаях (т. е. когда по крайней мере один из членов x_j не равен σ_j) эта конъюнкция равна нулю. Действительно, согласно свойству 5, например, конъюнкция $x_1 x_2 x_3 x_4$ (в которой все $\sigma_j = 1$) равна единице, если все $x_j = 1$. Конъюнкция $\bar{x}_1 x_2 x_3 x_4$ (в которой $\sigma_1 = \sigma_4 = 0$, а $\sigma_2 = \sigma_3 = 1$) равна единице только в случае, когда $x_1 = x_4 = 0$, а $x_2 = x_3 = 1$.

Подчеркнем, что законы 1—7 остаются справедливыми при замене фигурирующих в них переменных x, x_1, x_2, \dots любыми формулами алгебры логики.

Т е о р е м а 1. *Каждую функцию алгебры логики*

$$f(x_1, x_2, x_3, \dots, x_k, x_{k+1}, \dots, x_m) \quad (m \geq 1)$$

можно представить в виде следующей формулы:

$$\bigvee_{\sigma_1, \dots, \sigma_k} x_1^{\sigma_1} x_2^{\sigma_2} \dots x_k^{\sigma_k} f(\sigma_1, \sigma_2, \dots, \sigma_k, x_{k+1}, \dots, x_m), \quad (1.1)$$

где символ $\bigvee_{\sigma_1, \dots, \sigma_k}$ означает логическую сумму по всем $\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_k$.

Такое представление функции называется *разложением функции по k переменным*. Чтобы убедиться в правильности этой теоремы, рассмотрим произвольные

$\sigma_1, \sigma_2, \dots, \sigma_k$. Пусть $x_1 = \sigma_1, x_2 = \sigma_2, \dots, x_k = \sigma_k$. Тогда в левой части (1.1) получим $f(\sigma_1, \sigma_2, \dots, \sigma_k, x_{k+1}, \dots, x_m)$, а правая часть того же соотношения в силу свойств 3, 5 и 7 дает также $f(\sigma_1, \sigma_2, \dots, \sigma_k, x_{k+1}, \dots, x_m)$. Тем самым теорема доказана.

Пример. Разложить функцию $f(x_1, x_2, x_3)$ по переменным x_1 и x_2 . В соответствии с (1.1) получим:

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 f(0, 0, x_3) \vee \bar{x}_1 x_2 f(0, 1, x_3) \vee x_1 \bar{x}_2 f(1, 0, x_3) \vee x_1 x_2 f(1, 1, x_3).$$

Нетрудно убедиться в справедливости этого равенства. Действительно, для любых произвольно выбранных σ_j его правая и левая части совпадают. Пусть, например, $\sigma_1 = 0$, а $\sigma_2 = 1$. В левой части мы получим $f(0, 1, x_3)$, а в правой части (являющейся разложением функции $f(x_1, x_2, x_3)$ по переменным x_1 и x_2) все члены, кроме второго, обращаются в нуль. В силу свойств 7 и 5 второй член получает то же значение $f(0, 1, x_3)$.

Из теоремы 1 вытекает:

Теорема 2. Каждая функция алгебры логики может быть выражена через конъюнкцию, дизъюнкцию и отрицание.

Действительно, в силу предыдущей теоремы

$$f(x_1, x_2, \dots, x_m) = \bigvee_{\sigma_1, \dots, \sigma_m} x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m} f(\sigma_1, \sigma_2, \dots, \sigma_m). \quad (1.2)$$

Так как $f(\sigma_1, \sigma_2, \dots, \sigma_m)$ равна либо 0, либо 1, то в силу свойства 5 получаем:

$$f(x_1, x_2, \dots, x_m) = \bigvee_{f(\sigma_1, \sigma_2, \dots, \sigma_m)=1} x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m}, \quad (1.3)$$

причем, как видно из самой записи (1.3), логическое суммирование в данном случае ведется по тем наборам $\sigma_1, \sigma_2, \dots, \sigma_m$, для которых $f(\sigma_1, \sigma_2, \dots, \sigma_m) = 1$. Правая часть равенства (1.3) носит название *совершенной дизъюнктивной нормальной формы* (с.д.н.ф.) функции $f(x_1, x_2, \dots, x_m)$. Равенство (1.3) определяет способ построения совершенной дизъюнктивной нормальной формы функции, заданной

таблично. Этот способ состоит в следующем. Из таблицы отбираются все наборы $(\sigma_1, \dots, \sigma_m)$, для которых функция $f(x_1, \dots, x_m)$ равна 1. Далее, для каждого из таких наборов составляются конъюнкции $(x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_m^{\sigma_m})$, и затем все эти конъюнкции соединяются знаком дизъюнкции.

П р и м е р. Составить совершенную дизъюнктивную нормальную форму для функции $f(x_1, x_2, x_3)$, заданной таблицей I.6.

Т а б л и ц а I.6

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Р е ш е н и е. В соответствии с указанным правилом отбираем из таблицы наборы, расположенные во 2-й, 3-й, 5-й и 8-й строках (сверху) и строим по ним с.д.н.ф. функции $f(x_1, x_2, x_3)$:

$$f(x_1, x_2, x_3) = \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \vee \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \cdot x_3.$$

Члены вида $x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot x_3^{\sigma_3} \cdot \dots \cdot x_m^{\sigma_m}$, входящие в с.д.н.ф. некоторой функции алгебры логики, называются *дизъюнктивными членами* данной с.д.н.ф. В результате приведения функций алгебры логики к с.д.н.ф. они предстают в виде логических сумм, состоящих из некоторого числа дизъюнктивных членов каждая (включая «вырожденные» логические суммы, состоящие каждая только из одного члена). Представление функций алгебры логики в с.д.н.ф. возможно для всех функций, исключая функции, тождественно равные 0 (такие функции соответствуют тождественно ложным суждениям логики суждений, т. е. суждениям, принимающим значение «ложь» при любых значениях входящих в них составляющих суждений).

Можно вместе с тем показать, что любая функция алгебры логики представима в виде логического произведения (конъюнкции) некоторого числа членов вида $(x_1^{\sigma_1} \vee x_2^{\sigma_2} \vee \dots \vee x_m^{\sigma_m})$ (включая «вырожденные» случаи, когда логическое произведение состоит только из одного члена данного вида). Такая форма представления функции алгебры логики называется совершенной конъюнктивной нормальной формой (с.к.н.ф.) данной функции; члены вида $(x_1^{\sigma_1} \vee x_2^{\sigma_2} \vee \dots \vee x_m^{\sigma_m})$ носят название *конъюнктивных членов* с.к.н.ф. Представление функции алгебры логики в виде с.к.н.ф. возможно для всех функций алгебры логики, исключая функции, тождественно равные 1.

Из изложенного следует, что каждую функцию алгебры логики *) можно представить в любой из двух нормальных форм. При аналитическом представлении функции алгебры логики, заданной посредством таблицы, обычно пользуются дизъюнктивной совершенной нормальной формой, имеющей преимущество более легкой обозримости.

Для упрощения дальнейшего изложения установим взаимно-однозначное соответствие между наборами $(\sigma_1, \sigma_2, \dots, \sigma_m)$ из нулей и единиц и целыми числами g ($0 \leq g \leq 2^m$). Числу g соответствует набор, являющийся двоичной записью g , т. е. $g = \sum_{i=1}^{i=m} \sigma_i 2^{m-i}$. Число g , отвечающее набору $(\sigma_1, \sigma_2, \dots, \sigma_m)$, будем обозначать $g(\sigma_1, \sigma_2, \dots, \sigma_m)$, а набор, отвечающий числу g , обозначим $[\sigma_1(g), \sigma_2(g), \dots, \sigma_m(g)]$. Так, например, при $m=4$ числу 13 отвечает набор (1101), а набор (0101) соответствует числу 5.

При такой записи каждому дизъюнктивному члену (некоторой дизъюнктивной совершенной нормальной формы) и, аналогично, каждому конъюнктивному члену (некоторой конъюнктивной совершенной нормальной формы) от m переменных

$$P_j = x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m} \quad \text{и} \quad C_j = x_1^{\sigma_1} \vee x_2^{\sigma_2} \vee \dots \vee x_m^{\sigma_m}$$

присваивается номер $j(\sigma_1, \sigma_2, \dots, \sigma_m)$, т. е. номер, двоичной записью которого является набор $\sigma_1, \sigma_2, \dots, \sigma_m$ ($0 \leq j < 2^m$),

*) С отмеченными выше оговорками, касающимися функций, тождественно равных 1, и функций, тождественно равных 0.

и, наоборот, членам с номерами s соответствуют наборы $[\sigma_1(s), \sigma_2(s), \dots, \sigma_m(s)]$, где $0 \leq s < 2^m$.

Таким образом, например, при $m=3$ мы имеем:

$$P_0 = \bar{x}_1 \bar{x}_2 \bar{x}_3, P_1 = \bar{x}_1 \bar{x}_2 x_3, \dots, P_7 = x_1 x_2 x_3;$$

$$C_0 = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3, C_1 = \bar{x}_1 \vee \bar{x}_2 \vee x_3, \dots, C_7 = x_1 \vee x_2 \vee x_3.$$

Для всякой функции алгебры логики z , выраженной через отрицание, дизъюнкцию и конъюнкцию, отрицание этой формулы \bar{z} может быть получено путем замены аргументов их отрицаниями и обмена местами символов конъюнкции и дизъюнкции (теорема де Моргана^{*)}).

Пусть, например^{**}),

$$A = (x_1 \vee \bar{x}_2) \cdot (x_3 \vee x_4 \vee x_5) \cdot \bar{x}_6.$$

Тогда

$$\bar{A} = \bar{x}_1 \cdot x_2 \vee \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 \vee x_6.$$

Следствиями теоремы де Моргана являются, в частности, следующие соотношения:

$$\overline{x_1 \cdot x_2} = \bar{x}_1 \vee \bar{x}_2,$$

$$\overline{x_1 \vee x_2} = \bar{x}_1 \cdot \bar{x}_2.$$

Из теоремы де Моргана следует также, что отрицанием дизъюнктивного члена является некоторый конъюнктивный член, и наоборот:

$$\bar{P}_k = C_j; \quad \bar{C}_k = P_j,$$

причем j — это число, двоичная запись которого образуется из двоичной записи числа k путем замены единиц на

^{*)} Де Морган (Augustus de Morgan) (1806—1878) — английский логик и математик, один из основоположников математической логики.

^{**)} Знак «=» означает «есть по определению», «равно по определению».

нули и наоборот. Так, например, если

$$P_k = x_1 \cdot \bar{x}_2 \cdot x_3 = P_5 \quad (k = 101),$$

то

$$C_j = \bar{x}_1 \vee x_2 \vee \bar{x}_3 = C_2 \quad (j = 010).$$

Обратимся теперь к правилам, определяющим образование всегда истинных и всегда ложных выражений алгебры логики.

А) Логическое произведение двух разных дизъюнктивных членов, содержащих m переменных, равно нулю. Отметим, что любые два дизъюнктивных члена, содержащих m переменных, таковы, что один из членов включает отрицание по крайней мере одной какой-либо из переменных, входящих в другой дизъюнктивный член. Поэтому (в частности, на основании свойств 5 и 6) их произведение равно 0. Пусть, например, $m=3$. Тогда

$$(x_1 \cdot x_2 \cdot \bar{x}_3) \cdot (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3) = 0.$$

Б) Логическая сумма двух разных конъюнктивных членов, содержащих m переменных, равна единице. Действительно, два конъюнктивных члена, содержащих m переменных, обладают тем же свойством, которое в пункте А) было отмечено для конъюнктивных членов; поэтому (в частности, в силу свойств 5 и 6) их сумма равна 1. Например, при $m=3$ мы имеем:

$$(x_1 \vee \bar{x}_2 \vee x_3) \vee (\bar{x}_1 \vee \bar{x}_2 \vee x_3) = (\bar{x}_1 \vee x_1) \vee z_1;$$

здесь через z обозначено выражение $\bar{x}_2 \vee x_3 \vee \bar{x}_2 \vee x_3$ (равнозначное выражению $\bar{x}_2 \vee x_3$). Поскольку $x_1 \vee \bar{x}_1 = 1$, мы получаем:

$$(x_1 \vee \bar{x}_2 \vee x_3) \vee (\bar{x}_1 \vee \bar{x}_2 \vee x_3) = 1 \vee z_1 = 1.$$

В) Логическая сумма всех 2^m попарно различных дизъюнктивных членов от m переменных равна единице.

Г) Логическое произведение всех 2^m попарно различных конъюнктивных членов от m переменных равно нулю.

Рассмотрим, например, следующее выражение в с.д.н.ф., содержащее m переменных (обозначим его через A):

$$A = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee \\ \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3.$$

Подразделим эту сумму на две, каждая из которых содержит по 2^{m-1} членов (в рассматриваемом случае — по четыре члена), таким образом, чтобы первая сумма включала все члены, содержащие переменную \bar{x}_1 , а вторая — все члены, содержащие переменную x_1 ; вынесем в первой сумме за скобки \bar{x}_1 , а во второй вынесем за скобки x_1 ; результат соединим знаком \vee . Тогда:

$$A = \bar{x}_1 (\bar{x}_2 \bar{x}_3 \vee \bar{x}_2 x_3 \vee x_2 \bar{x}_3 \vee x_2 x_3) \vee \\ \vee x_1 (\bar{x}_2 \bar{x}_3 \vee \bar{x}_2 x_3 \vee x_2 \bar{x}_3 \vee x_2 x_3).$$

Вынося $\bar{x}_2 \bar{x}_3 \vee \bar{x}_2 x_3 \vee x_2 \bar{x}_3 \vee x_2 x_3$ за скобки, получаем: $A = (\bar{x}_2 \bar{x}_3 \vee \bar{x}_2 x_3 \vee x_2 \bar{x}_3 \vee x_2 x_3) (\bar{x}_1 \vee x_1)$. Так как $\bar{x}_1 \vee x_1 = 1$, то в силу свойства 5 мы имеем: $A = \bar{x}_2 \bar{x}_3 \vee \bar{x}_2 x_3 \vee x_2 \bar{x}_3 \vee x_2 x_3$.

Поступая аналогично применительно к переменной x_2 и ее отрицанию, получим:

$$A = \bar{x}_2 (\bar{x}_3 \vee x_3) \vee x_2 (\bar{x}_3 \vee x_3) = \bar{x}_2 \cdot 1 \vee x_2 \cdot 1 = \bar{x}_2 \vee x_2 = 1.$$

Пусть теперь $m=2$; рассмотрим функцию

$$B = \underset{\text{д.ф.}}{(\bar{x}_1 \vee \bar{x}_2) (\bar{x}_1 \vee x_2) (x_1 \vee \bar{x}_2) (x_1 \vee x_2)}.$$

Вычислим значение произведения

$$(\bar{x}_1 \vee \bar{x}_2) (\bar{x}_1 \vee x_2) = \bar{x}_1 \bar{x}_1 \vee \bar{x}_1 x_2 \vee \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 x_2 = \\ = \bar{x}_1 \vee \bar{x}_1 (x_2 \vee \bar{x}_2) \vee 0 = \bar{x}_1 \vee \bar{x}_1 \cdot 1 = \bar{x}_1.$$

Аналогично найдем, что

$$(x_1 \vee \bar{x}_2) (x_1 \vee x_2) = x_1.$$

Следовательно, $B = \bar{x}_1 x_1 = 0$.

Из В) и Г) следует, что функции

$$R\bar{x}_1 \bar{x}_2 \vee R\bar{x}_1 x_2 \vee Rx_1 \bar{x}_2 \vee Rx_1 x_2 = R \quad (1.4)$$

$$\begin{aligned} & \text{и} \\ & (R \vee \bar{x}_1 \vee \bar{x}_2)(R \vee \bar{x}_1 \vee x_2)(R \vee x_1 \vee \bar{x}_2) \& \\ & \& (R \vee x_1 \vee x_2) = R, \quad (1.5) \end{aligned}$$

где R — некоторая произвольная функция.

Укажем теперь способ перехода от одной нормальной формы представления функции алгебры логики к другой. Из В) следует, что отрицание логической суммы из некоторых p дизъюнктивных членов (из числа всех 2^m попарно различных дизъюнктивных членов) равно сумме остальных $2^m - p$ дизъюнктивных членов, т. е. что если дана функция

$$A = P_0 \vee P_1 \vee \dots \vee P_n \quad (\text{где } n = p - 1),$$

Df

то

$$\bar{A} = P_{n+1} \vee P_{n+2} \vee \dots \vee P_{2^m-p}.$$

Вместе с тем на основании теоремы де Моргана получим, что

$$A = \bar{\bar{A}} = \overline{P_{n+1} \vee P_{n+2} \vee \dots \vee P_{2^m-p}} = C_j C_i \dots C_l,$$

причем индексы при C образуются из индексов при P указанным ранее путем.

Следовательно, для перехода от с.д.н.ф. представления некоторой функции алгебры логики к с.к.н.ф. представления этой функции нужно:

1) образовать логическую сумму дизъюнктивных членов, не входящих в с.д.н.ф.;

2) обменять местами символы конъюнкции и дизъюнкции, а аргументы заменить их отрицаниями.

Аналогичным способом осуществляется переход от с.к.н.ф. представления функций к с.д.н.ф.

Пусть, например, дана функция K :

$$K = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 = P_2 \vee P_5 \vee P_6.$$

Df

Найдем ее с.к.н.ф. Образует функцию

$$\begin{aligned} \bar{K} &= \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3 = \\ &= P_0 \vee P_1 \vee P_3 \vee P_4 \vee P_7, \end{aligned}$$

и искомая форма представления функции K запишется так:

$$K = (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \& (\bar{x}_1 \vee x_2 \vee x_3) \& (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \& \\ \& (x_1 \vee x_2 \vee \bar{x}_3) \& (x_1 \vee x_2 \vee x_3) = C_0 C_3 C_4 C_6 C_7.$$

Остановимся теперь на вопросе о полноте системы функций алгебры логики.

О п р е д е л е н и е. Система функций f_1, f_2, \dots, f_s алгебры логики называется *полной*, если любую функцию алгебры логики можно представить в виде суперпозиции функций f_1, f_2, \dots, f_s .

Система функций $\{\bar{x}, (x_1 \vee x_2), x_1 x_2\}$ является полной, что следует из теоремы 2 о возможности представления любой функции алгебры логики посредством конъюнкции, дизъюнкции и отрицания.

В частности, посредством указанных трех функций можно представить остальные «элементарные» функции, перечисленные в начале этого параграфа и определенные таблицей I.3. Из таблицы получаем:

$$x_1 \rightarrow x_2 = \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 x_2 = \bar{x}_1 \vee x_2,$$

$$x_1 \equiv x_2 = \bar{x}_1 \bar{x}_2 \vee x_1 x_2.$$

Полными являются также системы $\{\bar{x}, (x_1 \vee x_2)\}$ и $\{\bar{x}, x_1 x_2\}$,

так как в первом случае $x_1 x_2 = \overline{\bar{x}_1 \vee \bar{x}_2}$, а во втором случае $x_1 \vee x_2 = \overline{\bar{x}_1 \bar{x}_2}$.

Следовательно, указанные наборы из двух функций позволяют получить третью функцию из набора, полнота

Таблица I.7

x_1	x_2	$x_1 x_2$
0	0	1
0	1	1
1	0	1
1	1	0

которого была ранее доказана. Полную систему образует одна функция $x_1 | x_2 = \bar{x}_1 \vee \bar{x}_2$, называемая *функцией Шеффера* и определяемая таблицей I.7.

Полнота следует из того, что $\bar{x} = x|x$, $x_1 \vee x_2 = (x_1|x_1)|(x_2|x_2)$, и предыдущего замечания.

При синтезе логических сетей часто пользуются функцией $f(x_1, x_2) = x_1 \oplus x_2$, реализующей операцию сложения по mod 2 и определяемой таблицей I.8.

Таблица I.8

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Функция $x_1 \oplus x_2$ соответствует строгому, или исключающему, употреблению логического союза «или» в разговорном языке.

Из таблицы I.8 следует, что $x_1 \oplus x_2 = \bar{x}_1 x_2 \vee x_1 \bar{x}_2$. Функция $x_1 \oplus x_2$ вместе с конъюнкцией $x_1 x_2$ (или дизъюнкцией $x_1 \vee x_2$) и константой 1 (или 0) образует полную систему. Действительно, $x \oplus 1 = \bar{x}$, и тем самым, как было показано раньше, обеспечивается полнота рассматриваемого набора.

Из приведенных примеров видно, что существуют различные полные системы функций алгебры логики. Каждая из них может быть принята в качестве набора «элементарных» функций, и любая функция алгебры логики может быть записана в виде формулы, выраженной через «элементарные» функции принятого набора. Выбор той или иной полной системы функции в качестве исходной зависит от характера рассматриваемой задачи и конкретных условий ее решения.

В связи с этим приобретает интерес вопрос о том, как установить функциональную полноту или неполноту заданной системы функций. Ниже мы сформулируем без доказательства критерии полноты системы функций алгебры логики. Предварительно отметим следующие пять свойств, каждым из которых функция алгебры логики $f(x_1, x_2, \dots, x_m)$ может обладать или не обладать:

1. Свойство *сохранения нуля*:

$$f(0, 0, \dots, 0) = 0.$$

2. Свойство *сохранения единицы*:

$$f(1, 1, \dots, 1) = 1.$$

Примерами функций, обладающих свойствами 1 и 2, являются $f_1(x_1, x_2) = x_1 x_2$ и $f_2(x_1, x_2) = x_1 \vee x_2$, так как

$$f_1(0, 0) = 0, \quad f_2(0, 0) = 0,$$

$$f_1(1, 1) = 1, \quad f_2(1, 1) = 1.$$

3. Свойство *самодвойственности*, или *нечетности*, функций:

$$f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m) = \bar{f}(x_1, x_2, \dots, x_m).$$

Примером самодвойственности функции является, например, функция $f(x) = \bar{x}$, так как $f(\bar{x}) = \bar{f}(x)$.

4. Свойство *монотонности*: если $x_1 \leq x'_1, x_2 \leq x'_2, \dots, x_m \leq x'_m$, то $f(x_1, x_2, \dots, x_m) \leq f(x'_1, x'_2, \dots, x'_m)$. Функции $f_1(x_1, x_2) = x_1 x_2$ и $f_2(x_1, x_2) = x_1 \vee x_2$ являются монотонными, поскольку, например,

$$f_1(0, 0) < f_1(1, 1); \quad f_2(0, 0) < f_2(1, 1).$$

Отметим попутно, что всякая формула, записанная через конъюнкцию и дизъюнкцию (без отрицания), задает некоторую монотонную функцию.

5. Свойство *линейности*: функция $f(x_1, x_2, \dots, x_m)$ называется линейной, если возможно представление

$$f(x_1, x_2, \dots, x_m) = a_0 \oplus a_1 x_1 \oplus \dots \oplus a_n x_n^*,$$

где a_i — постоянные (0 или 1). Отметим, что значение линейной функции $f(x_1, x_2, \dots, x_m)$ не изменяется, если четное число независимых переменных заменить их отрицаниями. Примером линейной функции является

$$f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3 =$$

$$= (x_1 x_2 \vee \bar{x}_1 \bar{x}_2) x_3 \vee (x_1 \bar{x}_2 \vee \bar{x}_1 x_2) \bar{x}_3.$$

*) Операция \oplus коммутативна и ассоциативна, в силу чего мы можем не писать здесь скобок, указывающих порядок выполнения операций сложения по mod 2.

Нетрудно убедиться, что функция

$$f'(x_1, x_2, x_3) = x_1 \oplus \bar{x}_2 \oplus \bar{x}_3 = f(x_1 x_2, x_3).$$

Теорема 3 о функциональной полноте системы. Для того чтобы система функций $\varphi_1(x_1, x_2, \dots, x_m)$, $\varphi_2(x_1, x_2, \dots, x_m)$, ..., $\varphi_k(x_1, \dots, x_m)$ алгебры логики была полной, необходимо и достаточно, чтобы она содержала:

1) функцию, не сохраняющую константу 0, 2) функцию, не сохраняющую константу 1, 3) функцию, не являющуюся самодвойственной, 4) функцию, не являющуюся линейной, 5) функцию, не являющуюся монотонной, т. е. чтобы каждое из перечисленных выше свойств не принадлежало какой-либо из функций в этой системе.

С л е д с т в и е. Если обнаружена полная система, насчитывающая более пяти элементарных функций, то из нее можно извлечь подсистему, содержащую не более пяти функций и также являющуюся полной *).

Разъясним сформулированную теорему на некоторых простых примерах. Система, состоящая из отрицания и дизъюнкции (или отрицания и конъюнкции), удовлетворяет критерию полноты. В самом деле, свойства 1, 2, 4 не принадлежат функции отрицания, так как она не сохраняет нуля, не сохраняет единицы и не является монотонной, а свойства 3 и 5 не принадлежат дизъюнкции (и конъюнкции), поскольку эти функции не являются самодвойственными и не линейны. Критерию полноты удовлетворяет также функция Шеффера, ибо она не обладает ни одним из отмеченных пяти свойств.

Система функций $x_1 x_2$, $x_1 \oplus x_2$, 1 является полной, поскольку для каждого из пяти свойств существует в этой системе такая функция, которая его не имеет. Легко проверить, что при выбрасывании любой из трех функций система становится неполной. Действительно, исключим из этой системы, например, константу 1. Тогда в ней не

*) С. В. Яблонский показал, что из всякой полной системы функций можно выбрать полную подсистему, состоящую не более чем из четырех функций. См. С. В. Яблонский, О суперпозициях функций алгебры логики, Математический сборник, т. 30 (72), 1952, стр. 329—348.

окажется функцией, для которой не выполняется свойство 1 (сохранение нуля). Если из системы исключить конъюнкцию, то в ней не окажется функций, для которых не выполняются свойства самодвойственности и линейности.

Рассмотрим систему, образуемую импликацией $(x_1 \rightarrow x_2)$ и конъюнкцией $(x_1 x_2)$. Она не является полной, поскольку в ней отсутствует функция, для которой не выполняется свойство сохранения единицы. Если включить в эту систему константу 0, то система становится полной. В этой системе отрицание определяется на основе следующего соотношения: $x \rightarrow 0 = \bar{x} \vee 0 = \bar{x}$, а отрицание, как было показано выше, вместе с конъюнкцией (или дизъюнкцией) дает полный набор.

§ 3. Минимизация формул алгебры логики

В предыдущем параграфе было показано, что всякая функция алгебры логики $f(x_1, x_2, \dots, x_m)$, заданная таблицей, может быть представлена совершенной дизъюнктивной нормальной формой

$$f(x_1, x_2, \dots, x_m) = \bigvee_{f(\sigma_1, \sigma_2, \dots, \sigma_m)=1} x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m}, \quad (1.3)$$

включающей не более 2^m дизъюнктивных членов. Вместе с тем та же функция может быть задана совершенной конъюнктивной нормальной формой, включающей не более 2^m конъюнктивных членов.

Совершенная нормальная форма часто допускает упрощение, достигаемое путем различных тождественных преобразований. При этом получается формула, эквивалентная исходной, но содержащая меньшее число вхождений букв.

Как было сказано в § 2, две формулы алгебры логики называются *эквивалентными*, если они задают одну и ту же функцию алгебры логики.

Будем в дальнейшем называть конъюнкцию *Рэлементарной*, если она является произведением r переменных, взятых из множества, состоящего из m переменных, каждая из которых входит в произведение со степенью $\sigma = 0$ или $\sigma = 1$ и при этом не более одного раза. Число сомножителей

в произведении определяет ранг элементарной конъюнкции. Так, например, конъюнкция $p(x_1, x_2, \dots, x_5) = x_1^{\sigma_1} x_2^{\sigma_2} x_4^{\sigma_4}$ является элементарной конъюнкцией 3-го ранга. Конъюнкция $p(x_1, x_2, \dots, x_m) = 1$ считается элементарной конъюнкцией 0-го ранга.

Дизъюнкцию элементарных конъюнкций ненулевого ранга назовем *дизъюнктивной нормальной формой* (д.н.ф.). Аналогично определим понятия *элементарной дизъюнкции* и *конъюнктивной нормальной формы* (к.н.ф.).

П р и м е р ы.

1. Формула $\bar{x}_1 x_3 \vee x_2 \bar{x}_3 \vee \bar{x}_1 x_3 \bar{x}_4$ является дизъюнктивной нормальной формой представления некоторой функции $f(x_1, x_2, x_3, x_4)$.

2. Дизъюнкция $g(x_1, x_2, x_3, x_4) = \bar{x}_1 \vee x_3 \vee \bar{x}_4$ является элементарной дизъюнкцией 3-го ранга. А формула

$$(\bar{x}_1 \vee x_3 \vee \bar{x}_4)(\bar{x}_1 \vee x_2)(\bar{x}_3 \vee \bar{x}_4)$$

является конъюнктивной нормальной формой представления некоторой функции $\varphi(x_1, x_2, x_3, x_4)$.

О п р е д е л е н и е. Дизъюнктивная нормальная форма (соответственно конъюнктивная нормальная форма) называется *минимальной*, если она включает минимальное число букв *) по сравнению с другими эквивалентными ей дизъюнктивными нормальными формами (соответственно конъюнктивными нормальными формами).

Одной из важных задач, возникающих при синтезе логической сети, является *минимизация формулы*, выражающей данную функцию алгебры логики, путем тождественных преобразований.

Тривиальный алгоритм решения этой задачи сводится к перебору всех эквивалентных формул с числом букв, не превосходящим максимально возможного числа букв μ совершенной нормальной формы, определяемого равенством

$$\mu = m \cdot 2^m. \quad (1.6)$$

*) Под «буквой» формулы здесь имеется в виду вхождение буквы в данную формулу, так что одна и та же буква, скажем, x_1 , встречающаяся в формуле, например, три раза, при подсчете числа букв в формуле принимается каждый раз за новую букву.

В результате такого перебора будет обнаружено минимальное представление заданной функции алгебры логики. Однако даже для функций, зависящих от небольшого числа переменных, такой алгоритм практически невыполним, так как обзору подлежит очень большое число эквивалентных формул.

В настоящее время при решении практических задач синтеза логической сети пользуются различными, более эффективными, алгоритмами для упрощения совершенной нормальной формы. Важно отметить, что применение любого из этих алгоритмов не гарантирует получения минимальных форм, но позволяет достаточно эффективно (в смысле числа шагов) получить формы, близкие к минимальным. Вместе с тем пользование этими алгоритмами не исключает полностью элемента перебора. Это обуславливается тем, что упрощение может быть достигнуто различными путями. Кроме того, результат упрощения не является однозначным. Наконец, неоднозначно определяется минимальное представление функции алгебры логики. Одна и та же функция может иметь несколько минимальных форм. Так, например, для функции

$$f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \\ \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3$$

существуют две эквивалентные минимальные дизъюнктивные формы:

$$\bar{x}_1\bar{x}_2 \vee x_1x_3 \vee x_2\bar{x}_3$$

и

$$\bar{x}_1\bar{x}_3 \vee x_1x_2 \vee \bar{x}_2x_3.$$

Упрощение совершенной дизъюнктивной нормальной формы чаще всего осуществляется посредством следующих преобразований:

1. $fx \vee f\bar{x} = f$ — склеивание,
2. $fx \vee f = f$ — поглощение,
3. $fx_1 \vee fx_2 = f(x_1 \vee x_2)$ — вынос за скобки,

а упрощение совершенной конъюнктивной нормальной формы — посредством преобразований:

$$1'. (f \vee x) \& (f \vee \bar{x}) = f,$$

$$2'. (f \vee x) \& f = f,$$

$$3'. (f \vee x_1) \& (f \vee x_2) = f \vee (x_1 \& x_2),$$

носящих аналогичные названия.

Отметим, что первые два вида преобразований не выводят функцию алгебры логики из дизъюнктивной (соответственно конъюнктивной) нормальной формы представления. Третье преобразование приводит к выражению, не являющемуся нормальной формой.

Содержание этого параграфа посвящено рассмотрению алгоритмов, основанных на операциях склеивания и поглощения. Минимизируемая функция задается полной таблицей или совершенной нормальной дизъюнктивной формой.

Укажем предварительно один полезный критерий, позволяющий использовать операцию поглощения для упрощения дизъюнктивной нормальной формы. Введем для этого понятие ортогональности элементарных конъюнкций.

О п р е д е л е н и е. Элементарные конъюнкции p_i и p_j называются ортогональными, если $p_i p_j = 0$.

Очевидно, что для ортогональности элементарных конъюнкций ненулевых рангов необходимо и достаточно, чтобы нашлась такая переменная, которая входит в одну конъюнкцию без отрицания ($\sigma=1$), а в другую — с отрицанием ($\sigma=0$). Примером ортогональных конъюнкций являются

$$p' = x_1 x_2 \quad \text{и} \quad p'' = \bar{x}_1 x_3;$$

их произведение $p' p'' = 0$.

О п р е д е л е н и е. Дизъюнкция $p_1 \vee p_2 \vee \dots \vee p_k$ элементарных конъюнкций p_j поглощает элементарную конъюнкцию p , если формула $p \rightarrow p_1 \vee p_2 \vee \dots \vee p_k$ выражает функцию, тождественно равную единице.

Понятие поглощения важно по следующей причине. Если известно, что $p_1 \vee p_2 \vee \dots \vee p_k$ поглощает p , т. е. что $p \rightarrow p_1 \vee p_2 \vee \dots \vee p_k = 1$, то $p \vee p_1 \vee p_2 \vee \dots \vee p_k = p_1 \vee p_2 \vee \dots \vee p_k$ (элементарная конъюнкция p может

быть вычеркнута). Введенное на стр. 47 понятие поглощения (f поглощает fx , т. е. $fx \vee f = f$) является частным случаем поглощения дизъюнкцией элементарных конъюнкций $p_1 \vee p_2 \vee \dots \vee p_k$ элементарной конъюнкции p (за p берется fx , за $p_1 \vee p_2 \vee \dots \vee p_k$ берется f).

Сформулируем теперь следующую теорему, доказательство которой приводить здесь не будем.

Т е о р е м а 4. **К р и т е р и й п о г л о щ е н и я *).** Для того чтобы дизъюнкция $p_1 \vee p_2 \vee \dots \vee p_k$ поглощала элементарную конъюнкцию p , не ортогональную каждой элементарной конъюнкции из $p_1 \vee p_2 \vee \dots \vee p_k$, необходимо и достаточно, чтобы каждую конъюнкцию p_i можно было представить в виде $p_i = p'_i p''_i$ (конъюнкции p'_i и p''_i могут вырождаться в 1) с соблюдением следующих условий:

$$1. \quad \bigvee_{i=1}^k p''_i = 1,$$

$$2. \quad p \rightarrow p'_1 p'_2 \dots p'_k = 1.$$

Здесь через p'_i обозначены конъюнкции всех членов, общих для p_i и p , а через p''_i — конъюнкции оставшихся членов из p_i .

П р и м е р. Пусть дана дизъюнкция

$$D \underset{\text{Df}}{=} \bar{x}_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3 \vee \bar{x}_1 x_3 x_4 \vee x_2 x_3 x_4.$$

Применим теорему 4 к дизъюнкции

$$p_1 \vee p_2 \vee p_3 \underset{\text{Df}}{=} \bar{x}_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3 \vee \bar{x}_1 x_3 x_4$$

и конъюнкции

$$p \underset{\text{Df}}{=} x_2 x_3 x_4.$$

*) См. Ю. Журавлев, Об отделимости подмножеств n -мерного единичного куба. Труды Матем. ин-та им. В. А. Стеклова, т. 51, 1958. С. В. Яблонский, Функциональные построения в k -значной логике, там же.

Представим элементарные конъюнкции в $p_1 \vee p_2 \vee p_3$ в следующем виде:

$$p_1 = p'_1 p''_1 = x_3 \cdot (\bar{x}_1 \bar{x}_2),$$

$$p_2 = p'_2 p''_2 = (x_2 x_3) \cdot x_1,$$

$$p_3 = p'_3 p''_3 = (x_3 x_4) \cdot \bar{x}_1.$$

При этом

$$\bigvee_{i=1}^3 p''_i = \bar{x}_1 \bar{x}_2 \vee x_1 \vee \bar{x}_1 = 1,$$

$$p'_1 p'_2 p'_3 = x_2 x_3 x_4$$

и, следовательно,

$$p \rightarrow p'_1 p'_2 p'_3 = x_2 x_3 x_4 \rightarrow x_2 x_3 x_4 = 1.$$

Так как оба условия, сформулированные в теореме 4, выполняются, то дизъюнкция $p_1 \vee p_2 \vee p_3$ поглощает элементарную конъюнкцию p , в силу чего

$$D = \bar{x}_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3 \vee \bar{x}_1 x_2 x_3.$$

Перейдем теперь к описанию более общих методов построения минимальных дизъюнктивных форм. Одним из широко используемых является метод непосредственного упрощения совершенной дизъюнктивной нормальной формы.

Назовем конъюнкции p_i^r и p_j^r ранга r соседними, если существует такая переменная x_k , что

$$p_i^r = p^{r-1} x_k^\sigma \quad \text{и} \quad p_j^r = p^{r-1} \bar{x}_k^\sigma.$$

Конъюнкция p ранга r называется **изолированной** по отношению к данному множеству конъюнкций r -го, ранга, если в нем не существует ни одной конъюнкции, соседней с p .

Непосредственную минимизацию заданной функции алгебры логики будем проводить в следующем порядке. Выделим из совершенной дизъюнктивной нормальной формы изолированные конъюнкции ранга r . Обозначим

их через $p_1^r, p_2^r, \dots, p_j^r$. Оставшиеся конъюнкции не являются изолированными. Для каждой из них выберем соседнее с ней конъюнкции, и к каждой паре соседних конъюнкций применим операцию склеивания:

$$p_i^r \vee p_u^r = p^{r-1} x_i^{\sigma} \vee p^{r-1} \bar{x}_i^{\sigma} = p^{r-1}.$$

В результате систематического применения операции склеивания ко всем соседним конъюнкциям получим множество конъюнкций ранга $r-1$. В дизъюнктивной нормальной форме, образованной этим множеством, выполним приведение подобных членов.

Для полученной таким образом дизъюнктивной формы повторим описанный процесс упрощения и получим множество изолированных конъюнкций ранга $r-1$ и дизъюнктивную форму, образуемую конъюнкциями ранга $r-2$ и т. д. Этот процесс завершится после s повторений. В результате получим упрощенную дизъюнктивную нормальную форму функции в виде

$$\begin{aligned} & (p_1^r \vee p_2^r \vee \dots \vee p_j^r) \vee (p_1^{r-1} \vee p_2^{r-1} \vee \dots \vee p_i^{r-1}) \vee \dots \\ & \dots \vee (p_1^{r-s} \vee p_2^{r-s} \vee \dots \vee p_h^{r-s}) \vee \dots \\ & \dots \vee (p_1^{r-s+1} \vee p_2^{r-s+1} \vee \dots \vee p_j^{r-s+1}). \end{aligned}$$

Применяя к этой форме указанный выше критерий поглощения, исключим из нее лишние конъюнкции и получим так называемую *тупиковую форму*, т. е. нормальную дизъюнктивную форму, эквивалентную заданной функции алгебры логики и не допускающую дальнейших поглощений. Полученная тупиковая форма может случайно оказаться минимальной. В общем же случае для отыскания минимальной формы необходим перебор тупиковых форм.

Пример. Минимизировать функцию $f(x_1, x_2, x_3)$, заданную следующей совершенной нормальной формой:

$$\bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3.$$

В ней отсутствуют изолированные конъюнкции 3-го ранга, а конъюнктивные члены образуют три пары

соседних конъюнкций

$$(\bar{x}_1 \bar{x}_2 \bar{x}_3, \bar{x}_1 \bar{x}_2 x_3),$$

$$(x_1 \bar{x}_2 x_3, x_1 x_2 x_3),$$

$$(\bar{x}_1 \bar{x}_2 x_3, x_1 \bar{x}_2 x_3).$$

Применяя к ним операции склеивания, получим конъюнкции второго ранга: $\bar{x}_1 \bar{x}_2$, $x_1 x_3$, $\bar{x}_2 x_3$. Строим из них дизъюнктивную нормальную форму

$$\bar{x}_1 \bar{x}_2 \vee x_1 x_3 \vee \bar{x}_2 x_3.$$

Она состоит только из изолированных конъюнкций. Применяя к ней критерий поглощения, получим тупиковую форму, являющуюся в данном случае минимальной:

$$\bar{x}_1 \bar{x}_2 \vee x_1 x_3.$$

Отметим, что в этом примере минимальная форма может быть получена без операции поглощения, а только посредством операции склеивания, примененной к парам $\{\bar{x}_1 \bar{x}_2 \bar{x}_3, \bar{x}_1 \bar{x}_2 x_3\}$ и $\{x_1 \bar{x}_2 x_3, x_1 x_2 x_3\}$ соседних конъюнкций. Однако в более сложных случаях возникает необходимость систематической реализации изложенного алгоритма минимизации.

Если исходная функция задана в конъюнктивной нормальной форме, то построение минимальной формы методом непосредственного упрощения осуществляется так. Вначале, пользуясь законом распределительности конъюнкции относительно дизъюнкции, раскрывают в конъюнктивной нормальной форме скобки. После этого приводят подобные члены и устраняют элементарные поглощения. Таким образом получается дизъюнктивная нормальная форма, минимизацию которой проводят указанным алгоритмом.

П р и м е р. Минимизировать функцию A , заданную в конъюнктивной нормальной форме

$$A = (\bar{x}_1 \vee x_2 \vee x_3) \& (x_1 \vee \bar{x}_2 \vee \bar{x}_3).$$

Раскроем скобки и получим дизъюнктивную нормальную форму

$$A = \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 \bar{x}_3 \vee x_2 \bar{x}_3 \vee x_2 \bar{x}_3 \vee x_1 x_2 \vee x_1 x_3,$$

включающую только изолированные конъюнкции 2-го ранга. Применим к ней критерий поглощения. Заметим, что конъюнкции, входящие в дизъюнктивную нормальную форму

$$S = \bar{x}_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee x_1 x_3,$$

можно представить следующими выражениями:

$$\begin{aligned} p_1 &= p'_1 p''_1 = \bar{x}_1 \& \bar{x}_2; & p_3 &= 1 \& (\bar{x}_1 \& \bar{x}_2); & p_1 &= \bar{x}_2 \& \bar{x}_1, \\ p_2 &= p'_2 p''_2 = \bar{x}_3 \& x_2; & p_1 &= x_2 \& \bar{x}_3; & p_3 &= 1 \& (x_2 \& \bar{x}_3), \\ p_3 &= p'_3 p''_3 = 1 \& (x_1 \& x_3); & p_2 &= x_1 \& x_3; & p_2 &= x_3 \& x_1. \end{aligned}$$

В каждом из указанных трех случаев $\bigvee_{i=1}^3 p_i'' = 1$.

Кроме того, в первом случае $p'_1 p'_2 p'_3 = \bar{x}_1 \bar{x}_3$, во втором случае это произведение равно $x_1 x_2$ и в третьем случае оно равно $\bar{x}_2 x_3$. Следовательно, дизъюнктивная форма S поглощает остальные три конъюнкции указанной выше дизъюнктивной нормальной формы функции A и дает минимальное представление этой функции. Поступая аналогичным образом в отношении дизъюнктивной формы

$$S' = x_1 x_2 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 x_3,$$

найдем, что она поглощает остальные конъюнкции в A . Таким образом, функция A имеет две минимальные дизъюнктивные формы:

$$A = \bar{x}_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee x_1 x_3,$$

$$A = x_1 x_2 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 x_3.$$

Из приведенных примеров видно, что второй этап метода непосредственной минимизации — применение критерия поглощения — сопряжен с перебором различных вариантов.

Практически удобным методом минимизации функций с числом переменных, не превышающим пяти, является метод неопределенных коэффициентов *). Сущность этого

*) См. «Синтез электронных вычислительных и управляющих схем», перев. с англ., ИЛ, М., 1954 (книга написана коллективом вычислительной лаборатории Гарвардского университета, США).

метода мы покажем в применении к минимизации функции алгебры логики от трех переменных. Составим для нее дизъюнктивную форму, включающую всевозможные конъюнкции 1-го, 2-го и 3-го рангов, причем каждая из них входит с двоичным коэффициентом $A_{ijk}^{\sigma_i \sigma_j \sigma_k}$. Эта дизъюнктивная форма запишется так:

$$f(x_1, x_2, x_3) = \left. \begin{aligned} & A_1^0 \bar{x}_1 \vee A_2^0 \bar{x}_2 \vee A_3^0 \bar{x}_3 \vee A_1^1 x_1 \vee \\ & \vee A_2^1 x_2 \vee A_3^1 x_3 \vee A_{12}^{00} \bar{x}_1 \bar{x}_2 \vee A_{13}^{00} \bar{x}_1 \bar{x}_3 \vee A_{23}^{00} \bar{x}_2 \bar{x}_3 \vee \\ & \vee A_{12}^{01} \bar{x}_1 x_2 \vee A_{13}^{01} \bar{x}_1 x_3 \vee A_{23}^{01} x_2 x_3 \vee A_{12}^{10} x_1 \bar{x}_2 \vee \\ & \vee A_{13}^{10} x_1 \bar{x}_3 \vee A_{23}^{10} x_2 \bar{x}_3 \vee A_{12}^{11} x_1 x_2 \vee A_{13}^{11} x_1 x_3 \vee \\ & \vee A_{23}^{11} x_2 x_3 \vee A_{123}^{000} \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee A_{123}^{001} \bar{x}_1 \bar{x}_2 x_3 \vee \\ & \vee A_{123}^{010} \bar{x}_1 x_2 \bar{x}_3 \vee A_{123}^{011} \bar{x}_1 x_2 x_3 \vee A_{123}^{100} x_1 \bar{x}_2 \bar{x}_3 \vee \\ & \vee A_{123}^{101} x_1 \bar{x}_2 x_3 \vee A_{123}^{110} x_1 x_2 \bar{x}_3 \vee A_{123}^{111} x_1 x_2 x_3. \end{aligned} \right\} (1.7)$$

При значениях аргументов

$$x_1 = x_2 = x_3 = 0$$

дизъюнктивная форма (1.7) примет следующий вид:

$$f(0, 0, 0) = A_1^0 \vee A_2^0 \vee A_3^0 \vee A_{12}^{00} \vee A_{13}^{00} \vee A_{23}^{00} \vee A_{123}^{000}.$$

Аналогично при

$$x_1 = x_2 = \bar{x}_3 = 0$$

мы имеем:

$$f(0, 0, 1) = A_1^0 \vee A_2^0 \vee A_3^1 \vee A_{12}^{00} \vee A_{13}^{01} \vee A_{23}^{01} \vee A_{123}^{001}$$

и так далее.

Система уравнений

$$\left. \begin{aligned} & A_1^0 \vee A_2^0 \vee A_3^0 \vee A_{12}^{00} \vee A_{13}^{00} \vee A_{23}^{00} \vee A_{123}^{000} = f(0, 0, 0), \\ & A_1^0 \vee A_2^0 \vee A_3^1 \vee A_{12}^{00} \vee A_{13}^{01} \vee A_{23}^{01} \vee A_{123}^{001} = f(0, 0, 1), \\ & A_1^0 \vee A_2^1 \vee A_3^0 \vee A_{12}^{01} \vee A_{13}^{00} \vee A_{23}^{10} \vee A_{123}^{010} = f(0, 1, 0), \\ & A_1^0 \vee A_2^1 \vee A_3^1 \vee A_{12}^{01} \vee A_{13}^{01} \vee A_{23}^{11} \vee A_{123}^{011} = f(0, 1, 1), \\ & A_1^1 \vee A_2^0 \vee A_3^0 \vee A_{12}^{10} \vee A_{13}^{10} \vee A_{23}^{00} \vee A_{123}^{100} = f(1, 0, 0), \\ & A_1^1 \vee A_2^0 \vee A_3^1 \vee A_{12}^{10} \vee A_{13}^{11} \vee A_{23}^{01} \vee A_{123}^{101} = f(1, 0, 1), \\ & A_1^1 \vee A_2^1 \vee A_3^0 \vee A_{12}^{11} \vee A_{13}^{10} \vee A_{23}^{10} \vee A_{123}^{110} = f(1, 1, 0), \\ & A_1^1 \vee A_2^1 \vee A_3^1 \vee A_{12}^{11} \vee A_{13}^{11} \vee A_{23}^{11} \vee A_{123}^{111} = f(1, 1, 1) \end{aligned} \right\} (1.8)$$

определяет искомые значения коэффициентов различных дизъюнктивных форм для заданной функции алгебры логики. Путем их сравнения легко определяется минимальная форма.

П р и м е р. Минимизировать методом неопределенных коэффициентов функцию $f(x_1, x_2, x_3)$, заданную таблицей I.9.

Таблица I.9

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Система уравнений (1.8) в рассматриваемом случае примет следующий вид:

$$\begin{aligned}
 A_1^0 \vee A_2^0 \vee A_3^0 \vee A_{12}^{00} \vee A_{13}^{00} \vee A_{23}^{00} \vee A_{123}^{000} &= 0, \\
 A_1^0 \vee A_2^0 \vee A_3^1 \vee A_{12}^{00} \vee A_{13}^{01} \vee A_{23}^{01} \vee A_{123}^{001} &= 1, \\
 A_1^0 \vee A_2^1 \vee A_3^0 \vee A_{12}^{01} \vee A_{13}^{00} \vee A_{23}^{10} \vee A_{123}^{010} &= 0, \\
 A_1^0 \vee A_2^1 \vee A_3^1 \vee A_{12}^{01} \vee A_{13}^{01} \vee A_{23}^{11} \vee A_{123}^{011} &= 0, \\
 A_1^1 \vee A_2^0 \vee A_3^0 \vee A_{12}^{10} \vee A_{13}^{10} \vee A_{23}^{00} \vee A_{123}^{100} &= 0, \\
 A_1^1 \vee A_2^0 \vee A_3^1 \vee A_{12}^{10} \vee A_{13}^{11} \vee A_{23}^{01} \vee A_{123}^{101} &= 0, \\
 A_1^1 \vee A_2^1 \vee A_3^0 \vee A_{12}^{11} \vee A_{13}^{10} \vee A_{23}^{10} \vee A_{123}^{110} &= 1, \\
 A_1^1 \vee A_2^1 \vee A_3^1 \vee A_{12}^{11} \vee A_{13}^{11} \vee A_{23}^{11} \vee A_{123}^{111} &= 1.
 \end{aligned}$$

Очевидно, что при этом

$$\begin{aligned}
 A_1^0 &= A_2^0 = A_3^0 = A_{12}^{00} = A_{13}^{00} = A_{23}^{00} = A_{123}^{000} = A_2^1 = A_{12}^{01} = A_{23}^{10} = \\
 &= A_{123}^{010} = A_3^1 = A_{13}^{01} = A_{23}^{11} = A_{123}^{011} = A_1^1 = A_{12}^{10} = A_{13}^{10} = A_{123}^{100} = \\
 &= A_{13}^{11} = A_{23}^{01} = A_{123}^{101} = 0.
 \end{aligned}$$

Написанная система уравнений примет вид:

$$\begin{aligned} A_{123}^{001} &= 1, \\ A_{12}^{11} \vee A_{123}^{110} &= 1, \\ A_{12}^{11} \vee A_{123}^{111} &= 1. \end{aligned}$$

Она допускает дальнейшее упрощение. Для этого нужно в каждом уравнении оставить коэффициенты, соответствующие конъюнкциям наименьшего ранга. Остальные коэффициенты принимаются равными нулю. В рассматриваемом случае примем

$$A_{123}^{110} = A_{122}^{111} = 0.$$

При этом $A_{123}^{001} = 1$, $A_{12}^{11} = 1$. Следовательно, минимальная дизъюнктивная нормальная форма заданной функции алгебры логики запишется так:

$$f(x_1, x_2, x_3) = x_1 x_2 \vee \bar{x}_1 \bar{x}_2 x_3.$$

Представляется очевидным, что трудности оперирования с функциями алгебры логики вообще существенно возрастают с увеличением числа переменных. Это обуславливается тем, что число функций, зависящих от n аргументов, очень быстро возрастает по мере роста n . Вместе

Таблица 1.10

$x_3 x_4 x_5$ $x_1 x_2$	$\bar{x}_3 \bar{x}_4 \bar{x}_5$	$\bar{x}_3 \bar{x}_4 x_5$	$\bar{x}_3 x_4 \bar{x}_5$	$\bar{x}_3 x_4 x_5$	$x_3 \bar{x}_4 \bar{x}_5$	$x_3 \bar{x}_4 x_5$	$x_3 x_4 \bar{x}_5$	$x_3 x_4 x_5$
$\bar{x}_1 \bar{x}_2$		1	1					
$\bar{x}_1 x_2$			1			1		
$x_1 \bar{x}_2$		1			1			
$x_1 x_2$				1			1	

с тем сильно усложняется задание функций многих переменных с помощью того типа таблиц, которыми мы до сих пор пользовались. Для задания функций, содержащих до 8—10 переменных, удобно применять более компактные таблицы. Ниже приведена такая таблица для функции пяти переменных (см. табл. 1.10).

В общем случае таблицу этого вида строят аналогичным образом. Наборы двоичных аргументов разбивают на две подгруппы по m' и m'' аргументов в каждой. При четном числе переменных подгруппы могут включать одинаковое число переменных. Таблицу составляют из $2^{m'}$ строк и $2^{m''}$ столбцов.

Если для некоторого набора $x_1^{\sigma_1}, x_2^{\sigma_2}, \dots, x_m^{\sigma_m}$ функция $f(x_1, x_2, \dots, x_m) = 1$, то на пересечении строки и столбца, содержащих подгруппы этого набора, ставится 1. Это продолжается для всех наборов значений переменных, для которых $f = 1$. Заметим, что иногда при табличном задании функции алгебры логики ее значения бывают указаны не для всех возможных наборов аргументов *).

Это значит, что таблица оказывается неполной, а представляемая ею функция алгебры логики — не всюду определенной. Между тем всякая формула алгебры логики задает функцию, которая всюду определена, т. е. к каждому набору значений своих (двоичных) аргументов ставит в соответствие некоторое определенное (двоичное) значение функции. Поэтому, если мы намерены функцию, заданную неполной таблицей, выразить посредством операций алгебры логики, возникает необходимость расширения заданной (неполной) таблицы до полной, т. е. содержащей указания о значении функции для всех наборов аргументов, причем это расширение можно про-

*) Эта ситуация аналогична той, которая возникает, например, при двоичном кодировании некоторого алфавита в случае, когда число попарно различных кодовых групп оказывается больше, чем число попарно различных символов в алфавите. Так, при кодировании десятичных цифр 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 равномерными кодовыми группами двоичных знаков минимальное число необходимых для этого знаков в каждой группе равно четырем. В этом случае используются только десять из шестнадцати возможных кодовых групп; остальные шесть групп остаются неиспользованными.

водить по-разному. Расширение достигается путем включения в таблицу неиспользованных наборов двоичных аргументов: поскольку в исходной информации не содержится указаний на то, какие значения функции алгебры логики сопоставлены этим наборам, эти значения могут быть выбраны произвольно. Во многих случаях надлежащий выбор этих значений помогает существенно упростить аналитическое представление таблично заданной функции.

Поясним изложенное на простом примере. Пусть некоторая функция алгебры логики от трех переменных задана таблицей I.11.

Таблица I.11

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	1
1	0	0	1
1	0	1	0
1	1	0	1

Эта таблица является неполной; в ней отсутствуют наборы 011 и 111 и соответствующие им значения функций. Включим в таблицу эти наборы аргументов и сопоставим им значения функций, выбранных так, как показано в таблице I.12.

Таблица I.12

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Записав эту функцию в с.д.н.ф. и проведя возможные упрощения, мы получаем следующую минимальную форму нашей функции:

$$f(x_1, x_2, x_3) = \bar{x}_1 x_2 \vee x_1 \bar{x}_3.$$

Любое другое доопределение заданной функции приводит к более длинной формуле *).

§ 4. Разложение функций алгебры логики

Соотношение (1.1), полученное в § 2, приводит к одному из частных случаев разложения функции алгебры логики по k переменным и ее представлению в виде дизъюнкции 2^k конъюнкций. При рассмотрении некоторых вопросов синтеза логической сети оказывается полезным применять и другие способы разложения. В связи с этим в настоящем параграфе излагается общая задача разложения функций алгебры логики.

Введем предварительно некоторые понятия. Будем говорить, что функция $f(v_1, v_2, \dots, v_i, \dots, v_n)$ существенно зависит от переменной v_i , если при некоторых фиксированных значениях остальных переменных

$$f(\sigma_1, \sigma_2, \dots, \sigma_{i-1}, 0, \sigma_{i+1}, \dots, \sigma_n) \neq f(\sigma_1, \sigma_2, \dots, \dots, \sigma_{i-1}, 1, \sigma_{i+1}, \dots, \sigma_n). \quad (1.9)$$

*) Вопросы учета неиспользуемых кодовых групп занимался В. Н. Рогинский. В наиболее общей форме решение этой задачи было получено Ю. И. Журавлевым.

Из литературы, освещающей проблему минимизации формул алгебры логики (эта литература весьма значительна), помимо работ, отмеченных ранее, укажем следующие: Г. Н. Поваров, Метод синтеза вычислительных и управляющих контактных схем, Автоматика и телемеханика, 1957, т. 18, № 2; В. Н. Рогинский и А. Д. Харкевич, Релейные схемы в телефонии, Связьиздат, М., 1955; В. И. Шестаков, Моделирование операций исчисления высказываний посредством релейно-контактных схем, Сб. «Логические исследования», Изд. АН СССР, М., 1959; W. Keister, A. E. Richi, S. H. Washburn, The design of switching circuits. Van Nostrand, N. J., 1951; W. Quine, The problem of simplifying truth functions. The American Mathem. Monthly, 1952, v. 58, № 8; A. Svoboda, Synthesa reléových sítí, Stroje na zpracování informací, Sborník, Z., Praha, 1954.

С л е д с т в и е. Из соотношения (1.9) следует, что при фиксированных значениях $\sigma_j (j \neq i)$ функция $f(\sigma_1, \sigma_2, \dots, v_i, \dots, \sigma_n)$ не есть постоянная и равна либо v_i , либо \bar{v}_i :

$$f(\sigma_1, \sigma_2, \dots, v_i, \dots, \sigma_n) = \begin{cases} v_i, \\ \bar{v}_i. \end{cases} \quad (1.10)$$

П р и м е р ы. Функции $Z = v_1 \vee v_2 \vee \dots \vee v_i \vee \dots \vee v_n$ и $Z' = v_1 v_2 \dots v_i \dots v_n$ существенно зависят от переменной v_i . Действительно, функция Z равна v_i , если $v_j = \sigma_j = 0$ ($j \neq i$), а функция Z' равна v_i , если $v_j = \sigma_j = 1$ ($j \neq i$).

Функция $f(v_1, v_2, v_3, v_4) = \overline{v_1 \vee v_2 \vee v_3 \vee v_4}$ также существенно зависит от каждой из переменных. Для отыскания тех фиксированных значений остальных переменных, при которых для рассматриваемой переменной имеет место соотношение (1.10), проще всего рассмотреть таблицу значений функции $f(v_1, v_2, v_3, v_4)$ (см. табл. I.13).

Из таблицы видно, что для переменной v_1 существует всего один набор фиксированных значений остальных переменных, а именно $\sigma_2 = \sigma_3 = \sigma_4 = 1$, при котором

$$f(v_1, \sigma_2, \sigma_3, \sigma_4) = \bar{v}_1.$$

Таблица I.13

v_1	v_2	v_3	v_4	$\overline{v_1 \vee v_2 \vee v_3 \vee v_4}$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Для переменной v_2 соотношение (1.10) выполняется при $\sigma_1 = \sigma_3 = \sigma_4 = 1$, причем оказывается, что

$$f(\sigma_1, v_2, \sigma_3, \sigma_4) = \bar{v}_2.$$

Для переменной v_3 можно подобрать три набора фиксированных значений остальных переменных, при которых выполняются соотношения (1.10). Таковыми являются наборы:

$$\sigma_1 = \sigma_2 = 0, \sigma_4 = 1; \quad \sigma_1 = 0, \quad \sigma_2 = \sigma_4 = 1 \quad \text{и} \quad \sigma_1 = \sigma_4 = 1, \quad \sigma_2 = 0.$$

Во всех этих случаях $f(\sigma_1, \sigma_2, v_3, \sigma_4) = v_3$. Наконец, для переменной v_4 можно подобрать пять разных наборов фиксированных значений остальных переменных, при которых выполняются соотношения (1.10). Таким является, например, набор: $\sigma_1 = \sigma_2 = 1, \sigma_3 = 0$, при котором

$$f(\sigma_1, \sigma_2, \sigma_3, v_4) = \bar{v}_4.$$

Это можно представить в виде таблицы I.14.

Таблица I.14

	v_1	v_2	v_3	v_4	$\bar{v}_1 \vee \bar{v}_2 \vee \bar{v}_3 \vee \bar{v}_4$
v_1	1	1	1	1	\bar{v}_1
v_2	1	1	1	1	\bar{v}_2
v_3	0	0	1	1	v_3
v_3	0	1	1	1	v_3
v_3	1	0	1	1	v_3
v_4	0	0	0	1	\bar{v}_4
v_4	0	1	0	1	\bar{v}_4
v_4	1	0	0	1	\bar{v}_4
v_4	1	1	0	1	\bar{v}_4
v_4	1	1	1	1	\bar{v}_4

Следовательно, соотношения (1.12) запишутся так ($0 \leq j < 4$):

$$F(w_0, c_{01}, c_{02}, c_{03}) = w_0 \oplus c_{00},$$

$$F(c_{10}, w_1, c_{12}, c_{13}) = w_1 \oplus c_{11},$$

$$F(c_{20}, c_{21}, w_2, c_{23}) = w_2 \oplus c_{22},$$

$$F(c_{30}, c_{31}, c_{32}, w_3) = w_3 \oplus c_{33}.$$

Из характера функции F (дизъюнкция) ясно, что эти соотношения выполняются при всех $c_{ji} = 0$.

Для отыскания сопряженных функций в соответствии с соотношением (1.13) составим таблицу (см. табл. I.15), имея в виду, что

$$\psi_i[\sigma_1(j), \sigma_2(j), y] = 0 \quad (\text{при } j \neq i),$$

$$\psi_i[\sigma_1(j), \sigma_2(j), y] = y \oplus 0 \quad (\text{при } j = i)$$

(j — номера аргументов нашей функции $F(w_1, w_2, w_3, w_4)$, а i — номера сопряженных функций $\psi_0, \psi_1, \psi_2, \psi_3$; упорядоченные пары значений x_1, x_2 : 00, 01, 10 и 11 рассматриваются как двоичная запись чисел j).

Таблица I.15

x_1	x_2	y	ψ_0	ψ_1	ψ_2	ψ_3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

Следовательно,

$$\psi_0 = \bar{x}_1 \bar{x}_2 y, \quad \psi_2 = x_1 \bar{x}_2 y,$$

$$\psi_1 = \bar{x}_1 x_2 y, \quad \psi_3 = x_1 x_2 y,$$

и в соответствии с (1.11) разложение функции $f(x_1, x_2, y)$

запишется в следующем виде:

$$f(x_1, x_2, y) = \bar{x}_1 \bar{x}_2 f(0, 0, y) \vee \bar{x}_1 x_2 f(0, 1, y) \vee \\ \vee x_1 \bar{x}_2 f(1, 0, y) \vee x_1 x_2 f(1, 1, y).$$

Оно совпадает с разложением, полученным в § 2 на основании соотношения (1.1).

2. Разложить функцию $f(x_1, x_2, y)$ по переменным (x_1, x_2) , принимая в качестве внешней функции функцию

$$F(w_0, w_1, w_2, w_3) = w_0 w_1 w_2 w_3.$$

В этом случае соотношения (1.12) выполняются при константах $c_{ji} = 1$, а сопряженные функции определяются зависимостями

$$\begin{aligned} \psi_i[\sigma_1(j), \sigma_2(j), y] &= 1 & (\text{при } j \neq i), \\ \psi_i[\sigma_1(j), \sigma_2(j), y] &= y \oplus 1 & (\text{при } j = i). \end{aligned}$$

Таблица, составленная по этим формулам, имеет следующий вид (табл. I.16):

Таблица I.16

x_1	x_2	y	ψ_0	ψ_1	ψ_2	ψ_3
0	0	0	1	1	1	1
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	0	1	1
1	0	0	1	1	1	1
1	0	1	1	1	0	1
1	1	0	1	1	1	1
1	1	1	1	1	1	0

Из таблицы находим:

$$\psi_0 = \overline{x_1 x_2 y} = x_1 \vee x_2 \vee \bar{y},$$

$$\psi_1 = \overline{x_1 x_2 y} = x_1 \vee \bar{x}_2 \vee y,$$

$$\psi_2 = \bar{x}_1 \vee x_2 \vee \bar{y},$$

$$\psi_3 = \bar{x}_1 \vee \bar{x}_2 \vee \bar{y}.$$

Следовательно, искомое разложение функции $f(x_1, x_2, y)$ запишется так:

$$f(x_1, x_2, y) = [\bar{x}_1 \vee \bar{x}_2 \vee \bar{f}(0, 0, y)] \& [\bar{x}_1 \vee x_2 \vee \bar{f}(0, 1, y)] \& [x_1 \vee \bar{x}_2 \vee \bar{f}(1, 0, y)] \& [x_1 \vee x_2 \vee \bar{f}(1, 1, y)].$$

3. Разложить функцию $f(x_1, x_2, y)$ предыдущего примера по переменным (x_1, x_2) , принимая в качестве внешней функции функцию

$$F(\omega_0, \omega_1, \omega_2, \omega_3) = \overline{\omega_0 \vee \omega_1 \vee \omega_2 \vee \omega_3}.$$

В рассмотренном ранее примере (см. стр. 60—61) были найдены фиксированные значения c_{ji} ($j \neq i$), при которых выполняются соотношения (1.12) (см. табл. I.14). Принимая во внимание эти результаты, составим таблицу, определяющую сопряженные функции (табл. I.17).

Таблица I.17

x_1	x_2	y	ψ_0	ψ_1	ψ_2	ψ_3
0	0	0	1	1	1	1
0	0	1	0	1	1	1
0	1	0	1	1	0	1
0	1	1	1	0	0	1
1	0	0	1	1	0	0
1	0	1	1	1	1	0
1	1	0	1	1	1	0
1	1	1	1	1	1	1

Из таблицы находим:

$$\psi_0 = \overline{x_1 x_2 y},$$

$$\psi_1 = \overline{x_1 x_2 y},$$

$$\psi_2 = \overline{x_1 x_2 \vee x_1 x_2 y},$$

$$\psi_3 = \overline{x_1 x_2 \vee x_1 x_2 y}.$$

Приведем теперь доказательство сформулированной выше теоремы. Рассмотрим произвольный набор $(\sigma_1, \sigma_2, \dots, \sigma_k)$. Он является двоичной записью числа $j = j(\sigma_1, \sigma_2, \dots, \sigma_k)$

($0 \leq j < 2^k$). Тогда из (1.12) и (1.13) следует, что

$$F\{\psi_0[\sigma_1(j), \sigma_2(j), \dots, \sigma_k(j), f(0, 0, \dots, 0, 0, y_1, y_2, \dots, y_l)], \dots, \psi_{2^k-1}[\sigma_1(j), \sigma_2(j), \dots, \sigma_k(j), f(1, 1, \dots, 1, 1, y_1, y_2, \dots, y_l)]\} = \\ = F\{c_{j_0}, \dots, c_{j_{j-1}}, f[\sigma_1(j), \sigma_2(j), \dots, \sigma_k(j), y_1, y_2, \dots, y_l] \oplus \\ \oplus c_{j_j}, c_{j_{j+1}}, \dots, c_{j_{j_{n-1}}}\} = f[\sigma_1(j), \sigma_2(j), \dots, \sigma_k(j), y_1, y_2, \dots, y_l],$$

и, следовательно, теорема доказана.

Полная формулировка теоремы О. Б. Лупанова охватывает общий случай разложения функции алгебры логики по k переменным, когда внешняя функция $F(w_0, w_1, \dots, w_{N-1})$ является функцией $N \geq 2^k$ переменных. В этом случае любая функция алгебры логики $f(x_1, x_2, \dots, x_k, y_1, \dots, y_l)$ может быть представлена в следующем виде:

$$f(x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_l) = \\ = F\{\psi_0[x_1, x_2, \dots, x_k, f(0, 0, \dots, 0, 0, y_1, y_2, \dots, y_l)], \dots, \psi_{2^k-1}[x_1, x_2, \dots, x_k, f(1, 1, \dots, 1, 1, y_1, y_2, \dots, y_l)], \\ \chi_{2^k}(y_1, y_2, \dots, y_l), \dots, \chi_{N-1}(y_1, y_2, \dots, y_l)\}. \quad (1.14)$$

Здесь сопряженные функции ψ_i определяются теми же соотношениями (1.13), а функции

$$\chi_i[\sigma_1(j), \sigma_2(j), \dots, \sigma_k(j)] = c_{ji} \quad (2^k \leq i < N). \quad (1.15)$$

§ 5. Элементы логики предикатов

1) Предикаты и кванторы

Обычно, когда функция алгебры логики $f(x_1, \dots, x_n)$ задана посредством словесного описания, удается легко представить ее в виде формулы, построенной с участием элементарных функций $x_1 \& x_2$, $x_1 \vee x_2$, \bar{x} в силу их соответствия логическим операциям «и», «или»^{*}), «не». Именно по этой причине указанные формулы, которые называются формулами *исчисления суждений*, приспособлены к выражению логических условий, встречающихся в словесных формулировках.

^{*}) Напомним, что «или» понимается здесь не в строго раздельном, а в соединительно-разделительном смысле.

Пример 1. Пусть функция $f(x_1, x_2, x_3)$ равна 1 при условии, что x_1 или x_2 (т. е. что x_1 равен 1 или x_2 равен 1) и не- x_3 (т. е. x_3 не равно 1). Тогда она задается формулой

$$(x_1 \vee x_2) \& \bar{x}_3.$$

Разумеется, операции исчисления суждений можно применять и к таким двоичным переменным, которые сами являются функциями от любого числа аргументов, причем эти аргументы пробегает значения из произвольного конечного или бесконечного множества \mathfrak{M} . Такие функции в логике называют *предикатами* в *предметной области* \mathfrak{M} . В данной книге типичной будет ситуация, когда в качестве предметной области взято множество натуральных чисел: 1, 2, 3, ..., интерпретируемых как дискретные значения времени.

Пример 2. Пусть равенство единице предиката $F(x, y)$ означает $x < y$; равенство единице предиката $r(x)$ означает « x четно», равенство единице предиката $Q(x)$ означает « x есть простое число». Рассмотрим следующую словесную формулировку, определяющую предикат $R(x, y)$: « $R(x, y)$ (равно 1) при условии, что x меньше y и той же четности, что y , и (кроме того) x простое число или y не простое число». Отмеченная выше близость языка исчисления суждений к обычному языку позволяет в этом случае перейти к формуле

$$F(x, y) \& [(r(x) \& r(y)) \vee (\bar{r}(x) \& \bar{r}(y))] \& [Q(x) \vee \bar{Q}(y)],$$

выражающей предикат $R(x, y)$.

Однако язык исчисления суждений оказывается недостаточным для выражения более сложных логических условий.

Пример 3. Пусть, например, предикаты $S(x, y)$ и $S'(x, y)$ определяются условиями:

« $S(x, y)$ (равно 1) при условии, что:

x меньше y и той же четности, что y , и (кроме того) существует такое z между x, y , которое просто.

$S'(x, y)$ (равно 1) при условии, что:

x меньше y и той же четности, что y , и (кроме того) всякое z , которое лежит между x, y , не просто».

Эти условия уже не удается выразить в терминах логики суждений, в которой отсутствуют операции, выражающие

суждения существования (ибо в определении предиката $S(x, y)$ содержится суждение о существовании) или суждения всеобщности (ибо в определении предиката $S'(x, y)$ содержится суждение всеобщности). Такие операции, называемые *кванторами* (кванторами существования и общности), вводятся в *логику предикатов* и позволяют строить язык формул, — так называемое *исчисление предикатов*, — который существенно богаче исчисления суждений. В этом параграфе даются краткие сведения об исчислении предикатов, которые будут в основном использованы в главе V.

Предикат $F(x)$, определенный на предметной области M , выделяет подмножество M_F предметной области, состоящее из тех элементов, для которых $F(x)$ равно 1. В этом смысле он задает *свойство* F некоторых элементов из M ; а именно свойство принадлежать множеству M_F . В математической логике запись $F(x)$ интерпретируется как обозначение суждения « x обладает свойством F », причем равенство $F(x)$ единице интерпретируется как истинность этого суждения, а равенство $F(x)$ нулю — как его ложность. Точно так же предикат $f(x_1, \dots, x_n)$ задает n -членное отношение для элементов из M и запись $f(x_1, \dots, x_n)$ может быть интерпретирована как обозначение суждения « x_1, x_2, \dots, x_n находится между собою в отношении f ». Предикат от n переменных называется n -местным предикатом.

Пусть $F(x)$ — одноместный предикат. *Квантор общности*, примененный к этому предикату, выражает суждение «для всех x (из предметной области M) имеет место F », что записывается так: $(x) F(x)$. Аналогично, *квантор существования*, примененный к предикату $F(x)$, выражает суждение «существует такое x (в области M), для которого имеет место F », и записывается так: $(\exists x) F(x)$. Таким образом, применение квантора общности к одноместному предикату, зависящему от одной переменной, дает константу, равную единицу (порождает истинное суждение), если $F(x)$ тождественно равен 1, и константу, равную 0 (ложное суждение), если $F(x)$ не равен тождественно единице. Применение же квантора существования дает константу, равную 0 (ложное суждение), если $F(x)$ тождественно равен 0, и единицу (истинное суждение), если $F(x)$ не равен тождественно нулю. Так, в примере 2 суждение $(\exists x) r(x)$ истинно (ибо четные натуральные числа действительно

существуют), в то время как $(\exists x) F(x, x)$ ложно (здесь $F(x, x)$ — одноместный предикат « x меньше самого себя»).

Квантор общности можно рассматривать как обобщение конъюнкции. Если предметная область M конечна и состоит из предметов

$$a_1, a_2, \dots, a_n,$$

то формула $(x) F(x)$ равносильна конъюнкции

$$F(a_1) \& F(a_2) \& \dots \& F(a_n),$$

а формула $(\exists x) F(x)$ равносильна дизъюнкции

$$F(a_1) \vee F(a_2) \vee \dots \vee F(a_n).$$

Для бесконечных предметных областей кванторы играют роль бесконечных конъюнкций и дизъюнкций. Здесь напрашивается аналогия с переходом в математическом анализе от конечной суммы $f_1 + f_2 + \dots + f_k$ к ряду $\sum_{x=1}^{\infty} f_x$ или

к интегралу $\int_a^b f(x) dx$. В частности, подобно тому как

в интеграле допустимо переименование переменной интегрирования, можно было бы и в нашем случае вместо $(\exists x) F(x)$ брать $(\exists y) F(y)$ или $(\exists z) F(z)$. В принятой символике предметная переменная, которая отнесена квантору, называется *связанной переменной* (аналог переменной интегриации), в отличие от других переменных, которые называются *свободными*.

Кванторы можно применять и к многоместным предикатам. Однократное применение квантора к одной из n переменных n -местного предиката порождает $(n-1)$ -местный предикат, подобно тому как однократное интегрирование функции n переменных порождает функцию $n-1$ переменных:

$$\int_a^b f(x, y, z) dx = R(y, z).$$

Пусть, например, мы имеем двуместный предикат $F(x, y)$ из примера 2. Тогда $(\exists x) F(x, y)$ задает одноместный предикат, зависящий от свободной переменной y , истинный при $y=2, 3, 4, \dots$ и ложный при $y=1$,

Путем применения операций логики суждений, кванторов и переименования предметных переменных можно из каких-то заданных предикатов образовывать другие предикаты. Состав операций, применяемых для этой цели, и порядок их выполнения отражены в формулах исчисления предикатов. Воспользуемся этими соображениями для формальной записи определений предикатов $S(x, y)$ и $S'(x, y)$ из примера 2. Очевидно:

$$\left. \begin{aligned} S(x, y) &=_{Df} F(x, y) \& [(r(x) \& r(y)) \vee (\bar{r}(x) \& \bar{r}(y))] \& \\ &\& (Ez) [F(x, z) \& F(z, y) \& Q(z)], \\ S'(x, y) &=_{Df} F(x, y) \& [(r(x) \& r(y)) \vee (\bar{r}(x) \& \bar{r}(y))] \& \\ &\& (z) [F(x, z) \& F(z, y) \rightarrow Q(z)]. \end{aligned} \right\} \quad (1.16)$$

В каждой из этих формул x, y являются свободными переменными, а z — связанной переменной.

Приведем ряд других примеров, иллюстрирующих «выразительность» и удобство этого языка формул.

Пусть $S(x, y, z)$ — трехместный предикат, означающий « $x + y = z$ », а $\Pi(x, y, z)$ — предикат, который означает « $x \cdot y = z$ ». Покажем, как через эти предикаты выражаются некоторые другие предикаты.

Двуместный предикат $F(x, y)$ (« x меньше y ») определяется условием: «существует такое натуральное число u , что $x + u = y$ ». Соответствующая формула имеет вид

$$(Eu) S(x, u, y).$$

Предикат $C(x)$, означающий « x есть составное число», определяется условием «существуют такие числа, меньшие x , произведение которых равно x »:

$$(Ev)(Ew) \{F(v, x) \& F(w, x) \& \Pi(v, w, x)\}.$$

Если учесть найденное выше выражение для $F(x, y)$, то для предиката $C(x)$ мы получим формулу

$$(Ev)(Ew) \{(Eu) S(v, u, x) \& (Eu) S(w, u, x) \& \Pi(v, w, x)\}. \quad (1.17)$$

Предикат $\Pi(x)$, означающий « x есть простое число», может быть получен отрицанием предыдущей формулы.

Предикат $H_r(x)$, означающий « x — нечетное число»,

может быть охарактеризован условием «для всякого y x не равно $y+y$ »:

$$(y) \overline{S(y, y, x)}.$$

Предикат же $r(x)$ (« x — четное число») выражается формулой

$$(E y) S(y, y, x). \quad (1.18)$$

Для формулы, в которой встречаются предикатные символы F, G, \dots, H и свободные переменные x_1, x_2, \dots, x_m , введем обозначение $\mathfrak{A}[F, G, \dots, H, x_1, \dots, x_m]$. При записи предыдущих формул мы имели в виду конкретную предметную область (натуральный ряд чисел) и вполне определенную интерпретацию предикатных символов, встречающихся в формуле. При этих условиях формула $\mathfrak{A}[F, G, \dots, H, x_1, \dots, x_m]$ задает однозначным образом m -местный предикат, зависящий от свободных переменных x_1, x_2, \dots, x_m . Если же предикатные символы рассматривать как переменные, вместо которых можно представлять любые предикаты, определенные на произвольной предметной области, то указанная формула будет задавать оператор T , перерабатывающий систему переменных предикатов F, G, \dots, H в m -местный переменный предикат

$$A(x_1, \dots, x_m) = T[F, G, \dots, H].$$

Пример 4. Обозначим формулу (1.18) через $\mathfrak{A}[S, x]$ и рассмотрим, кроме прежней интерпретации символа $S(x, y, z)$ как « $x+y=z$ », и следующую интерпретацию: « $x \cdot y = z$ ». Тогда формула $\mathfrak{A}[S, x]$ породит вместо прежнего одноместного предиката « x есть четное число» предикат « x есть точный квадрат».

Благодаря своей близости к обычному языку формальный язык логики предикатов удобно применять для задания операторов, перерабатывающих предикаты (или системы предикатов) в предикаты (подобно тому, как формулы алгебры логики удобно применять для задания двоичных функций от двоичных переменных).

Не всякое выражение, построенное из символов, принятых для обозначения предикатных и предметных переменных, операций логики суждений и кванторов, может быть рассматриваемо как формула исчисления предика-

тов. Предварительно можно пояснить, что под формулой мы понимаем выражение, построенное осмысленным образом из указанных символов. Так, например, выражение

$$F(x) \& (y) (Ez)$$

явно бессмысленно. Однако целесообразно исключить из рассмотрения и такие выражения, как $(x) F(x) \& (Ey) H(x, y)$, которые, хотя и осмысленны, обладают тем недостатком, что в них одна и та же переменная (в данном случае x) имеет как свободное, так и связанное вхождение. Мы могли бы переименовать связанную переменную x в другую переменную, скажем в z , и тем самым устранить эту ситуацию, которая при дальнейших рассмотрениях могла бы породить путаницу из-за смешения переменных разных сортов. Можно дать чисто формальное определение понятия «формула исчисления предикатов» путем четкой регламентации способов ее построения из исходного запаса символов, обозначающих предикатные переменные (прописные буквы, за которыми в скобках указаны в надлежащем числе пустые места для аргументов $F()$, $G(,)$, $H(, ,)$ и т. п.), предметные переменные (малые буквы x, y, z , и т. п.), знаки логических операций. Одновременно определяются понятия свободных и связанных вхождений переменных.

2) Индуктивное определение формулы исчисления предикатов

1. Каждый предикатный символ, все пустые места которого замещены предметными переменными, есть формула; все вхождения этих переменных в формулу являются свободными*). Формулами также считаются знаки *И* (логическая константа «истина», иногда обозначаемая также символом 1) и *Л* (логическая константа «ложь», иногда обозначаемая также символом 0).

2. Пусть $\Phi(A_1, A_2, \dots, A_h)$ — произвольная формула исчисления высказываний; результат подстановки вместо переменных (двоичных) A_1, A_2, \dots, A_h любых формул $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_h$ будет также формулой при следующем условии: если какая-либо предметная переменная имеет свободное вхождение в одной из формул \mathcal{A}_i , то она не имеет связанных

*) Формулы этого вида называются атомарными.

вхождений ни в одной из остальных. В результирующей формуле (обозначим ее через $\Phi [A_1, A_2, \dots, A_k]$) вхождения предметных переменных являются свободными или связанными, если они были таковыми в формулах A_1, A_2, \dots, A_k .

3. Если $A [...x_i...]$ есть формула, в которую переменная x_i входит свободно, то $(x_i) (A [...x_i...])$ и $(Ex_i) (A [...x_i...])$ тоже являются формулами. В них переменная x_i является связанной, а остальные переменные имеют такие же вхождения, какие имели в A .

4. Других формул в логике предикатов не существует.

П р и м е ч а н и е. Пункт 3 этого определения предписывает брать в скобках область действия квантора. В некоторых случаях там, где это не вызывает недоразумений, скобки могут быть опущены; это пояснено в пунктах б) и д) приводимого ниже примера.

Если в условиях пункта 3 формула A содержит кванторы, то каждый из них считается подчиненным квантору (Ex_i) или квантору (x_i) , действующему на A . Это позволяет определить индуктивно понятие *ранга квантора*:

1°. Квантор имеет ранг 1, если ему не подчинены другие кванторы.

2°. Квантор имеет ранг $k+1$, если наивысший из рангов подчиненных ему кванторов равен k . Кванторной глубиной формулы называется наивысший из рангов ее кванторов.

Приведенные определения отражают процесс построения формулы путем «заготовки» все более обширных ее подформул (т. е. таких частей, которые сами тоже являются формулами). Проиллюстрируем это на примере формулы (1.17).

Для построения этой формулы строятся предварительно подформулы:

а) $S(v, u, x)$, $S(w, u, x)$, $\Pi(v, w, x)$ (ср. п. 1 индуктивного определения формулы исчисления предикатов);

б) $(Eu) S(v, u, x)$, $(Eu) S(w, u, x)$ (ср. п. 3 индуктивного определения формулы; скобки, охватывающие область действия квантора, мы опускаем, ибо в данном случае это не вызывает недоразумений);

в) $(Eu) S(v, u, x) \& (Eu) S(w, u, x) \& \Pi(v, w, x)$ (ср. п. 2 определения: подстановка в формулу алгебры логики $A_1 \& A_2 \& A_3$);

- г) $(Ew) [(Eu) S(v, u, x) \& (Eu) S(w, u, x) \& \Pi(v, w, x)]$
 (п. 3 определения);
 д) наконец, получается сама формула (1.17):

$$(Ev) (Ew) \{ (Eu) S(v, u, x) \& (Eu) S(w, u, x) \& \Pi(v, w, x) \}$$

(п. 3 определения; область действия квантора (Ev) взята без скобок).

В построенной формуле каждый из кванторов (Eu) имеет ранг 1, квантор (Ew) — ранг 2, квантор (Ev) — ранг 3. Кванторная глубина формулы равна 3.

3) Основные тождественные преобразования в исчислении предикатов

До сих пор мы рассматривали исчисление предикатов с точки зрения тех возможностей, которые оно предоставляет для символической записи сложных суждений и предикатов; эту роль, выполняемую исчислением, можно было бы называть коммуникативной. Вместе с тем большое значение имеет тот факт, что над формулами исчисления предикатов можно осуществлять тождественные преобразования в соответствии с простыми и обозримыми правилами; в этом заключается оперативная функция, выполняемая исчислением.

Пусть формулы $\mathcal{A}_1 [F, G, \dots, H, x_1, \dots, x_m]$ и $\mathcal{A}_2 [F, G, \dots, H, x_1, \dots, x_m]$ задают один и тот же оператор, перерабатывающий систему предикатов F, G, \dots, H в предикат от аргументов x_1, \dots, x_m ; в таком случае будем говорить, что \mathcal{A}_1 и \mathcal{A}_2 эквивалентны и обозначать это так: \mathcal{A}_1 экв. \mathcal{A}_2 .

П р и м е р 5. Следующие формулы попарно эквивалентны:

$$(z) H(z) \& \overline{G(x, x) \& (Ey) F(x, y)}, \quad (1.19a)$$

$$(z) H(z) \& \overline{(G(x, x) \vee (Ey) F(x, y))}, \quad (1.19б)$$

$$((z) H(z) \& \overline{G(x, x)}) \vee ((z) H(z) \& \overline{(Ey) F(x, y)}). \quad (1.19в)$$

Это легко усматривается из эквивалентности формул алгебры логики: $A_1 \& \overline{A_2 \& A_3}$, $A_1 \& (\overline{A_2} \vee \overline{A_3})$, $(A_1 \& \overline{A_2}) \vee (A_1 \& \overline{A_3})$.

Переход от данной формулы исчисления предикатов к эквивалентной ей формуле мы будем называть, как и в

исчислении суждений, *тождественным преобразованием*. В предыдущем примере были фактически использованы тождественные преобразования, установленные еще ранее для операций алгебры логики. Разумеется, все те тождественные преобразования, которые вытекают из свойств операций алгебры логики и не затрагивают специфических свойств кванторов, применимы и к формулам исчисления предикатов. Особого рассмотрения требуют те преобразования, которые основаны на специфических свойствах кванторов. Их формулировка и обоснование облегчаются аналогией, подмеченной еще ранее, между кванторами общности и конъюнкцией, а также между квантором существования и дизъюнкцией. Из самого смысла кванторов общности и существования непосредственно вытекают следующие эквивалентности.

I. Эквивалентности с отрицанием:

$$(Ex) A(x) \text{ экв. } \overline{(x) \overline{A(x)}}, \quad (Ia)$$

$$(x) A(x) \text{ экв. } \overline{(Ex) \overline{A(x)}}, \quad (Iб)$$

на основании которых можно заменять квантор общности квантором существования и наоборот. Эти эквивалентности являются аналогами установленных ранее:

$$A_1 \vee A_2 \text{ экв. } \overline{\overline{A_1} \& \overline{A_2}},$$

$$A_1 \& A_2 \text{ экв. } \overline{\overline{A_1} \vee \overline{A_2}}.$$

Применяя (Ia) к подформуле $\overline{(Ey) F(x, y)}$ формулы (1.19б) и учитывая, что $\overline{\overline{A}}$ экв. A , преобразуем (1.19б) к виду

$$(z) H(z) \& \overline{(G(x, x) \vee (y) F(x, y))}.$$

II. Эквивалентности для II-операций (конъюнкция и квантор общности):

$$(x) (A_1(x) \& A_2(x)) \text{ экв. } (x) A_1(x) \& (x) A_2(x), \quad (IIa)$$

$$(x)(y) A(x, y) \text{ экв. } (y)(x) A(x, y). \quad (IIб)$$

Мы будем преимущественно пользоваться соотношением (IIa) для замены левой его части правой.

III. Эквивалентности для Σ -операций (дизъюнкция и квантор существования):

Они аналогичны соотношениям предыдущей группы:
 $(\exists x)(A_1(x) \vee A_2(x))$ экв. $(\exists x) A_1(x) \vee (\exists x) A_2(x)$, (IIIa)

$(\exists x)(\exists y) A(x, y)$ экв. $(\exists y)(\exists x) A(x, y)$. (IIIб)

Заметим, что эквивалентности (IIa) и (IIIa) естественным образом распространяются на любое число конъюнкций (соответственно дизъюнкций).

Эквивалентности групп II и III указывают на определенную перестановочность однотипных операций, т. е. на перестановочность квантора общности с конъюнкцией, квантора общности с квантором общности (и аналогично для Σ -операций). Эта закономерность не распространяется на тот случай, когда в формуле участвуют разнотипные операции (например, квантор общности и дизъюнкция). Приведенные ниже пары формул не являются эквивалентными:

$$(x)[A_1(x) \vee A_2(x)], \quad (x) A_1(x) \vee (x) A_2(x), \quad (1.20a)$$

$$(\exists x) A_1(x) \& (\exists x) A_2(x), \quad (\exists x)(A_1(x) \& A_2(x)), \quad (1.20б)$$

$$(x)(\exists y) A(x, y), \quad (\exists y)(x) A(x, y). \quad (1.20в)$$

Так, например, если $A_1(x)$ есть предикат « x есть четное число», а $A_2(x)$ есть предикат « x есть нечетное число», то первая из формул (1.20б) истинна, в то время как вторая ложна. Вообще, каждое из утверждений, символически записанных во втором столбце, сильнее, чем соответствующее утверждение из первого столбца, т. е. из истинности каждого из утверждений правого столбца вытекает истинность соответствующего ему утверждения из левого столбца, но не наоборот.

IV. Эквивалентности с относительной константой. Пусть в области действия квантора (x) или $(\exists x)$ имеется подформула \mathfrak{B} , не содержащая предметной переменной x , которая связана этим квантором; в таком случае будем говорить, что \mathfrak{B} является *константой относительно данного квантора*. Эквивалентности четвертой группы устанавливают случаи, когда подобную константу можно вынести из-под знака квантора (или же внести в область действия квантора); здесь имеется

аналогия с выносом константы за знак интеграла. Пусть $[x]$ обозначает любой из двух возможных кванторов (общности или существования), связывающих переменную x , а знак \odot — любую из двух операций, конъюнкцию или дизъюнкцию. Эквивалентности группы IV имеют вид

$$[x] \{ \mathcal{A} \odot \mathcal{B}(x) \} \text{ экв. } \mathcal{A} \odot [x] \mathcal{B}(x), \quad (\text{IV})$$

где \mathcal{A} обозначает подформулу, не содержащую переменную x . Конкретизируя смысл операций $[x]$ и \odot , мы получаем четыре эквивалентности:

$$(x) \{ \mathcal{A} \& \mathcal{B}(x) \} \text{ экв. } \mathcal{A} \& (x) \mathcal{B}(x), \quad (\text{IVa})$$

$$(x) \{ \mathcal{A} \vee \mathcal{B}(x) \} \text{ экв. } \mathcal{A} \vee (x) \mathcal{B}(x), \quad (\text{IVб})$$

$$(\text{E}x) \{ \mathcal{A} \& \mathcal{B}(x) \} \text{ экв. } \mathcal{A} \& (\text{E}x) \mathcal{B}(x), \quad (\text{IVв})$$

$$(\text{E}x) \{ \mathcal{A} \vee \mathcal{B}(x) \} \text{ экв. } \mathcal{A} \vee (\text{E}x) \mathcal{B}(x). \quad (\text{IVг})$$

Соотношения (IVб) и (IVв) полезно сравнить с формулами (1.20а) и (1.20б), для которых аналогичная закономерность не проходит.

Пример 6. Дана формула

$$(\tau) \{ F(\tau) \vee X(t) \rightarrow G(\tau) \& X(t) \}.$$

Покажем, как можно вынести из-под знака квантора относительную константу $X(t)$. Для этого мы сначала приводим подкванторное выражение к конъюнктивной нормальной форме с тем, чтобы можно было воспользоваться эквивалентностями (IIa) и (IV). После замены импликации дизъюнкцией и отрицанием и применения теоремы де Моргана мы имеем:

$$(\tau) [(\overline{F(\tau)} \& \overline{X(t)}) \vee (G(\tau) \& X(t))]$$

и, наконец,

$$(\tau) [(\overline{F(\tau)} \vee G(\tau)) \& (\overline{F(\tau)} \vee X(t)) \& \\ \& (\overline{X(t)} \vee G(\tau)) \& (\overline{X(t)} \vee X(t))],$$

где конъюнктивный член $\overline{X(t)} \vee X(t)$ равен 1 (истинен) и поэтому может быть вычеркнут. Далее применение (IIa) дает:

$$(\tau) [\overline{F(\tau)} \vee G(\tau)] \& (\tau) (\overline{F(\tau)} \vee \overline{X(t)}) \& (\tau) (X(t) \vee G(\tau)),$$

после чего можно уже применить (IVб) к каждому из последних двух конъюнктивных членов. Окончательный результат имеет вид

$$(\tau)(\overline{F(\tau)} \vee G(\tau))(\&((\tau)\overline{F(\tau)} \vee X(t)) \& (\overline{X(t)} \vee (\tau)G(\tau)).$$

Совершенно аналогично выносятся относительные константы из области определения квантора существования; для этого подкванторное выражение приходится предварительно представлять в виде дизъюнкции конъюнкций. Подобные процессы выноса констант будут нами широко применяться (см. ниже, гл. V).

4) О расширенном исчислении предикатов

Рассмотренное до сих пор исчисление предикатов, благодаря применению кванторов по предметным переменным, является языком значительно более выразительным, чем язык логики суждений. Еще большее усиление «выразительности» может быть достигнуто, если допустить, наряду с кванторами по предметным переменным, также и кванторы по предикатным переменным. Возникающее таким образом исчисление называют *расширенным исчислением предикатов*, в отличие от ранее рассмотренного *узкого исчисления предикатов*.

Приведем примеры формул расширенного исчисления предикатов:

$$(F)\{F(x) \vee \overline{F}(x)\}, \quad (1.21)$$

$$(EF)\{F(x) \& (y)H(x, y)\}. \quad (1.22)$$

Формула 1.21 утверждает, что выражение $F(x) \vee \overline{F}(x)$ справедливо для всякого одноместного предиката F . В ней F является связанной предикатной переменной, а x — свободной предметной переменной. Эта формула истинна при любом x ; можно считать, что она задает одноместный предикат, зависящий от x , который является тождественно истинным. В формуле (1.22) буква H является свободной предикатной переменной, а x — свободной предметной переменной. Можно считать, что эта формула задает некоторый одноместный предикат, зависящий от x . Каков этот предикат — это в свою очередь зависит от выбора двуместного предиката H .

Можно дать индуктивное определение формулы расширенного исчисления предикатов, аналогичное определению, данному выше, на стр. 73—74. Различие заключается лишь в том, что вместо свободных и связанных вхождений предметных переменных и кванторов, действующих на такие переменные, приходится рассматривать вхождения и кванторы, относящиеся как к предметным, так и к предикатным переменным.

На предикатные кванторы распространяются эквивалентности типа (I)—(IV). В частности, применяя соответствующую модификацию эквивалентности (IVв), мы можем преобразовать (1.22) к виду

$$(EF) F(x) \& (y) H(x, y). \quad (1.22a)$$

Кроме указанных эквивалентностей, имеет место и ряд специфических соотношений, вытекающих из содержательного смысла кванторных предикатов. Например, $(EF) F(x)$ истинно (ведь можно предикат определить так, чтобы он был истинен для x). Отсюда видно, что формула (1.22) эквивалентна формуле $(y) H(x, y)$.

5) Исчисление предикатов с ограниченными кванторами

Имея в виду последующие приложения (см. гл. V), мы укажем в этом разделе некоторую модификацию языка формул исчисления предикатов — исчисление предикатов с ограниченными кванторами. Эта модификация касается интерпретации формул (см. I), а также их видов (см. II).

I. Условимся в том, что впредь при употреблении формул исчисления предикатов предметная область будет интерпретироваться как натуральный ряд чисел. Это соглашение вызвано тем, что для описания операторов, перерабатывающих системы предикатов в предикаты, мы будем пользоваться именно таким образом интерпретируемыми формулами; при этом, как это было отмечено и ранее, нас интересуют в основном одноместные предикаты, аргументы которых пробегает натуральный ряд чисел (понимаемых как дискретные значения времени).

II. При указанной интерпретации, наряду с обычными кванторами по предметам, разрешим себе пользоваться также и так называемыми *ограниченными кванторами*,

смысл и правила употребления которых явствуют из следующих пояснений.

Формула $(E\tau) X(\tau)$ с ограниченным сверху квантором существования $\tau < t$ соответствует словесному выражению «существует такое τ , меньшее t , для которого справедливо $X(\tau)$ »; формула же $(\tau) X(\tau)$ с ограниченным сверху квантором общности $\tau < t$ соответствует выражению «для всякого τ , которое меньше t , справедливо $X(\tau)$ ».

Отсюда видно, что ограниченные кванторы не представляют собой существенно новой операции и могут быть выражены обычными формулами исчисления предикатов с участием специального предиката « $x < y$ »; в наших примерах такими формулами являются:

$$(E\tau) [\tau < t \& X(\tau)], \quad (\tau) [\tau < t \rightarrow X(\tau)].$$

Однако мы предпочтем пользоваться символикой ограниченных кванторов, рассматривая также и кванторы типа $[\tau]$ и двусторонние кванторы типа $[\tau]$, где $<$ может обозначать как $<$, так и \leq .

Заметим, что применение ограниченных кванторов, вообще говоря, не уменьшает числа свободных переменных. В рассматриваемых выше примерах вместо переменной τ , связываемой квантором, появляется свободная переменная t (в ограниченном сверху кванторе) или даже пара переменных q, t (в двусторонне ограниченном кванторе).

Приведем пример более сложной формулы, записанной с ограниченными кванторами:

$$(E\tau) \{X(\tau) \& (q) [Z(q) \vee \overline{X(t)}]\}. \quad (1.23)$$

$\tau < t$ $\tau < q < t$

В этой формуле имеется единственная свободная переменная t , имеющая вхождения как граница квантора $(E\tau)$, а также как аргумент предиката $\overline{X(t)}$. В подформуле же $(q) [Z(q) \vee \overline{X(t)}]$ имеются две свободные переменные: τ и t .

Особенность формулы (1.23) заключается еще в том, что в ней встречаются лишь одноместные предикатные переменные. О таких формулах мы будем в дальнейшем говорить как о формулах *исчисления одноместных предикатов*

с ограниченными кванторами. При этом термин «узкое (или расширенное) истечение» будет применяться, как и прежде, для указания того, что в формуле (формулах), о которых идет речь в данном контексте, отсутствуют (соответственно, присутствуют) кванторы по предикатам. Так, например, формула $(E\tau) \{(\tau) \Gamma(\tau) \rightarrow X(t)\}$ является формулой расширенного исчисления одноместных предикатов с ограниченными кванторами.

Укажем еще вкратце некоторые особенности тождественных преобразований рассматриваемых формул с ограниченными кванторами. Установленные ранее эквивалентности (I)—(IV), за исключением (IIб) и (IIIб), справедливы, в частности, и для формул с ограниченными кванторами. Покажем, например, как в приведенной выше формуле (1.23) относительная константа $\overline{X(t)}$ выносится сначала за знак квантора первого ранга (τ) , а потом и за знак квантора второго ранга $(E\tau)$. Мы имеем:

$$(E\tau) [X(\tau) \& (\tau < t) (Z(\tau) \vee \overline{X(t)})] \text{ экв.}$$

$$\text{экв. } (E\tau) [X(\tau) \& ((\tau < t) Z(\tau) \vee \overline{X(t)})] \text{ экв.} \quad (IVб)$$

$$\text{экв. } (E\tau) [(X(\tau) \& (\tau < t) Z(\tau)) \vee (X(\tau) \& \overline{X(t)})] \text{ экв.}$$

$$\text{экв. } (E\tau) [X(\tau) \& (\tau < t) Z(\tau)] \vee (E\tau) [X(\tau) \& \overline{X(t)}] \text{ экв.} \quad (IIIa)$$

$$\text{экв. } (E\tau) [X(\tau) \& (\tau < t) Z(\tau)] \vee [\overline{X(t)} \& (E\tau) X(\tau)]. \quad (IVв)$$

Нарушение эквивалентностей (IIб) и (IIIб) для формул с ограниченными кванторами можно иллюстрировать на следующем примере. Формула $(E\tau) (E\sigma) X(\sigma)$ в результате применения (IIIб) переходит в формулу $(E\sigma)(E\tau) X(\sigma)$; в этой последней переменная τ имеет как свободное, так и связанное вхождение, что недопустимо.

Легко показать, что посредством формул с ограниченными предметными кванторами (но без привлечения

предикатных кванторов) выразимы предикаты:

$\tau < t$, $\tau = t$, $t = k$, $t < k$, $t = \tau - k$, $t = \tau + k$, (ж)
где t , τ — предметные переменные, k — произвольная константа. Например:

$$\tau < t = (E\sigma)(E\varrho)U, \quad t = 1 = (\tau)L.$$

$$\text{Df } \sigma < t \quad \tau \leq \varrho \leq \sigma \qquad \text{Df } \tau < t$$

Мы будем в дальнейшем пользоваться расширением языка формул с ограниченными кванторами за счет применения указанных предикатов, поскольку оно облегчает и упрощает формулировки. При этом учитывается, что имеется простая процедура элиминации этих «вторичных» предикатов.

При преобразовании формул с ограниченными кванторами мы будем пользоваться еще некоторыми соотношениями, вытекающими из специфики выбранной нами предметной области и свойств предиката порядка $x < y$:

1. $(E\tau) X(\tau)$ ложно (равно 0), ибо единице ничего не предшествует.

2. $(\tau) X(\tau)$ истинно (равно 1); это видно из равнозначной записи: $(\tau) [\tau < 1 \rightarrow X(\tau)]$, ибо в силу ложности $\tau < 1$ импликация $\tau < 1 \rightarrow X(\tau)$ истинна.

Аналогично, в случае двусторонних кванторов:

1'. $(E\tau) X(\tau)$ и $(E\tau) X(\tau)$ ложны.

2'. $(\tau) X(\tau)$ и $(\tau) X(\tau)$ истинны.

Далее:

3. $(E\tau) X(\tau)$ экв. $(E\tau) X(\tau) \vee X(t)$.

4. $(\tau) X(\tau)$ экв. $(\tau) X(\tau) \& X(t)$.

Аналогично

3'. $(E\tau) X(\tau)$ экв. $(E\tau) X(\tau) \vee (\varrho < t \& X(t))$

4'. $(\tau) X(\tau)$ экв. $(\tau) X(\tau) \& (\varrho < \tau \rightarrow X(t))$.

ГЛАВА II

ОПЕРАТОРЫ. ЛОГИЧЕСКИЕ СЕТИ

§ 1. Общие замечания

В этой главе определяется ряд понятий и формулируются некоторые основные задачи теории конечных автоматов. Понятие *конечного автомата* вводится как математическая модель устройства синхронного действия, перерабатывающего дискретную информацию. Всякая реальная физическая система, в которой происходит процесс преобразования информации, характеризуется своей внутренней конструкцией, способом функционирования и осуществляемыми операциями. В соответствии с этим понятие конечного автомата определяется как совокупность двух понятий: *схемы (логической сети)* и *оператора* (реализуемого схемой *). Формулировке определений почти везде предпосылаются содержательные рассуждения, которые призваны оправдать естественность и разумность принимаемых соглашений и формальных построений. На этой стадии изложения применяются также и такие термины, которые точно не определены и понимаются интуитивно («устройство», «состояние физической системы», «эффективное преобразование» и т. п.).

В §§ 2 и 3 определяется класс *ограниченно-детерминированных операторов* **) и приводятся доводы в пользу тезиса о том, что операторы, реализуемые в физических системах рассматриваемого типа, обязаны принадлежать

*) В абстрактной теории автоматов термин «конечный автомат» употребляется зачастую в несколько ином смысле (см. далее понятие «элементарного автомата», стр. 107).

**) Ограниченно-детерминированные операторы называются иначе автоматными операторами или автоматными соответствиями.

Пусть дана конечная занумерованная система алфавитов

$$A_1, A_2, \dots, A_k, \quad (2.3)$$

насчитывающих соответственно m_1, m_2, \dots, m_k букв, причем не исключается возможность совпадения некоторых или даже всех алфавитов этой системы. Очевидно, что существуют в точности $m_1 \cdot m_2 \cdot \dots \cdot m_k$ слов вида $\lambda_1 \lambda_2 \dots \lambda_k$, где буквы $\lambda_1, \lambda_2, \dots, \lambda_k$ пробегают независимо друг от друга алфавиты A_1, A_2, \dots, A_k . Объявим эти слова буквами нового алфавита \mathcal{A} , называемого *прямым произведением* алфавитов (2.3), а операцию образования прямого произведения обозначим так: $\mathcal{A} = A_1 \times A_2 \times \dots \times A_k$; в частности, если алфавиты (2.3) попарно совпадают, то будем говорить о *прямой степени* k алфавита A , обозначаемой A^k . Очевидно, если алфавит A состоит из m букв, то в A^k их m^k . Такой подход объясняется тем, что в дальнейшем часто удобно будет k букв, одновременно следующих по параллельным каналам (которые являются либо все входными, либо все выходными каналами), рассматривать как одну букву из прямого произведения соответствующих алфавитов, следующую по единому «склеенному» каналу *). Этот переход от алфавитов входных каналов и алфавитов выходных каналов к единому входному (соответственно выходному) алфавиту и соответствующее ему «склеивание» параллельных каналов будут применяться без особых оговорок там, где это не вызывает недоразумений. В частности, это позволяет ограничиться рассмотрением устройств с одним входным и одним выходным каналом. С другой стороны, мы будем применять и обратный процесс, а именно — представлять заданный алфавит как прямое произведение или как часть прямого произведения других алфавитов (каждый из которых насчитывает меньше букв, чем исходный алфавит). Чаще всего мы будем представлять буквы исходного алфавита в виде слов одинаковой длины в двоичном алфавите, т. е. будем прибегать к изображению исходных букв в *равномерном двоичном коде*.

*) При этом, разумеется, придется зафиксировать вполне определенный порядок нумерации этих каналов с тем, чтобы каждую систему k букв можно было бы рассматривать как слово

П р и м е р. Русский алфавит, насчитывающий 28 букв, может быть представлен двадцатью восемью словами в алфавите $(0,1)$ посредством пятизначного двоичного кода, в котором буквам

$a, б, в, \dots$

соответствуют кодовые группы:

10 000, 00110, 01101, ...

В том же двоичном алфавите цифровой десятичный алфавит может быть представлен посредством четырехзначных кодовых групп.



Рис. 2.2.

Очевидно, алфавит из k букв всегда может быть реализован в виде прямой степени S или части прямой степени S двоичного алфавита (т. е. буквы

этого алфавита могут быть закодированы кодовыми группами по S двоичных знаков в каждой), лишь бы натуральное число S удовлетворяло следующему необходимому и достаточному условию:

$$S \geq \lg_2 k. \quad (2.4)$$

Имея в виду изложенные соображения, рассмотрим устройство с одним входным и одним выходным каналом (рис. 2.2); алфавиты, соответствующие этим каналам, обозначим через A и B . Слова, преобразуемые устройством, могут быть сколь угодно большой длины. Поэтому удобно рассматривать бесконечные последовательности букв, которые могут поступать на вход:

$$f(1) f(2) \dots f(\mu) \dots \quad (2.5)$$

Здесь $f(t)$ ($t=1, 2, \dots, \mu, \dots$) является произвольной функцией от натурального аргумента, которая принимает значения из заданного алфавита A . Множество всех таких функций (т. е. множество всех возможных последовательностей букв из алфавита A) обозначим через A^1 . Подача последовательности вида (2.5) по входному каналу

порождает на выходном канале некоторую последовательность

$$g(1)g(2)\dots g(\mu)\dots \quad (2.6)$$

Таким образом, каждое устройство синхронного действия реализует физически некоторый оператор θ , действующий на функции $f(t) \in A^t$ и преобразующий их в функции

$$g(t) = \theta[f(t)]; \quad g(t) \in B^t. \quad (2.7)$$

Наша ближайшая задача заключается в классификации этих операторов.

§ 2. Детерминированность операторов

Очевидно, всякое реальное устройство синхронного действия является устройством без предвосхищения. Это означает, что выходная буква $g(t)$ не зависит от входных букв $f(\tau)$ при $\tau > t$, а может зависеть лишь от буквы, поданной на вход в том же такте, т. е. от $f(t)$, и от ранее поданных входных букв, т. е. от $f(\tau)$ при $\tau < t$. Это замечание приводит нас к следующему определению.

Определение. Оператор $g(t) = \theta[f(t)]$ называется *детерминированным оператором*, или *оператором без предвосхищения*, если при любом t буква $g(t)$ является однозначной функцией от слова:

$$f(1)f(2)\dots f(t-1)f(t).$$

Возможность реализации только детерминированных операторов устройствами синхронного действия можно описать также в терминах преобразования слов (конечных) следующим образом:

а) всякое входное слово преобразуется в слово той же длины;

б) если у двух входных слов совпадают первые l букв, то и у соответствующих выходных слов совпадают первые l букв.

Приведем примеры некоторых простейших типов детерминированных операторов.

1. *Истинностный оператор (оператор без памяти).* Оператор этого вида характеризуется тем, что для

каждого t выходная буква $g(t)$ однозначным образом определяется входной буквой $f(t)$, а следовательно, не зависит ни от значений $f(\tau)$ при $\tau > t$ (условие детерминированности), ни от значения $f(\tau)$ при $\tau < t$ (отсутствие памяти). Истинностный оператор вполне определяется заданием (хотя бы в виде таблицы) функции, сопоставляющей каждой букве входного алфавита A соответствующую букву выходного алфавита. Для оператора, реализуемого триодом, эта функция такова:

Вход	Выход
0	1
1	0

Если входные буквы представляют собой наборы $\sigma_1 \sigma_2 \dots \sigma_m$ из нулей и единиц, а выходные буквы — аналогичные наборы $r_1 r_2 \dots r_n$, то истинностный оператор описывается системой n функций алгебры логики от m переменных каждая:

$$z_1 = f_1(x_1, x_2, \dots, x_m),$$

$$z_2 = f_2(x_1, x_2, \dots, x_m),$$

$$\dots \dots \dots$$

$$z_n = f_n(x_1, x_2, \dots, x_m).$$

В таком случае истинностный оператор может быть задан аналитически посредством системы формул исчисления высказываний. В частности, для триода ($m = n = 1$): $z = \bar{x}$. Допустим, что оператор задан таблицей II.1:

Таблица II.1

x_1	x_2	x_3	$\Phi_1(x_1, x_2, x_3)$	$\Phi_2(x_1, x_2, x_3)$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
1	0	0	0	1
1	0	1	0	0
1	1	0	1	0

В данном случае не все 2^3 возможных наборов из 1 и 0 использованы в качестве входных букв, и, следовательно, мы имеем дело с функциями алгебры логики, которые не всюду определены. Поскольку, как мы знаем, всякая формула исчисления высказываний задает функцию, всюду определенную, то, как отмечено в гл. I, § 3, аналитическое задание функций $\varphi_1(x_1, x_2, x_3)$ и $\varphi_2(x_1, x_2, x_3)$ предполагает доопределение таблицы путем включения в нее недостающих входных кодовых групп, которым сопоставлены определенные значения функций. Такое доопределение возможно многими способами, и, как отмечено в гл. I, надлежащим выбором этого способа удастся иногда существенно упростить аналитическое представление таблично заданного оператора. В качестве одного из возможных вариантов доопределения функций, представленных таблицей II. 1, возьмем таблицу II.2.

Таблица II.2

x_1	x_2	x_3	z_1	z_2
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	(1)	(0)
1	0	0	0	1
1	0	1	0	0
1	1	0	1	0
1	1	1	(0)	(0)

В этой таблице в скобках указаны доопределенные значения. Этой таблице соответствуют две функции алгебры логики, задаваемые формулами

$$z_1 = \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3,$$

$$z_2 = \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3.$$

После известных читателю преобразований алгебры логики мы получаем:

$$z_1 = \bar{x}_1 x_3 \vee x_2 \bar{x}_3,$$

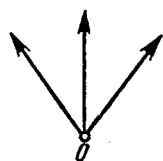
$$z_2 = \bar{x}_1 (x_2 \vee x_3) \vee x_1 (\overline{x_2 \vee x_3}).$$

2. *Константный оператор.* Оператор этого вида характеризуется тем, что любую входную последовательность $f(1) f(2) \dots f(t) \dots$ он перерабатывает в одну и ту же фиксированную выходную последовательность $g(1) g(2) \dots g(t) \dots$. Очевидно, что для любого t выходная буква зависит только от аргумента t и уж во всяком случае не зависит от значений $f(\tau)$ при $\tau > t$ (условие детерминированности). Константный оператор вполне определяется заданием фиксированной выходной последовательности $g(1) g(2) \dots g(t) \dots$, о которой мы будем говорить, что она порождается константным оператором.

Можно считать, что физическими носителями константных операторов являются, в частности, так называемые *автономные устройства*, т. е. устройства, не имеющие входов и вырабатывающие выходную информацию за счет внутренних процессов. Правда, напрашивается мысль рассматривать такие устройства как устройства, реализующие оператор с пустым входным алфавитом, который действует на пустое множество входных последовательностей. Однако математической моделью может служить здесь и константный оператор, у которого входной алфавит состоит из единственной буквы.

В дальнейшем изложении удобно пользоваться графическим изображением детерминированных операторов посредством *деревьев*, к описанию которых мы и переходим.

Рис. 2.3.



Построим вначале для заданного алфавита A с m буквами *информативное бесконечное дерево* следующим образом.

Зафиксируем точку O (рис. 2.3), называемую *вершиной первого ранга*, или *корнем дерева*, и проведем из нее m отрезков, называемых *ребрами первого ранга*. Эти ребра, взятые в некотором порядке — для определенности мы всюду назначаем порядок по часовой стрелке, — будут изображать соответственно буквы $0, 1, \dots, m-1$ (на рис. 2.3 $m=3$; левый отрезок изображает букву 0, средний отрезок изображает букву 1, а правый — букву 2).

Концы отрезков первого ранга, отличные от вершины первого ранга, объявляются *вершинами второго ранга*; из каждой вершины второго ранга проводятся также m

щих входные последовательности букв в алфавите A в выходные последовательности букв в алфавите B .

Исходя из дерева D_A , входного алфавита A , построим *нагруженное дерево* D_{A-B} , приписывая произвольным образом всем ребрам дерева D_A буквы из алфавита B . Каждому пути в дереве D_A соответствует тогда последовательность букв из алфавита B , и тем самым индуцируется оператор, перерабатывающий последовательности букв из алфавита A в последовательности букв из алфавита B . Так, например, в соответствии с нагруженным деревом, изображенным на рис. 2.5, входная последовательность

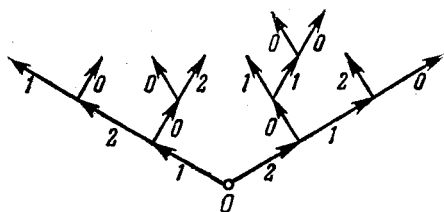


Рис. 2.5.

1011... (в двоичном алфавите) перерабатывается в выходную последовательность 2010... (в троичном алфавите).

Очевидно, что всякий оператор $g(t) = \theta[f(t)]$, задаваемый посредством нагруженного дерева, является

детерминированным. Действительно, если две входные последовательности $f(t)$ и $f'(t)$ совпадают при всех $\tau \leq t$, то и соответствующие им пути вместе с несомыми ими буквами алфавита B совпадают на нижних ярусах ранга $\leq t$. Поэтому последовательности $g(t)$ и $g'(t)$, в которые они перерабатываются, также совпадают для всех $\tau \leq t^*$. Вместе с тем нетрудно убедиться в справедливости обратного утверждения: всякий детерминированный оператор со входным и выходным алфавитом A и B может быть задан посредством нагруженного дерева D . Для этого зафиксируем произвольное ребро r ранга t_v и покажем, какой выходной буквой его нагрузить. Заметим, что для всех путей, проходящих через это ребро, являются общими и все t_{v-1} ребра низших рангов, а поэтому и у всех соответствующих этим путям входных последовательностей в алфавите A общими являются первые t_v букв $f(1), f(2), \dots, f(t_v)$.

*) В частности, $g(t)$ однозначным образом определяется значениями $f(1), f(2), \dots, f(t)$.

Поскольку рассматриваемый оператор по предположению является детерминированным, то все такие последовательности перерабатываются в выходные последовательности, для которых первые t_v букв: $g(1), g(2), \dots, g(t_v)$

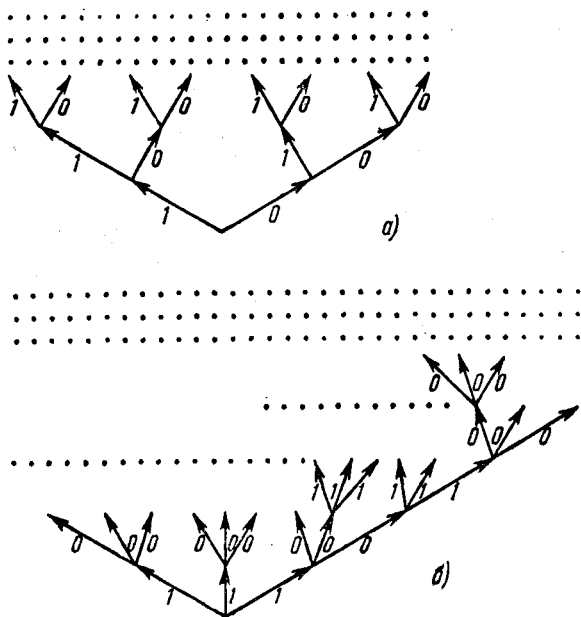


Рис. 2.6.

также являются одними и теми же. Теперь остается нагрузить выделенное (произвольным образом) ребро r выходной буквой $g(t_v)$.

Примеры.

1. Истинностный оператор, реализуемый триодом, задается нагруженным деревом, изображенным на рис. 2.6,а. Очевидно, что специфической особенностью любого истинностного оператора является то, что в его дереве все ребра, выходящие из некоторой вершины, нагружены так же, как и ребра, выходящие из любой другой вершины.

2. К о н с т а н т н ы й о п е р а т о р, порождающий последовательность 101001000100001..., задается нагруженным деревом, изображенным на рис. 2.6, б (здесь предполагается, что входной алфавит является троичным). Всем ребрам первого ранга приписано выходное значение 1, всем ребрам второго ранга — выходное значение 0 и, вообще говоря, каждому ребру i -го ранга — i -я буква в указанной последовательности. Очевидно, что специфической особенностью любого константного оператора является то, что в его дереве всем ребрам одного и того же яруса приписана одна и та же буква выходного алфавита.

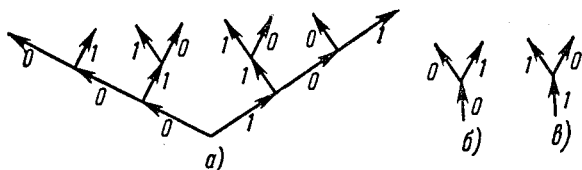


Рис. 2.7.

3. Рассмотрим детерминированный оператор с двоичными входным и выходным алфавитами, задаваемый деревом, изображенным на рис. 2.7, а, в котором каждое ребро, которому приписан выходной 0, разветвляется, так, как это показано на рис. 2.7, б, а каждое ребро, которому приписана выходная 1, — как на рис. 2.7, в*).

Этот оператор переводит, например, тождественно-нулевую последовательность в тождественно-нулевую, а последовательность, состоящую из одних единиц, в последовательность, состоящую из чередующихся единиц и нулей.

§ 3. Ограниченно-детерминированные операторы (операторы с конечным весом)

Какова бы ни была вершина γ ранга $j > 1$ в дереве D , она может быть рассмотрена как корень дерева D_γ , образованного всеми путями, выходящими из γ и уходящими в ярусы рангов $\geq j$. Такое дерево D условимся называть

*) При этом будем предполагать, что ребра, выходящие из корня, нагружены, соответственно их порядку, буквами 0 и 1.

ветвью ранга j заданного дерева D . Пусть $f(1) f(2) \dots f(j-1)$ — входное слово длины $j-1$, изображенное ломаной линией, состоящей из $(j-1)$ ребер и ведущей от корня дерева D к вершине γ . Если дерево D является нагруженным и, следовательно, задает некоторый детерминированный оператор θ , то и дерево D_γ также является нагруженным и задает некоторый детерминированный оператор, который мы назовем *остаточным оператором ранга j* и обозначим через θ_γ (сам оператор θ будем при этом рассматривать как единственный *остаточный оператор ранга 1*).

О п р е д е л е н и е. Два оператора с одним и тем же входным алфавитом (в частности, два остаточных оператора одного и того же оператора) называются *различимыми*, если существует хотя бы одна такая входная последовательность, которая перерабатывается ими в различные выходные последовательности; в противном случае операторы считаются *неразличимыми*.

Эта терминология естественным образом переносится и на соответствующие *деревья*, а также *ветви* данного дерева. Геометрический смысл неразличимости деревьев ясен; при наложении одного дерева (или ветки) на другое каждому из двух совмещающихся ребер оказывается приписанной всегда одна и та же буква выходного алфавита.

П р и м е р ы.

1. В дереве *истинностного оператора* (рис. 2.6, а) все ветви попарно неразличимы и каждая из них неразличима от всего исходного дерева. Очевидно, это характерно для дерева любого истинностного оператора.

2. В дереве *константного оператора* (рис. 2.6, б) имеется бесчисленное множество попарно различных ветвей. Точнее, ветви различных рангов всегда различимы, в то время как ветви одного ранга неразличимы. Все остаточные операторы здесь являются константными, причем остаточный оператор ранга j порождает остаточную выходную последовательность, получаемую из последовательности 101001000100001... путем удаления первых $(j-1)$ букв.

3. В дереве, изображенном на рис. 2.7, а, наибольшее число попарно различных ветвей равно двум. Именно,

если в корень ветви втекает снизу ребро, которому приписан выходной нуль, то такая ветвь неразличима от исходного дерева. Если же этому ребру приписана выходная единица, то соответствующая ветвь неразличима от ветви с правым корнем второго ранга.

Если детерминированный оператор θ является оператором реального физического устройства и если остаточные операторы θ и θ_γ различимы, то обстоятельства, отмеченные выше, должны быть интерпретированы следующим образом: подача на устройство входного слова $f(1)f(2)\dots f(j-1)$ переводит его из начального состояния, в котором оно реализовало оператор θ , в новое состояние, в котором оно реализует оператор θ_γ . Мы, естественно, допускаем, что всякое реальное устройство, перерабатывающее дискретную информацию, может пребывать лишь в конечном числе (функционально) различных состояний. Это побуждает нас выделить в классе всех детерминированных операторов подкласс тех операторов, которые подчинены дополнительным ограничениям, вытекающим из сделанного замечания.

О п р е д е л е н и е. Детерминированный оператор называется *ограниченным* (*ограниченно-детерминированный оператор*), или *оператором с конечной памятью*, если среди его остаточных операторов имеется лишь конечное число попарно различных операторов; в противном случае он называется *неограниченным*.

О п р е д е л е н и е. *Весом K ограничено-детерминированного оператора* называется максимальное число его попарно различных остаточных операторов.

С л е д с т в и е. Для того чтобы оператор имел вес 1, необходимо и достаточно, чтобы он был истинностным.

П р и м е ч а н и е. Неограниченному детерминированному оператору будем относить вес ∞ .

В соответствии с принятым нами допущением о конечности числа различных состояний, в которых может пребывать реальное физическое устройство, объектом дальнейших рассмотрений будет не класс всех детерминированных операторов, а лишь его подкласс — класс ограничено-детерминированных операторов. В частности,

это означает, что константный оператор из примера 2 не может быть реализован в устройствах рассматриваемого типа. Вопрос о физической реализуемости любого ограниченно-детерминированного оператора будет рассмотрен позднее (см. § 6), причем будет установлено, что всякий такой оператор на самом деле реализуем физически в некотором устройстве синхронного действия.

Будем называть *базисом детерминированного оператора* любую систему попарно различных его остаточных операторов, такую, что всякий остаточный оператор неразличим по крайней мере от одного оператора этой системы.

Легко проверяется справедливость следующего утверждения: если ограниченно-детерминированный оператор имеет вес, равный K , то любой его базис состоит в точности из K остаточных операторов и любая из систем K попарно различных операторов является базисом.

Определения и терминология, введенные выше применительно к операторам (системам операторов), будут применяться в соответствующем смысле также к деревьям (системам деревьев), изображающим их, а также к корням этих деревьев. В частности, перефразируя утверждение о базисах ограниченно-детерминированного оператора, можно сказать, что в дереве с весом K (т. е. с максимальным числом попарно различных ветвей, равным K) всякий базис состоит в точности из K ветвей (или из K вершин). Например, в дереве, изображенном на рис. 2.7, а, в качестве базиса можно брать систему двух ветвей (или систему из двух вершин) второго ранга.

К а н о н и ч е с к и е у р а в н е н и я о г р а н и ч е н н о - д е т е р м и н и р о в а н н о г о о п е р а т о р а . Пусть дан ограниченно-детерминированный оператор θ с весом K . Введем в рассмотрение K -значный алфавит $Q = \{q_0, q_1, \dots, q_{K-1}\}$, который будем называть *внутренним алфавитом (алфавитом состояний)* и припишем каждой вершине дерева оператора θ по одной букве из алфавита Q с соблюдением требования: двум вершинам приписывается одна и та же буква в том и только в том случае, когда эти вершины являются корнями неразличимых ветвей (короче, если вершины неразличимы). Дерево, пополненное таким образом буквами из алфавита Q , сопоставляет каждой входной последовательности $f(t)$ (каждому пути

вытекает одновременное выполнение равенств

$$z(t) = z'(\tau) \quad \text{и} \quad q(t+1) = q'(\tau+1). \quad (2.8')$$

Иными словами, это означает, что если два одноименных ребра $[x(t) = x'(\tau)]$ выходят из двух неразличимых вершин $[q(t) = q'(\tau)]$, то они являются носителями одной и той же выходной буквы $[z(t) = z'(\tau)]$ и втекают в две вершины, которые также неразличимы $[q(t+1) = q'(\tau+1)]$. Итак, выходная буква $z(t)$ и внутренняя буква $q(t+1)$ являются однозначными функциями от пары букв $x(t)$ и $q(t)$:

$$\left. \begin{aligned} z(t) &= \Phi[x(t), q(t)], \\ q(t+1) &= \Psi[x(t), q(t)]. \end{aligned} \right\} \quad (2.9)$$

Здесь $\Phi(x, q)$ и $\Psi(x, q)$ являются однозначными функциями, аргументы которых пробегают независимо входной алфавит A и алфавит состояний Q соответственно; эти функции могут быть заданы таблично; в случае, когда буквы x, z, q являются наборами из единиц и нулей, по этим таблицам известным читателю способом составляются соответствующие формулы исчисления высказываний (ср., например, формулы, составленные по таблице II.2 истинностного оператора, стр. 91).

П р и м е р. Для оператора, представленного деревом, изображенным на рис. 2.8, принимая 0 за q_0 и 1 за q_1 , получаем таблицу II.3.

Т а б л и ц а II.3

$x(t)$	$q(t)$	$\Phi[x(t), q(t)],$ т. е. $z(t)$	$\Psi[x(t), q(t)],$ т. е. $q(t+1)$
0	0	0	0
1	0	1	1
0	1	1	1
1	1	0	0

В соответствии с этой таблицей уравнения (2.9) в данном случае запишутся в логической символике следующим образом:

$$\begin{aligned} z(t) &= x(t) \cdot \bar{q}(t) \vee \bar{x}(t) \cdot q(t), \\ q(t+1) &= x(t) \cdot \bar{q}(t) \vee \bar{x}(t) \cdot q(t), \end{aligned}$$

Поскольку нас больше всего интересует двоичное кодирование, то обычно в развернутой записи (2.10) будут фигурировать двоичные переменные $x_1, \dots, x_m, z_1, \dots, z_n$.

q_1, q_2, \dots, q_k . Это позволяет задать правые части (2.10) в виде логических формул.

Нетрудно усмотреть, что справедливо и обратное утверждение: каковы бы ни были функции Φ и Ψ в уравнениях (2.9) или (2.10) с заданным начальным условием, они задают ограниченно-детерминированный оператор, вес которого не превышает числа букв в алфавите состояний. Дерево этого оператора строится так: 1) корню приписывается начальное состояние; 2) ребрам первого ранга и вершинам второго ранга приписываются выходные буквы и внутренние буквы состояния, согласно уравнениям (2.9), и т. д. Очевидно, вершины, к которым отнесено одно и то же состояние, окажутся обязательно неразличимыми, хотя и не исключена возможность, что вершины, к которым отнесены различные буквы состояний, на самом деле неразличимы.

Уравнения (2.9) и функции Φ, Ψ , фигурирующие в них, а также таблицы, задающие эти функции, будем называть *каноническими уравнениями (функциями*)*, *таблицами* над алфавитами X, Z, Q (где X — входной алфавит, Z — выходной алфавит, а Q — алфавит состояний).

Рассмотрим, например, следующую таблицу.

Таблица II.4

$x_1(t)$	$x_2(t)$	$q(t)$	$z(t)$	$q(t+1)$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Ее можно считать канонической таблицей, задающей соответственно выход $z(t)$ и состояние $q(t+1)$ некоторого

*) В литературе употребляются также термины: «внутренняя (Ψ) функция переходов» и «внешняя (Φ) функция переходов».

ограниченно-детерминированного оператора θ как функции от $x(t)$ и $q(t)$; при этом необходимо еще указать начальное состояние $q(1)$. Как бы мы ни выбрали это начальное состояние, вес полученного оператора не будет больше двух. Пусть для определенности $q(1) = q_0$; на рис. 2.9 изображен кусок дерева оператора, а именно две

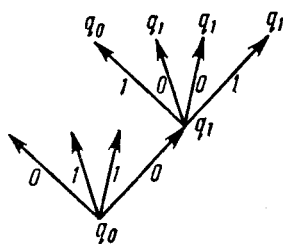


Рис. 2.9.

его вершины q_0 и q_1 с выходящими из них четверками ребер, соответствующих входным буквам 00, 01, 10, 11. Эти ребра, а также вершины, в которые они втекают, нагружены выходными буквами и буквами состояний в соответствии с таблицей II.4. На рис. 2.10 изображен более обширный «кусок» нагруженного дерева оператора θ , полученный путем продолжения куска дерева, изображенного на рис. 2.9.

Непосредственной проверкой легко убедиться, что оператор θ , задаваемый канонической таблицей (табл. II.4) и условием $q(1) = q_0$, является оператором поразрядного последовательного суммирования, перерабатывающего пару цифр слагаемых в соответствующую цифру суммы (все цифры одноименных разрядов). Если же в качестве начального состояния взять q_1 ,

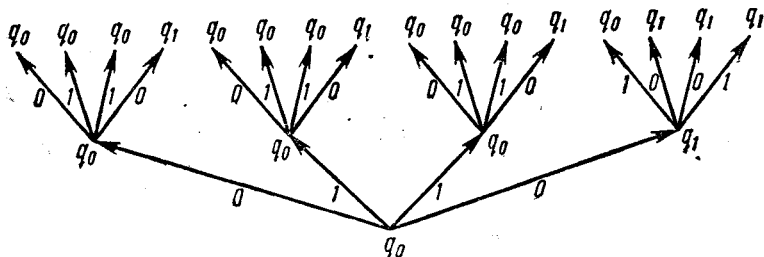


Рис. 2.10.

то мы получим оператор θ_1 , отличающийся от θ (θ_1 выдает сумму, увеличенную на единицу). По таблице II.4 обычным образом устанавливаются канонические урав-

нения в логической символике (т. е. $z(t)$ и $q(t+1)$ записываются как функции алгебры логики от $x(t)$ и $q(t)$):

$$z(t) = \bar{x}_1 x_2 \bar{q} \vee x_1 \bar{x}_2 \bar{q} \vee \bar{x}_1 \bar{x}_2 q \vee x_1 x_2 q = x_1 \oplus x_2 \oplus q,$$

$$q(t+1) = x_1 x_2 \vee x_1 q \vee x_2 q.$$

Аналогичным образом можно построить деревья операторов, описанных таблицей II.5, при том или ином выборе $q(1)$. Здесь внутренняя буква q может принимать четыре значения: 00, 01, 10, 11; через y_1 и y_2 обозначены первая и вторая (слева) двоичные цифры буквы q .

Таблица II.5

$x_1(t)$	$x_2(t)$	$y_1(t)$	$y_2(t)$	$z(t)$	$y_1(t+1)$	$y_2(t+1)$
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	1	1	0
0	1	0	1	0	1	1
0	1	1	0	1	1	0
0	1	1	1	0	1	1
1	0	0	0	1	0	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	0	1
1	1	0	1	1	0	1
1	1	1	0	0	1	1
1	1	1	1	1	1	1

Мы не станем строить куски соответствующих деревьев и ограничимся лишь следующим замечанием. Приведенная выше таблица при начальном условии $q(1) = 10$ задает оператор последовательного поразрядного суммирования, вес которого фактически равен двум ($K=2$), хотя алфавит состояний состоит из четырех букв. По существу буквы 00 и 10 обозначают одно и то же состояние (прежнее q_0); одно и то же состояние обозначают также и буквы 01 и 11. Практические приемы, удобные для определения веса оператора по таблицам и ликвидации избы-

точности алфавита состояний, будут рассмотрены позднее в связи с задачами синтеза и анализа.

Условимся называть буквы q_i, q_j *неразличимыми относительно канонических функций* Φ и Ψ , если эти функции задают один и тот же ограниченно-детерминированный оператор при каждом из начальных условий: $q(1) = q_i, q(1) = q_j$; в противном случае q_i, q_j считаются *различимыми*. Таким образом, паре канонических функций (уравнений) Φ и Ψ с K -значным алфавитом состояний соответствует система $\mathcal{A}_{\Phi, \Psi}$ из K ограниченно-детерминированных операторов, где K — максимальное число попарно различных букв алфавита Q . Эту систему $\mathcal{A}_{\Phi, \Psi}$ мы будем называть *общим ограниченно-детерминированным оператором* (или короче: *общим оператором*) *пары канонических функций* Φ, Ψ , а число K — *весом общего ограниченно-детерминированного оператора*, в отличие от *частного ограниченно-детерминированного оператора* и его *веса*, которые получаются при фиксации начального условия. Там, где это не вызывает недоразумений, мы будем говорить просто об операторах, опуская термины «частный» или «общий».

Очевидно, число частных операторов и вес каждого из них не превосходят веса общего оператора. Отметим следующие свойства общего оператора $\mathcal{A}_{\Phi, \Psi}$.

1) $\mathcal{A}_{\Phi, \Psi}$ является конечной системой детерминированных операторов $\{\theta_\mu\}$ с общими входным и выходным алфавитами.

2) Если в $\mathcal{A}_{\Phi, \Psi}$ входит оператор θ , то в него входят и все его остаточные операторы (замкнутость).

Нетрудно усмотреть, что для всякой системы \mathcal{A} , удовлетворяющей этим условиям, может быть построена каноническая пара Φ, Ψ , такая, что $\mathcal{A} = \mathcal{A}_{\Phi, \Psi}$.

§ 4. Элементы (элементарные автоматы) и схемы

Физическое устройство, в котором реализуются сложные процессы преобразования дискретной информации, представляет собой обычно набор более простых частей, надлежащим образом взаимодействующих благодаря целесообразному соединению в единую систему. В предыдущих параграфах при изучении операторов эти конструк-

тивные особенности не рассматривались. Пользуясь термином, получившим распространение в кибернетической литературе, можно сказать, что до сих пор мы подходили к рассматривавшимся нами устройствам как к «черным ящикам», конструкция которых неизвестна, но поведение которых описывается с помощью надлежащих систем канонических уравнений. Заглянем теперь в «черный ящик» с тем, чтобы выяснить принципы его конструкции, а также взаимосвязь между этой конструкцией и выполняемой ею функцией. Мы будем рассматривать лишь тот случай, когда составные части системы, не подлежащие дальнейшему расчленению, сами тоже являются устройствами («элементарными автоматами»), работа которых описывается системами канонических уравнений.

В соответствии с этим интуитивное представление о конструкции автомата мы уточним посредством формального определения понятия *схемы*. В рамках нашей теории схема предназначена для описания того, каким образом данный *сложный автомат* построен из не подлежащих дальнейшему расчленению *элементарных автоматов* или просто *элементов*. Элементарные автоматы рассматриваются, по существу, как входящие в состав сложных автоматов «черные ящики», о которых известно только то, как они функционируют (т. е. какие операторы они реализуют). Предполагается также, что если заданы схемы и элементы, то оператор, реализуемый сложным автоматом, определяется однозначно с помощью некоторого *алгоритма (алгоритма функционирования схемы)*. В настоящем параграфе даются определения понятий *элемента* (элементарного автомата)*), *схемы* и других связанных с ними понятий и излагаются некоторые общие соображения об алгоритме функционирования схемы; этот последний вопрос подробнее рассматривается в следующем параграфе.

Избегая чрезмерной формализации, ограничимся описательным определением понятия *элемента*. Каждый элемент определяется заданием: 1) *ячейки*, употребляю-

*) Приводимое нами определение понятия *элементарного автомата* в основных своих чертах совпадает с тем определением понятия *конечного автомата*, которым пользуются авторы работ, в которых поведение автоматов изучается при отвлечении от их конструкции.

переменной функции Φ_i . Это еще не исключает указанной подчиненности; может оказаться, например, что физическое срабатывание выходного канала z_i обусловливается предварительной подачей в том же такте входного сигнала x_j (причем в силу фиктивности переменной x_j безразлично, какая именно буква подана). Для выяснения того, имеет ли место такая подчиненность или нет, обычно необходим специальный анализ физических принципов, лежащих в основе реальной системы. Такой анализ лежит вне нашей теории, и поэтому при рассмотрении «элементарных» автоматов мы будем предполагать, что отношение подчиненности выходных каналов входным уже установлено и явно указано. В тех же случаях, когда такие указания не даны, будет подразумеваться, что отношение подчиненности канала z_i каналу x_j равносильно существенной зависимости функции Φ_i от переменной x_j в уравнениях (2.11).

Пусть задано некоторое множество элементов $\{M_i\}$. Предполагаемые ниже определения являются естественным обобщением определений Бёркса и Райта*), сформулированных ими в более ограниченном виде.

О п р е д е л е н и е. *Схемой* над множеством элементов $\{M_i\}$ называется любой набор элементов из $\{M_i\}^{**}$, для которых указано, какие их полюса считаются отождествленными, причем отождествлять можно лишь такие полюса, к которым отнесен один и тот же алфавит. *Множеством полюсов схемы* называется объединение множеств полюсов ее элементов (отождествленные полюса считаются за один полюс). *Множеством ячеек* и *множеством каналов* схемы называются соответственно объединение множества ячеек и объединение множеств каналов элементов; ячейкам и полюсам схемы отнесены те же алфавиты, что и в составных элементах.

Таким образом, схема представляет собою *граф* с ориентированными *ребрами* (каналами) и *вершинами* двух сортов: *ячейками* (носителями внутренних алфавитов) и *полюсами* (носителями внешних алфавитов). Этот граф

*) A. W. Burks and G. B. Wright, Theory of logical nets. Proceedings of IRE, v. 41, № 10, 1953.

**) Причем в набор может входить много экземпляров одного и того же элемента из множества $\{M_i\}$.

характеризует способ соединения элементарных автоматов друг с другом, а также течение информации в полученной сложной системе.

О п р е д е л е н и е. Полюс схемы называется *входным*, если в него не втекают никакие входные каналы схемы; в противном случае полюс считается *выходным*.

Для полюсов схемы индуктивно определяется отношение *подчиненности* (обозначение: $\alpha \succ \beta$ или $\beta \prec \alpha$).

1. Если α и β таковы, что существует элемент (в составе данной схемы), выходной канал которого g оканчивается в полюсе α (втекает в α) и подчинен входному каналу f , который начинается в полюсе β (вытекает из β), то полюс схемы α подчинен полюсу схемы β (относительно этого элемента), то есть $\alpha \succ \beta$.

2. Если $\alpha \succ \beta$, $\beta \succ \gamma$, то $\alpha \succ \gamma$.

3. В остальных случаях отношение подчиненности не имеет места.

П р и м е р. Рассмотрим следующую систему элементов, обозначаемых буквами $\mathfrak{M}_\Sigma, \mathfrak{M}_\odot, \mathfrak{M}_\oplus, \mathfrak{M}_\exists$, с соответствующими им каноническими уравнениями.

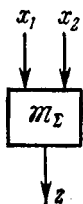


Рис. 2.11.

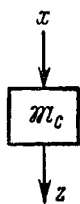


Рис. 2.12.

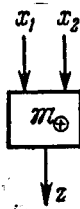


Рис. 2.13.

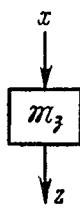


Рис. 2.14.

1. \mathfrak{M}_Σ — сумматор последовательного действия при $q(1) = 0$ (рис. 2.11):

$$z(t) = x_1(t) \oplus x_2(t) \oplus q(t),$$

$$q(t+1) = x_1(t)x_2(t) \vee q(t)x_1(t) \vee q(t)x_2(t).$$

2. \mathfrak{M}_\odot — двоичный счетчик при $q(1) = 0$ (рис. 2.12):

$$z(t) = x(t) \oplus q(t),$$

$$q(t+1) = x(t) \oplus q(t).$$

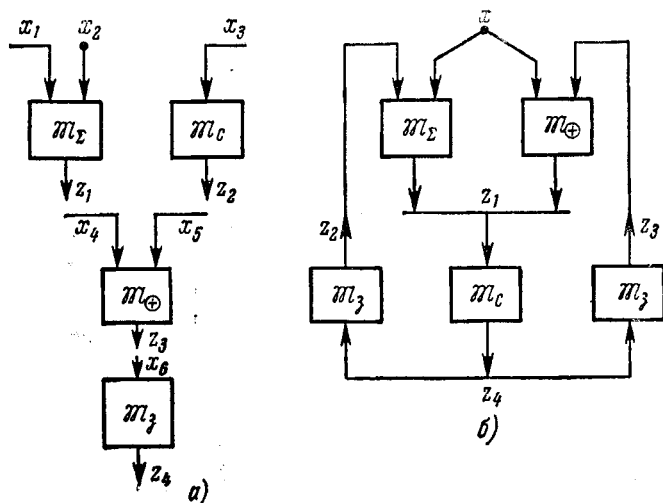


Рис. 2.16.

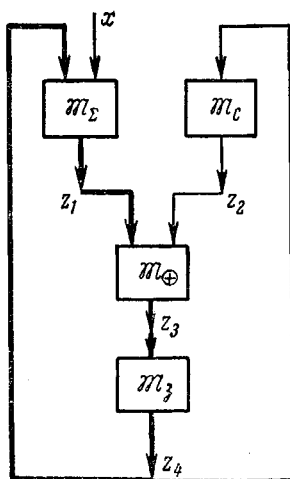


Рис. 2.17.

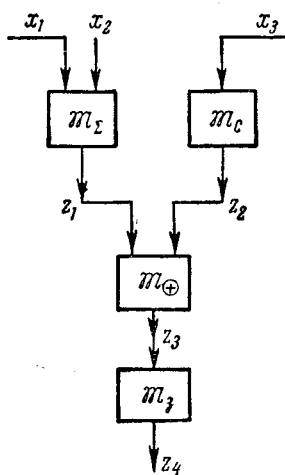


Рис. 2.18.

и *неправильной* в противном случае. Легко видеть, что в неправильной петле все полюсы подчинены друг другу. Содержательное значение этих понятий заключается в том, что в неправильной петле возникает *п о р о ч н ы й к р у г в п е р е д а ч е и н ф о р м а ц и и*. Правильность петли является, по существу, условием разрыва порочного круга; такие разрывы происходят как раз в тех полюсах, которые не подчинены другим полюсам петли.

Перейдем теперь к рассмотрению вопроса о *ф у н к ц и о н и р о в а н и и с х е м ы*. Здесь необходимо отметить два аспекта этого вопроса. Во-первых, в классе всех схем нужно выделить класс «правильно организованных» (в определенном смысле) схем, описывающих такие соединения физических элементов, поведение которых можно рассматривать как работу автомата. С другой стороны, для каждой «правильной» схемы должно быть указано, какой ограниченно-детерминированный оператор ею реализуется. Требования, предъявляемые к «правильно организованным» схемам, выражают физические условия правильного течения информации в системе и правильного обмена информацией между отдельными ее частями. С математической же точки зрения они представляют собой постулаты, привлекаемые для формулировки соответствующих определений.

Пусть рассматривается класс всех схем над некоторым множеством элементов $\{M_i\}$. Наши абстрактные элементы допускают, вообще говоря, различные физические интерпретации (например, они могут быть истолкованы как электронные лампы, ферриты и т. п.; см. гл. III), от которых может существенно зависеть поведение схемы в целом. Укажем здесь в общих чертах две типичные ситуации, которые чаще всего возникают на практике:

1. В рамках всей схемы каждый из составных элементов продолжает выполнять операции, свойственные ему до соединения. В полюсах же, где соединены несколько каналов, никакой переработки информации не происходит, а лишь имеет место разветвление потока информации с передачей ее из одних каналов в другие. Можно сказать, что в этом случае поведение схемы в целом представляет суммарное поведение ее составных элементов.

Точнее это обстоятельство может быть охарактеризовано следующими двумя требованиями.

1) Выходная буква, следующая по любому выходному каналу, в каждом такте однозначно определена свойствами того элемента, которому этот канал принадлежит.

2) В каждом такте во всех каналах, присоединенных к одному полюсу, проходит одна и та же буква. Схемы

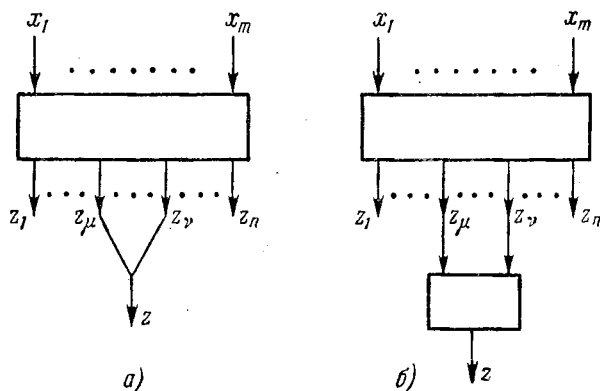


Рис. 2.19.

такого типа исследовались Бёрксом и Райтом под названием *логических сетей*.

II. Наряду с операциями, осуществляемыми за счет самих составных элементов (в соответствии с их каноническими уравнениями), может происходить еще и преобразование информации за счет соединения каналов. Так, в некоторых реальных устройствах, например в электронно-ламповых (см. гл. III), соединение выходных каналов z_μ и z_ν обеспечивает иногда переработку двух сигналов, идущих по соединенным каналам, в другой сигнал, выдаваемый на общий выход (рис. 2.19, а). Функционирование схем такого типа будет описано в гл. III в связи с рассмотрением соответствующих физических элементов. Однако на данном этапе изложения подобные случаи удобнее — в целях достижения общности метода — трактовать так, как будто бы буквы, идущие по каналам z_μ и z_ν , поступают далее на входы некоторой («фиктивной») ячейки, перераба-

тывающей их в букву, выдаваемую на выход (рис. 2. 19, б). Заметим, что эти «фиктивные» ячейки допускают и более реальную интерпретацию, которая делает их не столь уже фиктивными. Иногда на стыке соединения выходных каналов приходится помещать некоторые дополнительные элементы, например, усилительные.

Наша ближайшая задача заключается в формальном уточнении тех представлений о правильно организованной схеме и ее работе, которые были намечены в пункте I. Понятие *логической сети*, намеченное в этом пункте, исследуется в следующем параграфе.

§ 5. Логические сети

В этом параграфе содержится точное определение понятия логической сети, включающее алгоритм ее функционирования и представляющее собой естественное обобщение определения Бёркса и Райта*); определение носит индуктивный характер. При этом термин «правильно организованная схема» применяется неформально на этапе эвристического рассмотрения вопроса.

Каждой схеме N можно сопоставить систему уравнений $E(N)$, описывающую ее структуру, а в случае, когда схема является правильно организованной, также и ее поведение. В этих целях одно-однозначным образом припишем полюсам и ячейкам схемы переменные, причем:

1) каждому полюсу припишем переменную, могущую принимать любое значение из внешнего алфавита, соответствующего этому полюсу, 2) каждой ячейке в составе схемы припишем переменную, могущую принимать значения из соответствующего внутреннего алфавита.

Термины «внешняя», «внутренняя», «выходная», «входная» будут применяться к этим переменным в соответствии с разделением полюсов на входные и выходные, а алфавитов на внешние и внутренние.

Система $E(N)$ получается следующим образом. Для каждого из составных элементов берутся его канонические уравнения с той лишь оговоркой, что наименования

*) Отметим, что Бёркс и Райт при этом пользуются термином «вполне правильная логическая сеть»; их термин «логическая сеть» соответствует нашему термину «схема».

функциональных переменных берутся такими же, какими они были приписаны полюсам этого элемента и его ячейке. Объединение всех таким образом составленных уравнений и составляет систему $E(N)$.

П р и м е р. Для схемы, изображенной на рис. 2.17, система $E(N)$ имеет вид:

$$\left. \begin{aligned} z_1(t) &= z_4(t) \oplus x(t) \oplus q(t), \\ q(t+1) &= z_4(t) x(t) \vee q(t) z_4(t) \vee q(t) x(t), \\ z_2(t) &= z_4(t) \oplus p(t), \\ p(t+1) &= z_4(t) \oplus p(t), \\ z_3(t) &= z_1(t) \oplus z_2(t), \quad z_4(t) = r(t), \quad r(t+1) = z_3(t). \end{aligned} \right\} \quad (2.12)$$

Вернемся к системе $E(N)$. Пусть в этой системе (p и r — буквы внутреннего алфавита элементов \mathfrak{M}_\oplus и \mathfrak{M}_\vee) фигурируют входные переменные x_1, x_2, \dots, x_m и выходные переменные z_1, z_2, \dots, z_n .

Рассмотрим следующие свойства, которыми система может обладать или не обладать:

А. Для любой системы входных функций $x_1(t), \dots, x_m(t)$ существует система выходных функций $z_1(t), \dots, z_n(t)$ такая, что функции $x_1(t), \dots, x_m(t), z_1(t), \dots, z_n(t)$ в совокупности удовлетворяют уравнениям $E(N)$.

Б. При условии пункта А система $z_1(t), \dots, z_n(t)$ единственна.

Очевидно, при одновременном выполнении свойств А и Б система $E(N)$ задает однозначный оператор θ , перерабатывающий любой набор входных функций в соответствующий набор выходных функций.

В связи с этим напрашивается рассмотрение свойства

В. Оператор θ , определяемый условиями А и Б, является ограниченно-детерминированным.

Если система $E(N)$ обладает свойствами А, Б и В, то будем говорить, что система *правильна*.

Неформальная характеристика поведения правильно организованной схемы, которая была дана в предыдущем параграфе, подсказывает нам, что определение логической сети и алгоритма ее функционирования должно предусматривать правильность системы $E(N)$ и объявление огра-

В таком случае совокупность этих схем естественно также рассматривать как правильно организованную схему Ω с $m + \mu$ входными и $n + \nu$ выходными полюсами, как это изображено на рис. 2.20, б. Такое допущение согласуется и с тем, что система уравнений $E(\Omega)$, являющаяся объединением систем $E(\Omega_1)$ и $E(\Omega_2)$, будет, как это очевидно, тоже правильной, причем ограниченно-детерминированный оператор, задаваемый ею, описывается объединением систем канонических уравнений (2.13) и (2.13').

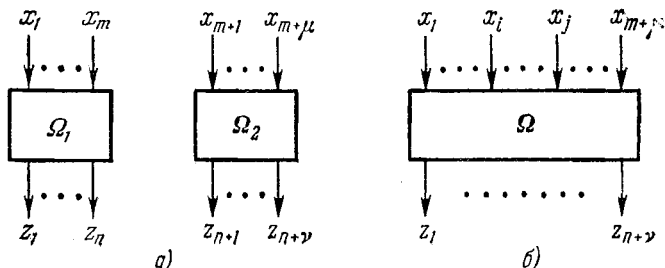


Рис. 2.20.

Положение, обрисованное нами для двух схем, естественным образом обобщается на любое их число.

Пример. Совокупность S двоичных счетчиков, каждый из которых реализует оператор с весом 2, реализует в указанном выше смысле оператор, перерабатывающий S -разрядные двоичные числа в двоичные числа той же разрядности. Легко проверить, что его вес равняется в точности 2^S .

Рассмотрим теперь возможность отождествления полюсов правильно организованной схемы с целью получения другой «правильно организованной» системы передачи и переработки информации. Обратимся сначала к случаю одноименных (т. е. либо только входных, либо только выходных) полюсов. Соединение двух выходных каналов (см. рис. 2.19, а) физически не представляется абсурдным и в реальных устройствах на самом деле иногда применяется. В некоторых случаях, как уже было отмечено ранее, таким путем осуществляется переработка информации, следующей по соединяемым каналам; однако мы

условились исключить пока из рассмотрения подобную ситуацию.

Соединение выходных каналов может преследовать и другие цели. Именно, может оказаться, что в физической системе по соединяемым каналам идут параллельно на протяжении всего процесса одинаковые сигналы; их объединение в этом случае диктуется исключительно соображениями надежности (например, оно может обеспечивать усиление сигнала). Поскольку мы намерены рассматривать идеализированные математические модели, в которых сигналы (буквы), идущие по разным каналам, не различаются по своей интенсивности, не подвержены ошибкам и искажениям, соединение выходных каналов в целях большей надежности не имеет смысла.

В силу изложенных выше обстоятельств мы будем пока считать недопустимым непосредственное соединение двух или более выходных каналов *).

Отождествление двух входных полюсов физически означает подключение их к общему источнику информации и широко применяется в реальных физических системах.

Пусть в правильно организованной схеме Ω (рис. 2.21) отождествлены входные полюсы x_i и x_j ; в результате получится схема $\tilde{\Omega}$, имеющая на один входной полюс меньше, причем ее система уравнений $E(\tilde{\Omega})$ будет также правильной. Очевидно, оператор, задаваемый системой $E(\tilde{\Omega})$, может быть описан и посредством системы канонических уравнений, получаемых из Ω путем подстановки вместо переменных x_i и x_j одной и той же переменной (например, x_i).

П р и м е р. Если соединить оба входных полюса сумматора последовательного действия, то получится схема

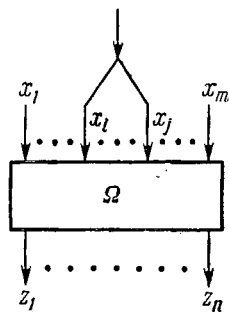


Рис. 2.21.

*) Позднее, в гл. III, в связи с рассмотрением конкретных физических элементов это ограничение будет снято.

устройства, реализующего оператор умножения на 2. Его вес, так же как и вес сумматора, равен двум.

Перейдем теперь к рассмотрению возможных отождествлений выходных полюсов со входными полюсами. Пусть, для определенности, в схеме Ω выходной полюс z_i подведен к входному x_j (рис. 2.22, а). В реальных физических устройствах сигнал, следующий по петле обратной связи, поступает на вход с некоторой временной задержкой

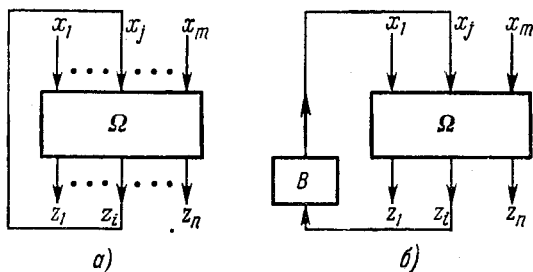


Рис. 2.22.

относительно момента его выработки (и выдачи). В идеализированной же математической модели без ограничения общности полагают, что прохождение буквы по петле происходит мгновенно, так что в соответствующий момент времени происходит одновременно выдача $z_i(t)$ и поступление его на прежний вход x_j . В самом деле, если за счет петли получается значительная задержка, сравнимая с длительностью такта работы устройства, то это обстоятельство удобно трактовать так, как будто бы буква, выработанная в канале z_i , прежде чем поступить на вход, проходит через «фиктивную» ячейку (B) с одним входом и одним выходом, осуществляющую задержку на один такт (рис. 2.22, б).

Итак, будем считать, что выдача буквы по выходу z_i и прием ее на вход x_j отнесены к одному и тому же моменту времени. В таком случае, естественно, напрашиваются некоторые требования, призванные предотвращать неопределенность в течении информации по замкнутой петле; именно, нужно потребовать, чтобы выход $z_i(t)$ в каком-то

В сумматоре последовательного действия выход z подчинен каждому из двух входов x_1, x_2 . Здесь образование петли путем присоединения z к x_1 (или x_2) недопустимо (рис. 2.24, а). При наличии же ячейки, реализующей задержку буквы z на один такт, допустимо соединение (указанное на рис. 2.24, б), которое порождает циркуляцию информации по петле.

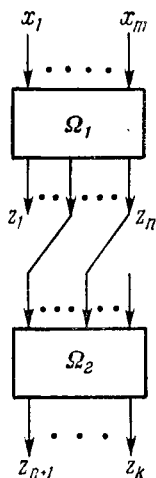


Рис. 2.23.

Высказанные выше неформальные соображения оправдывают выделение в особый класс всех схем (над данным множеством элементов \mathfrak{M}_i) из подкласса логических сетей*), определяемого индуктивно следующим образом.

О п р е д е л е н и е.

1. Всякий элемент из данного множества элементов является логической сетью над $\{\mathfrak{M}_i\}$.
2. Совокупность двух логических сетей без общих полюсов есть логическая сеть.
3. В результате отождествления двух входных полюсов x_i, x_j логической сети получается логическая сеть.
4. В результате отождествления выходного полюса z_i логической сети с таким ее входным полюсом x_j , которому z_i не подчинен, получается логическая сеть.
5. Других логических сетей не существует.

Операции, фигурирующие в пунктах 3 и 4, носят, соответственно, название *склеивания входов* и *подключения к выходу*. Повторным применением склеивания пары входов можно осуществить склеивание нескольких входов в один. Частными случаями подключения к выходу являются *каскад* и образование *петли*, смысл которых

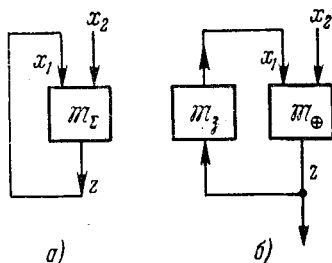


Рис. 2.24.

*) В дальнейшем изложении слова «над множеством \mathfrak{M}_i » часто опускаются.

ясен из рис. 2.24 и 2.25. Повторным подключением к выходу можно осуществить многократный каскад (см. рис. 2.23).

Из рассмотрений, которые мы предпослали этому определению, непосредственно вытекает следующее утверждение.

Для всякой логической сети N соответствующая система уравнений $E(N)$ является правильной, а следовательно, задает ограниченно-детерминированный оператор θ_N .

О п р е д е л е н и е. О логической сети N говорят, что она *реализует* оператор θ_N .

Из предыдущих рассмотрений вытекает также, что система канонических уравнений \mathfrak{N}_N , описывающая оператор θ_N , может быть получена индуктивно, в соответствии с пунктами 1—4 определения логической схемы, следующим образом:

1. Каноническая система заданного элемента представляет собой систему канонических уравнений \mathfrak{N}_N , описывающую ограниченно-детерминированный оператор θ_N .

2. Объединение двух систем канонических уравнений \mathfrak{N}_{N_1} и \mathfrak{N}_{N_2} , описывающих соответственно ограниченно-детерминированные операторы θ_{N_1} и θ_{N_2} , порождает систему канонических уравнений \mathfrak{N}_N , описывающую ограниченно-детерминированный оператор θ_N .

3. Подстановка в систему канонических уравнений \mathfrak{N}_N вместо переменных $x_i(t)$ и $x_j(t)$ одной и той же переменной $x(t)$ порождает систему канонических уравнений \mathfrak{N}_N .

4. Подстановка в системе канонических уравнений \mathfrak{N}_N во всех уравнениях, кроме уравнения $z_i(t) = \Phi_i[x_1(t), \dots]$, вместо переменной $x_j(t)$ выражения $\Phi_i[x_1(t), \dots]$ порождает систему канонических уравнений \mathfrak{N}_N .

Если логическая сеть N состоит из v элементов, внутренние алфавиты которых насчитывают соответственно k_1, k_2, \dots, k_v букв, то прямое произведение этих алфавитов, насчитывающее $k = k_1 \cdot k_2 \cdot \dots \cdot k_v$ букв, будет внутренним алфавитом в системе уравнений \mathfrak{N}_N . Логарифм этого числа мы будем называть *объемом внутренней памяти* логической сети N . Отсюда вытекает, что: (1) объем внут-



Рис. 2.25.

где $\eta_{i_1}, \eta_{i_2}, \dots, \eta_{i_v}$ — какое-то подмножество выходных полюсов. В дальнейшем там, где это не вызывает недоразумения, мы будем пользоваться без особых оговорок любым из двух вариантов определения. Легко показать, что вес любого из операторов вида (2.14) не более веса оператора, отнесенного ко всем выходным полюсам.

Пример. Пусть \mathcal{M}_c и \mathcal{M}'_c — два двоичных счетчика, а \mathcal{M}_Σ , \mathcal{M}'_Σ , \mathcal{M}''_Σ — три последовательных сумматора. На рис. 2.26 изображены схемы, образованные из этих ячеек. Схема первая слева реализует в полюсе z_1 оператор счетчика четности, в полюсе z_2 — оператор счетчика четырехкратности, а в обоих выходных полюсах одновременно — оператор счета по модулю 4. Вторая схема реализует в полюсе z_3 сумму четырех слагаемых, последовательно подаваемых по каналам x_1, x_2, x_3, x_4 .

Легко видеть, что оператор полюса z_2 в первой схеме имеет вес, равный весу оператора, отнесенного ко всем выходным полюсам, а именно 4. Во второй же схеме оператор, отнесенный к полюсу z_3 , имеет вес, равный 4, в то время как вес оператора, отнесенного ко всем выходным полюсам, равен 6.

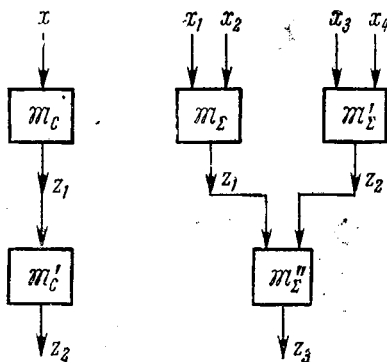


Рис. 2.26.

Перейдем теперь к обсуждению вопроса о физической реализуемости ограниченно-детерминированных операторов. При этом мы ограничиваемся рассмотрением лишь того случая, когда буквы всех алфавитов заданы в двоичном коде: на практике как раз этот случай представляет наибольший интерес в связи с преимущественным применением двоичных сигналов в физических элементах.

Рассмотрим сначала истинностный оператор, входные и выходные буквы которого представляют собою кодовые группы из нулей и единиц длины m и n соответственно. Реализация такого оператора означает, по существу, реализацию n функций алгебры логики от m переменных каждая:

$$\begin{aligned} z_1 &= \Phi_1(x_1, \dots, x_m), \\ &\dots \dots \dots \\ z_n &= \Phi_n(x_1, \dots, x_m). \end{aligned}$$

собой функции алгебры логики. Именно таким образом может быть задан любой ограниченно-детерминированный оператор, алфавиты которого состоят из двоичных кодовых групп. Пусть \mathfrak{M}_8 обозначает элемент с одним двоичным каналом, реализующий задержку одной буквы на единицу времени; его канонические уравнения таковы:

$$z(t) = q(t), \quad q(t+1) = x(t).$$

По уравнениям (2.16) соответствующая логическая сеть строится так, как это указано на рис. 2.28, не требующем пояснений, где для простоты взяты: $m = 2$, $n = 1$, $k = 1$.

Таким образом, при наличии истинностно-полного набора элементов и элементов задержки могут быть построены логические сети, реализующие любые из рассматриваемых ограниченно-детерминированных операторов.

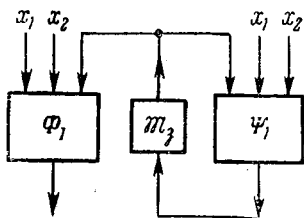


Рис. 2.28.

Будем говорить, что система элементов $\mathfrak{M} = \{\mathfrak{M}_1, \dots, \mathfrak{M}_n\}$ является *полной*, если для любого ограниченно-детерминированного оператора θ существует логическая сеть, реализующая θ . Из сказанного выше вытекает, что система элементов, содержащая элемент задержки и истинностно-полный набор элементов, является *полной*. Поскольку существуют физические элементы задержки и истинностно-полные системы физических элементов, то существуют и полные системы физических элементов; в этом смысле все ограниченно-детерминированные операторы физически реализуемы. Критерии полноты систем элементов, обобщающие теорему 3 гл. I, были установлены В. Кудрявцевым*). Некоторые полные системы физических элементов будут рассмотрены нами в гл. III.

Дадим теперь предварительную формулировку задач анализа и синтеза конечных автоматов.

*) В. Б. Кудрявцев, Вопросы полноты для систем автоматов, ДАН СССР, 1960, т. 130, № 6; с его же, Теорема полноты для одного класса автоматов без обратных связей, ДАН СССР, 1960, т. 132, № 2.

Анализ автомата. Исходя из имеющихся сведений о схеме (логической сети) автомата, требуется определить, какой ограниченно-детерминированный оператор эта схема реализует (или указать свойства этого оператора).

Синтез автомата. Сформулированы требования к преобразованию информации (подлежащему реализации) и указана система элементов \mathfrak{M} ; требуется построить логическую сеть над \mathfrak{M} , реализующую ограниченно-детерминированный оператор, удовлетворяющий предписанным условиям.

Процесс синтеза складывается из двух этапов:

Первый этап состоит в изучении заданных требований с тем, чтобы выяснить, существует ли удовлетворяющий им ограниченно-детерминированный оператор, и если да, то получить его канонические уравнения.

Второй этап состоит в построении (синтезе) логической сети по полученным каноническим уравнениям оператора; при этом обычно ставится задача синтеза оптимальной (в смысле принятых критериев) логической сети.

Здесь необходимо еще уточнить, в каких терминах допустима первоначальная формулировка требований, что значит оптимальная логическая сеть, и т. д. Эти уточнения будут даны ниже, в гл. IV—VII, в которых подробно исследуются задачи анализа и синтеза автоматов.

В заключение этого параграфа укажем еще на некоторые обобщения того определения реализуемости оператора в логической сети, которого мы придерживались до сих пор. Без такого обобщения трактовка задач анализа и синтеза оказывается на практике стеснительной. Пусть θ и θ' — операторы с одними и теми же алфавитами (входными и выходными). Введем следующие определения.

О п р е д е л е н и е 1. Оператор $z(t) = \theta[x(t)]$ называется τ -сдвигом (τ — фиксированное натуральное число) оператора $z' = \theta'[x(t)]$, если для любой входной последовательности $x(1), x(2), \dots$ и для любого $t: z(t + \tau) = z'(t)$.

О п р е д е л е н и е 2. Оператор $z(t) = \theta[x(t)]$ называется σ -растяжением (σ — натуральное число) оператора $z'(t) = \theta'[x(t)]$, если существует буква x_0 такая, что если входная последовательность $\xi(1), \xi(2), \dots$ перерабатывается оператором θ' в выходную последовательность

$z'(1), z'(2), z'(3), \dots$, то последовательность

$$\xi(1) \underbrace{x_0 \dots x_0}_{(\sigma-1) \text{ раз}}, \quad \xi(2) \underbrace{x_0 \dots x_0}_{(\sigma-1) \text{ раз}}, \quad \dots$$

перерабатывается оператором в выходную последовательность

$$\dots, \underbrace{z'(1)}_{(\sigma-1) \text{ букв}}, \quad \dots, \underbrace{z'(2)}_{(\sigma-1) \text{ букв}}, \quad \dots, \underbrace{z'(3)}_{(\sigma-1) \text{ букв}}, \quad \dots$$

О п р е д е л е н и е 3. Будем говорить, что логическая сеть N реализует данный ограниченно-детерминированный оператор θ со сдвигом τ (в масштабе времени σ), если N реализует τ -сдвиг (вместе с σ -растяжением) оператора θ .

В соответствии с этим определением можно поставить задачу синтеза со сдвигом (в масштабе времени) и надлежащим образом истолковать понятие полноты множества элементов. Реализация оператора со сдвигом, а тем более в масштабе времени, в реальных устройствах связана с определенным замедлением процесса окончательной выработки преобразованной информации. Однако в определенных конкретных обстоятельствах это может оказаться выгодным и удобным. Забегая вперед, укажем на то, что логические сети, построенные на ферритах, применяются в основном для реализации ограниченно-детерминированных операторов со сдвигом и в масштабе времени.

ГЛАВА III

ОПЕРАТОРЫ ФИЗИЧЕСКИХ ЭЛЕМЕНТОВ

§ 1. Вводные замечания

В предыдущей главе было выяснено, что для построения логической сети необходимо иметь функционально полные наборы элементов, реализующих логические функции и временную задержку сигналов. В современных автоматических системах дискретного действия находят применение самые разнообразные наборы элементов. Они строятся на различных физических принципах, относящихся к электрическим, магнитным, механическим, химическим и другим явлениям, и реализуют различные элементарные функции, образуя функционально полные системы.

Использование разнообразных типов элементов обусловливается многими причинами. Важнейшее значение имеют такие характеристики элементов, как **б** **ы** **с** **т** **р** **о** **д** **е** **й** **с** **т** **в** **и** **е** и **н** **а** **д** **е** **ж** **н** **о** **с** **т** **ь**, а также **с** **т** **о** **и** **м** **о** **с** **т** **ь**, которую стремятся сделать как можно меньшей. Выбор тех или иных элементов зависит от назначения автомата и его технических параметров. Чаще всего в реальных системах одновременно используются элементы, основанные на различных физических принципах, но эквивалентные в отношении тех логических функций, которые они реализуют.

В настоящей книге мы ограничиваемся рассмотрением некоторых из физических элементов, получивших в настоящее время наиболее широкое практическое применение в дискретных автоматах. Физические основы излагаются в той мере, в какой это необходимо для понимания работы элементов в логической сети. Основное внимание уде-

ляется описанию операторов элементов и выяснению полноты системы элементов, реализуемых в устройствах, основанных на различных физических принципах. При этом истинностно-полные наборы элементов интерпретируются как реальные объекты, физически моделирующие те или иные полные системы функций алгебры логики, указанные в гл. I.

В § 2 описываются электронно-ламповые и полупроводниковые элементы, § 3 посвящен описанию триггеров, а в последнем параграфе рассматриваются магнитные элементы.

§ 2. Электронно-ламповые и полупроводниковые элементы

1) Физические основы *)

Электронно-ламповые логические схемы строятся в основном на *триодах* (рис. 3.1) и *пентодах* (рис. 3.2). На обоих рисунках E_+ — анодное напряжение, R_a — сопротивление в цепи анода лампы и E_{gi} — сеточные напряжения.

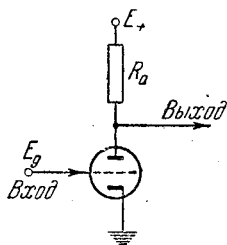


Рис. 3.1.

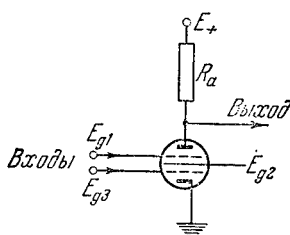


Рис. 3.2.

Входные сигналы поступают в виде импульсов электрического тока или в виде различных уровней потенциалов на сетки ламп, а соответствующие выходные сигналы образуются в анодной цепи лампы.

При построении логических схем на пентодах управление лампой осуществляется по первой и третьей сеткам.

*) См. А. М. Бонч-Бруевич, Применение электронных ламп в экспериментальной физике, Гостехиздат, М., 1956.

Токи этих сеток при отрицательных напряжениях на них практически равны нулю. Поэтому использование указанных сеток в качестве входных каналов требует малой мощности. Управление по второй (экранной) сетке сопряжено с потреблением большей мощности, и поэтому к нему прибегают крайне редко.

Междукаскадная связь ламповых элементов в логической сети, работающей на сигналах, реализуемых различными уровнями потенциалов, осуществляется посредством

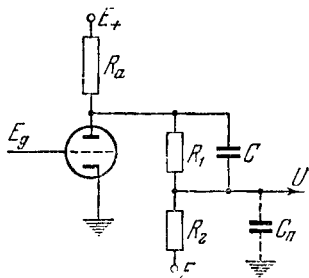


Рис. 3.3.

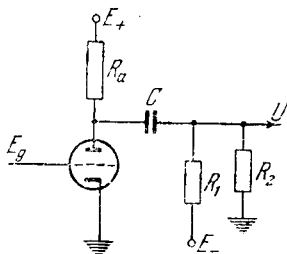


Рис. 3.4.

реостатных звеньев (рис. 3.3). В логической сети, работающей на импульсах, междукаскадные связи осуществляются емкостными звеньями (рис. 3.4).

Потенциал на выходе реостатного звена при запертой лампе определяется следующим соотношением:

$$U_{\text{в}} = \frac{E_+ + E_-}{R_a + R_1 + R_2} R_2 - E_-, \quad (3.1)$$

а когда лампа открывается, потенциал на выходе падает и принимает значение

$$U_{\text{н}} = \frac{E_- + \frac{R_i}{R_a + R_i} E_+}{R_1 + R_2 + \frac{R_a R_i}{R_a + R_i}} R_2 - E_-, \quad (3.2)$$

где R_i — внутреннее сопротивление лампы. При крутом фронте изменения уровня потенциала паразитная емкость компенсируется конденсатором C в цепи реостатного звена.

В емкостном звене при запираании лампы происходит заряд конденсатора C через сопротивление R_a и сопротивления R_1 и R_2 (общее сопротивление которых в силу их параллельного включения равно $\frac{R_1 R_2}{R_1 + R_2}$) до величины напряжения $E_+ + \frac{E_- \cdot R_2}{R_1 + R_2}$. При этом изменение зарядного тока носит экспоненциальный характер. Когда лампа отпирается, конденсатор C разряжается через сопротивления $\frac{R_a R_1}{R_a + R_1}$ и $\frac{R_1 R_2}{R_1 + R_2}$. Если постоянная времени заряда и разряда много меньше периода поступления импульсов от

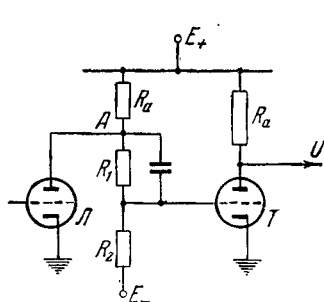


Рис. 3.5.

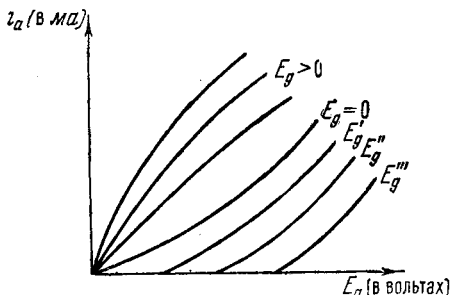


Рис. 3.6.

лампы, то изменение разности потенциалов на конденсаторе в течение периода практически не вызывает искажения импульсов при их передаче.

Рассмотрим теперь реализацию элементарных логических операций на ламповых схемах.

Оператор отрицания осуществляется посредством электронно-ламповой схемы, называемой *инвертором*. Она обычно строится на триоде. На рис. 3.5 представлена схема подключения инвертора к источнику входного сигнала — выходу лампы \mathcal{L} .

Заметим, что работа триода в схемах дискретного действия определяется двумя семействами характеристик: 1) зависимостью анодного тока i_a от напряжения на аноде E_a при различных значениях сеточного напряжения E_g . Примерный характер кривых $i_a = f(E_a)$ представлен на рис. 3.6;

2) зависимостью сеточного тока i_g от напряжения на сетке при различных анодных напряжениях. На рис. 3.7 показаны некоторые из кривых семейства $i_g = \varphi(E_g)$.

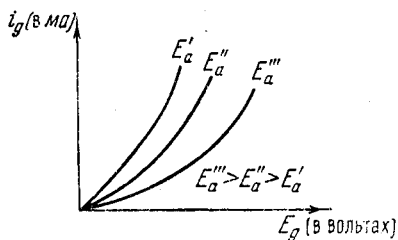


Рис. 3.7.

Вернемся к схеме, изображенной на рис. 3.5. Когда лампа L запerta, потенциал в точке A приблизительно равен напряжению E_+ , сопоставляемому со значением 1 двоичной переменной. При этом на сетке триода T устанавливается небольшое поло-

жительное напряжение E'_g , обусловленное сеточным током. Оно легко определяется с помощью эквивалентной

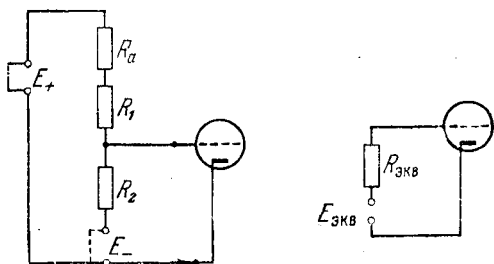


Рис. 3.8.

схемы, изображенной на рис. 3.8. По ней вычисляем

$$E_{\text{экв}} = \frac{E_+ + E_-}{R_a + R_1 + R_2} R_2 - E_- \quad (3.3)$$

и

$$R_{\text{экв}} = \frac{R_1 + R_2}{R_a + R_1 + R_2} R_2. \quad (3.3')$$

Теперь по сеточной характеристике триода, взятой для $E_a = E_+$, находим напряжение на его сетке. Из точки с абсциссой $E_a = E_{\text{экв}}$ (рис. 3.9) проводим прямую под углом α ,

определяемым соотношением

$$\operatorname{tg} \alpha = \frac{1}{R_{\text{экв}}}.$$

Абсцисса точки b пересечения этой прямой с сеточной характеристикой триода определяет сеточное напряжение E'_g . При положительном сеточном напряжении триод T открыт и на его аноде устанавливается низкий выходной

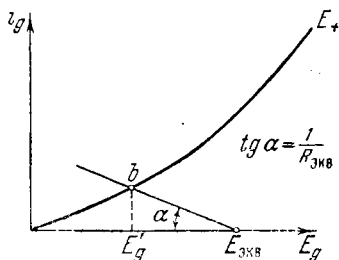


Рис. 3.9.

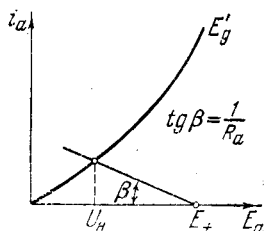


Рис. 3.10.

потенциал U_n , величина которого определяется по анодной характеристике лампы (рис. 3.10). Ему сопоставляется нулевое значение двоичной переменной. Если лампа L открыта, то потенциал точки A равен U_n , т. е. соответствует нулевому значению двоичной переменной. Тогда потенциал на сетке триода равен

$$E_g'' = \frac{U_n + E_-}{R_1 + R_2 + \frac{R_a R_i}{R_a + R_i}} R_2 - E_- \quad (3.4)$$

Параметры схемы выбираются так, чтобы E_g'' было ниже потенциала запираения триода. Тогда на аноде запертой лампы T устанавливается высокий потенциал $U_b = E_+$, соответствующий значению 1 двоичной переменной.

Как уже отмечалось, в электронно-ламповых схемах наряду с триодами широкое применение находят пентоды с управлением по первой и третьей сеткам. Для наших целей достаточно ограничиться указанием на то, что управление по каждой из сеток осуществляется независимо, и при выборе режимов управления и параметров

междукаскадных связей здесь не возникает существенных отличий сравнительно со схемами на триодах.

Схема включения пентода представлена на рис. 3.11. Когда лампа заперта хотя бы по одной из управляющих сеток (т. е. когда соответствующая двоичная переменная равна нулю), ток через сопротивление R_a не течет, и потенциал на выходе Z близок к E_+ , т. е. соответствует значению 1 двоичной переменной. При одновременной подаче вы-

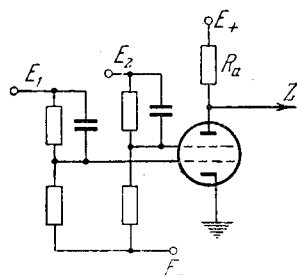


Рис. 3.11.

сокого потенциала на обе сетки лампа открывается и выходной потенциал становится равным U_n .

В качестве усилителя мощности в электронно-ламповых схемах обычно используется *катодный повторитель* (рис. 3.12). В нем управляющее напряжение между сеткой и катодом $U_{гк}$ равно разности между входным E_g и выходным U напряжениями. Если подать на сетку катодного повторителя положительное напряжение

(относительно земли), то посредством небольшого управляющего напряжения можно получить на выходе напряжение, почти равное входному.

На рис. 3.13 представлена схема *двойного катодного повторителя* с общей катодной нагрузкой R_k . В силу

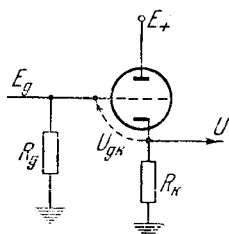


Рис. 3.12.

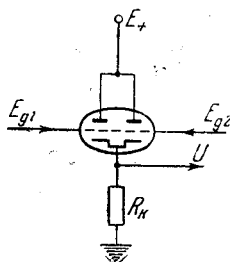


Рис. 3.13.

указанного выше свойства катодного повторителя в этой схеме при подаче высокого потенциала E_+ (соответствующего значению 1 двоичной переменной) на одну из сеток

выходной потенциал принимает ту же величину. Если высокие напряжения поданы на оба входа одновременно, то выходной потенциал несколько повышается. Но при надлежащем выборе сопротивления R_K это превышение не влияет на работу схемы в дискретном режиме.

В современных схемах дискретного действия ламповые элементы применяются только в сочетании с элементами других типов. При этом лампы наряду с логическими функциями часто выполняют роль усилителей мощности. Широкое использование в устройствах дискретного действия находят логические сети, построенные на ламповых и кристаллических диодных элементах.

Кристаллические или ламповые диоды удобны для реализации схем конъюнкции и дизъюнкции m двоичных переменных. Схема образования конъюнкции m переменных представлена на рис. 3.14. Ее работа основана на том, что сопротивление R намного больше сопротивления диода в прямом направлении (обозначим его через $R_{\text{пд}}$), т. е. $R \gg R_{\text{пд}}$. На входы 1, 2, 3, ..., m могут поступать два уровня напряжений: высокий потенциал E_+ , соответствующий значению 1 входной переменной, и низкий потенциал E_- , соответствующий ее нулевому значению. Когда на все входы одновременно поступают высокие напряжения в виде импульсов или положительных перепадов входных потенциалов, то разность потенциалов на концах диодов обращается в нуль. При этом прохождение тока через сопротивление R прекращается, и на выходе схемы устанавливается высокий потенциал $U_{\text{в}} = E_+$, соответствующий значению 1 выходной переменной.

Если же хотя бы на одном из входов схемы напряжение остается низким, т. е. разность потенциалов на концах соответствующего диода равна $E_+ - E_-$, то через сопротивление R протекает ток. Возникающее на нем падение напряжения обуславливает низкий выходной потенциал $U_{\text{н}}$.

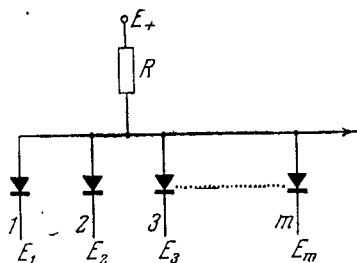


Рис. 3.14.

Так как $R \gg R_{\text{цд}}$, то значение этого потенциала равно примерно $E_{\text{н}}$, т. е. соответствует значению 0 выходной переменной.

Для обеспечения надежной работы диодной схемы приходится принимать во внимание различные дополнительные физические условия. При их рассмотрении удобно

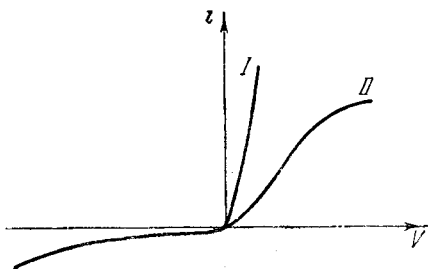


Рис. 3.15.

пользоваться вольтамперной характеристикой диода, т. е. зависимостью тока i , протекающего через диод от разности V напряжений, приложенных к его концам. Примерный вид вольтамперных характеристик диодов — кристаллического (кривая I) и лампового (кривая II) — представлен на рис. 3.15.

При построении диодной схемы, включающей большое количество элементов, нужно учитывать влияние прямого сопротивления диода. Так как $R_{\text{цд}} \neq 0$, то при входном потенциале, равном $E_{\text{ц}}$, выходное напряжение определяется следующим соотношением:

$$U = U_{\text{н}} + \Delta'U. \quad (3.5)$$

Величина $\Delta'U$ добавочного напряжения на выходе зависит от сопротивления $R_{\text{цд}}$ и числа m_0 открытых диодов:

$$\Delta'U = f(R_{\text{цд}}, m_0).$$

Влияние m_0 на $\Delta'U$ нетрудно установить с помощью вольтамперной характеристики диодной схемы. На рис. 3.16 показаны эти характеристики для схемы с одним (кривая 3), двумя (кривая 2) и тремя (кривая 1) открытыми диодами. Нагрузочная прямая, наклоненная под углом

$\alpha = \arctg \frac{1}{R}$ к оси абсцисс и проведенная через точку $E_+ - E_n$, пересекает характеристики в точках a_1 , a_2 и a_3 .

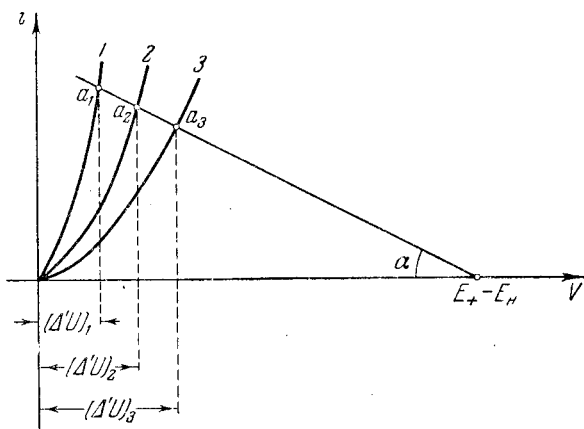


Рис. 3.16.

Абсциссы этих точек определяют величину добавочного напряжения для всех трех рассматриваемых случаев. Как видно из рисунка, $\Delta'U$ падает с увеличением числа открытых диодов. Следовательно, уровень выходного напряжения зависит от числа одновременно поданных на вход единичных сигналов. Этот разброс не вызывает опасных последствий в том случае, если выходной сигнал не превышает порога срабатывания последующей схемы.

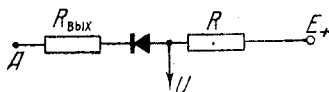


Рис. 3.17.

Отметим далее, что любой реальный (конечной мощности) источник входных сигналов, поступающих на диодную схему, имеет выходное сопротивление ($R_{\text{вых}}$), отличное от нуля (рис. 3.17). Оно также влияет на работу диодной схемы. Когда потенциал в точке А равен E_+ , выходное напряжение $U \approx E_+$. Если же потенциал в точке А равен U_n , то

$$U = U_n + \Delta'U + \Delta''U, \quad (3.6)$$

причем

$$\Delta'' U = \frac{E_+ - U_H}{R + R_{\text{пд}} + R_{\text{вых}}} \cdot R_{\text{вых}}. \quad (3.7)$$

При $R_{\text{пд}} \ll R$ величина добавочного выходного напряжения $\Delta'' U$ определяется отношением $\frac{R_{\text{вых}}}{R}$. Величина $R_{\text{вых}}$ зависит от мощности источника сигналов. Для большинства современных триодов с анодной нагрузкой, являющихся источником входных сигналов диодной схемы, $R_{\text{вых}}$ лежит в пределах 3—10 *ком*.

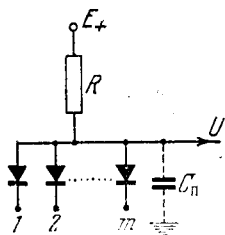


Рис. 3.18.

Для уменьшения разброса низкого выходного напряжения следует увеличивать сопротивление R . Однако эта возможность ограничивается наличием паразитных емкостей в схеме (ср. рис. 3.18). При запираании всех диодов начинается заряд паразитной емкости $C_{\text{п}}$ через сопротивление R до потенциала E_+ . Когда ка-

кой-либо диод открывается, емкость разряжается через него и выходное сопротивление соответствующего источника сигнала. Для того чтобы при подаче входных импульсов длительностью τ выходное напряжение достигло требуемого высокого уровня, необходимо обеспечить следующее соотношение:

$$3RC_{\text{п}} < \tau, \quad (3.8)$$

которое и приводит к ограничению величины сопротивления R . Это ограничение обуславливается вместе с тем влиянием на выходное напряжение обратного сопротивления диода $R_{\text{од}}$. Увеличение сопротивления R приводит к усилению этого влияния. Указанные соображения приводят к необходимости обеспечения следующих соотношений при реализации схемы конъюнкции на диодах, кристаллических или ламповых:

$$\left. \begin{aligned} R_{\text{од}} &\gg R \gg R_{\text{пд}}, \\ RC_{\text{п}} &< \frac{\tau}{3}, \\ R &> R_{\text{вых}}. \end{aligned} \right\} \quad (3.9)$$

Они определяют допустимое число входных переменных в схеме конъюнкции.

На рис. 3.19 представлена схема дизъюнкции на диодных элементах. Здесь, как и в предыдущей схеме, на каждый вход поступают двоичные сигналы: 0, соответствующий низкому входному напряжению E_n , и 1, соответствующая напряжению E_+ . При $R \gg R_{\text{пд}}$ подача напряжения E_+ хотя бы на один из входов вызывает такое же напряжение на выходе. Однако в реальных условиях работа схемы дизъюнкции зависит от тех же факторов, которые влияют на работу схемы конъюнкции.

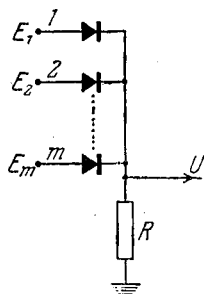


Рис. 3.19.

За счет падения напряжения на прямом сопротивлении диода даже при нулевом сопротивлении источника сигнала высокий уровень выходного напряжения отличается от E_+ :

$$U = E_+ - \Delta' U. \quad (3.10)$$

Сопротивление источника приводит к дополнительному снижению выходного потенциала, обусловленному падением напряжения на $R_{\text{вых}}$. Оно увеличивается с увеличением сопротивления вследствие того, что (рис. 3.20)

$$\Delta'' U = \frac{E_+}{R_{\text{вых}} + R_{\text{пд}} + R} R_{\text{вых}}. \quad (3.11)$$

Наконец, ΔU зависит от числа одновременно поданных на вход схемы «единиц».

Уменьшение ΔU во всех указанных случаях может быть достигнуто увеличением R . Однако, как и в схеме конъюнкции, оно ограничивается паразитной емкостью и обратным сопротивлением диодов. Для надежной работы схемы дизъюнкции необходимо обеспечить указанные

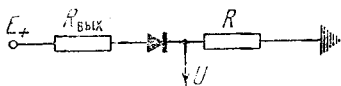


Рис. 3.20.

выше соотношения (3.9). Кроме того, здесь приходится вводить усилители мощности для компенсации потери мощности выходного сигнала.

Из всего сказанного выше следует, что при построении сложных многокаскадных сетей на диодных элементах необходим тщательный анализ схемы реализации сети и обеспечение соотношений типа (3.9) для конкретных условий работы.

В последние годы в качестве элементов логических сетей все более широкое применение получают *кристаллические триоды* — *точечные* и *плоскостные*. Первые полупроводниковые приборы строились на точечных триодах; плоскостные триоды появились несколько позже.

Триоды обоих типов отличаются друг от друга своими физическими параметрами (рабочее напряжение; шумы, порождаемые триодом; коэффициент полезного действия, и др.), а также технологией производства и режимами работы. Выбор того или иного типа триодов обуславливается конкретными условиями их использования в качестве логических элементов автомата. Рассмотрение этого вопроса выходит за рамки настоящей книги, поэтому мы ограничимся здесь только некоторыми общими замечаниями, относящимися к триодам обоих типов.

Кристаллический триод, как и ламповый, имеет три электрода: *коллектор* — элемент, аналогичный аноду в ламповых схемах, — улавливает носители электрических зарядов, *эмиттер* — элемент, аналогичный катоду, — является источником носителей электрических зарядов, *база* (о с н о в а н и е) кристаллического триода — элемент, аналогичный сетке лампового триода, — управляет потоком носителей электрических зарядов через триод.

Наличие трех электродов позволяет реализовать три схемы включения кристаллических триодов в электрическую цепь (см. рис. 3.21, где буквой *K* обозначен коллектор, буквой *Э* — эмиттер, а буквой *Б* — база):

а) схема с заземленным эмиттером (рис. 3.21, а), который соответствует включению лампового триода по схеме с заземленным катодом; нагрузочное сопротивление включается в цепь коллектора, а входной сигнал подается на базу;

б) схема с заземленной базой (рис. 3.21, б), аналогичная схеме включения лампового триода с заземленной сеткой; нагрузочное сопротивление, как и в предыдущей схеме,

включается в цепь коллектора, а входной сигнал поступает на эмиттер;

в) схема с заземленным коллектором (рис. 3.21, в), соответствующая схеме катодного повторителя, в которой заземлен анод лампового триода; нагрузочное сопротивление

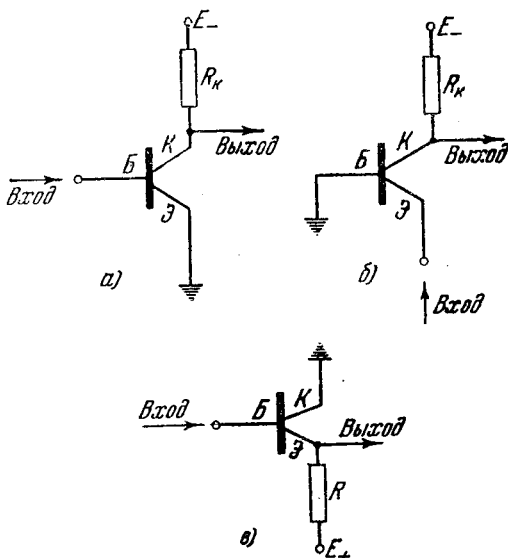


Рис. 3.21.

включается в цепь эмиттера, а входной сигнал подается на базу.

Вольтамперные характеристики кристаллического триода при различных схемах включения существенно отличаются друг от друга. В дискретных схемах наибольшее распространение получили схемы с заземленным эмиттером. Они требуют меньшей величины тока для переключения и обеспечивают значительные удобства при последовательном каскадировании элементов логической сети. Это обусловливается тем, что выходной сигнал предыдущего каскада достаточен для управления последующими каскадами.

В многокаскадных схемах, построенных на кристаллических триодах, применяются различные типы звеньев связи между каскадами, аналогичные звеньям в схемах на ламповых элементах. Вместе с тем, при использовании плоскостных триодов с заземленным эмиттером возникает возможность непосредственной связи каскадов между собой. Эти схемы обладают рядом преимуществ: в них почти

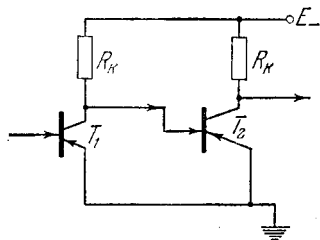


Рис. 3.22.

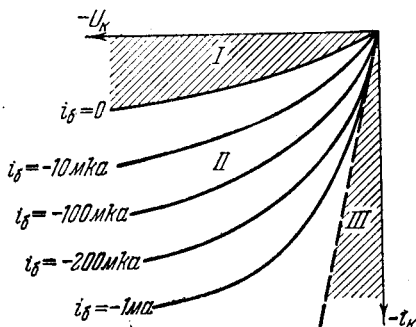


Рис. 3.23.

отсутствуют накопители энергии (конденсаторы и индуктивности), мала потребляемая мощность и сравнительно невелико количество сопротивлений.

В функциональном отношении оба типа кристаллических триодов — точечные и плоскостные — неразличимы и допускают реализацию одних и тех же элементарных логических операций. Поскольку в нашу задачу не входит подробное изложение всех аспектов построения дискретных схем, мы в целях упрощения изложения ограничимся рассмотрением схем, реализуемых из плоскостных триодов с непосредственной связью между элементами.

На рис. 3.22 представлена схема инвертора, построенная на двух триодах с непосредственной связью между коллектором триода T_1 и базой триода T_2 . Для анализа работы схемы воспользуемся двумя семействами характеристик триода: зависимостью $i_K = f(U_K)$ коллекторного тока от напряжения на коллекторе при постоянстве тока базы ($i_B = \text{const}$) и зависимостью $i_B = \varphi(U_6)$ тока базы от напряжения на базе при $U_K = \text{const}$. Примерный характер кривых первого семейства представлен на рис. 3.23. Здесь можно

выделить три области, соответствующие основным режимам работы триода. Во-первых, это область *I*, называемая *областью отсечки*, которая соответствует малому коллекторному току и большим значениям напряжения; во-вторых, область *II* — *область линейного режима усиления* и, в-третьих, область *III* — *область насыщения*, соответствующая большим значениям коллекторного тока при малых значениях напряжения. При использовании триода в качестве логического элемента рабочими областями являются *I* и *III*. Величина управляющего сигнала должна обеспечить перевод триода из режима отсечки в режим насыщения и наоборот. Примерный характер кривых $i_b = \varphi(U_b)$ при $U_k = \text{const}$ показан на рис. 3.24.

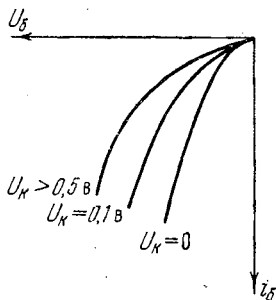


Рис. 3.24.

Вернемся к анализу работы инвертора, построенного на двух триодах с непосредственной связью (см. рис. 3.22).

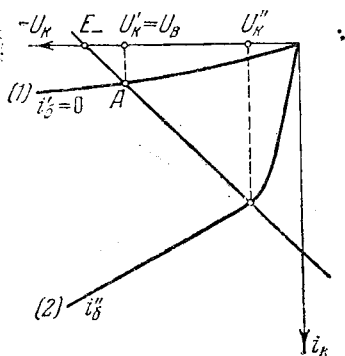


Рис. 3.25.

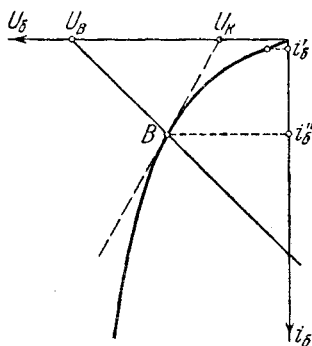


Рис. 3.26.

Предположим, что триод T_1 заперт, т. е. ток, протекающий через его базу, равен или близок к нулю. Проведем через точку E_- (рис. 3.25) нагрузочную прямую под углом α к оси абсцисс, определяемым сопротивлением R_k первого триода и входным сопротивлением последую-

щего каскада. Точка A пересечения этой прямой с характеристикой (1) определяет напряжение $U'_K = U_B$ на коллекторе триода T_1 . Оно соответствует значению 1 двоичной переменной. Это напряжение оказывается приложенным к базе триода T_2 . С помощью характеристики $i_b = \varphi(U_b)$ (рис. 3.26) находим ток базы второго триода. Для этого через точку U_B проводим нагрузочную прямую под углом к оси абсцисс, определяемым соотношением $\operatorname{tg} \alpha = \frac{R_K R_T}{R_K + R_T}$, где R_T — сопротивление триода. Ордината точки B определяет искомое значение тока. Теперь с помощью характеристики (2) (рис. 3.25) находим напряжение U''_K на коллекторе триода T_2 . Оно соответствует нулевому значению выходной двоичной переменной. С помощью тех же характеристик нетрудно установить, что при отпертом триоде T_1 , т. е. в случае, когда напряжение, приложенное к базе триода T_2 , соответствует нулевому значению двоичной переменной, выходное напряжение будет высоким, т. е. соответствует значению 1 двоичной переменной.

2) Реализация элементарных логических операторов

В логической сети, построенной на электронно-ламповых и полупроводниковых элементах, длительность такта подачи внешней информации обычно велика сравнительно с временем срабатывания элементов. Поэтому можно считать, что независимо от числа каскадов сети выходные сигналы возникают одновременно с входными.

Операторы элементов, рассмотренных в предыдущем разделе, позволяют построить несколько полных систем элементарных функций. Каждая из них допускает реализацию любого истинностного оператора в единичном масштабе времени, а при указанных выше допущениях — без временного сдвига.

При изложении физических основ работы элементов было установлено, что триод — ламповый или полупроводниковый — реализует функцию отрицания. При положительном потенциале на сетке (или базе) триода, сопоставленном значению 1 входной двоичной переменной, напряжение на выходе имеет низкий уровень, т. е. соответствует нулевому значению выходной переменной.

И наоборот, нулевому значению входной переменной, сопоставленному потенциалу запирающей сетки (или базы), соответствует высокий уровень напряжения на выходе триода, т. е. значение 1 выходной переменной. Таким образом, оператор триода имеет вид:

$$T(x) = \bar{x}. \quad (3.12)$$

В двойном катодном повторителе при подаче на одну из сеток положительного сигнала, соответствующего значению 1 входной переменной, выходная переменная при-

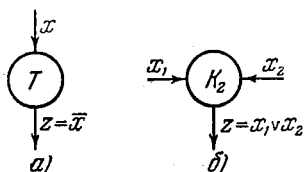


Рис. 3.27

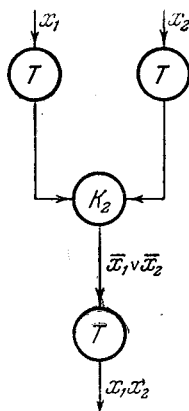


Рис. 3.28.

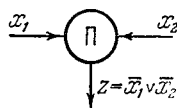


Рис. 3.29.

нимает то же значение. Следовательно, этот элемент реализует дизъюнкцию входных переменных; оператор двойного катодного повторителя имеет вид:

$$K_2(x_1, x_2) = x_1 \vee x_2, \quad (3.13)$$

а оператор одинарного катодного повторителя: $K(x) = x$. Условные схемы лампового триода и двойного катодного повторителя показаны на рис. 3.27, а, б.

Операторы указанных элементов образуют полную систему элементарных функций в силу того, что набор, составленный из отрицания и дизъюнкции, является функционально полным. На рис. 3.28 приведена схема реализации конъюнкции на этих элементах.

Функционально полную систему дает также оператор пентода, условная схема которого представлена на рис. 3.29. Примем во внимание сделанное в разделе 1) настоящего параграфа замечание об аналогичном характере режима работы триода и пентода при управлении по каждой из сеток и опишем его оператор таблицей III.1.

Таблица III.1

x_1	x_2	z
0	0	1
0	1	1
1	0	1
1	1	0

Из этой таблицы мы получаем, что $z = \bar{x}_1 \vee \bar{x}_2$. Следовательно, оператором пентода

$$\Pi(x_1, x_2) = \bar{x}_1 \vee \bar{x}_2 \quad (3.14)$$

является функция Шеффера, которая согласно § 3 гл. I образует полную систему. На рис. 3.30 показаны схемы

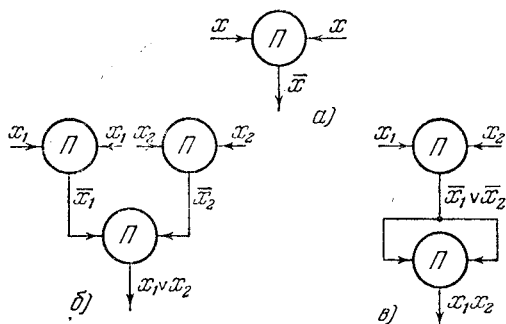


Рис. 3.30.

отрицания (а), дизъюнкции (б) и конъюнкции (в), построенные на пентодах.

Как мы отмечали выше, полные системы элементарных функций можно строить и на полупроводниковых элементах. Оператор отрицания в ней реализуется полупро-

водниковым триодом, условная схема которого показана на рис. 3.31, а дизъюнкция и конъюнкция переменных

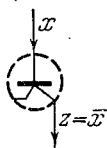


Рис. 3.31.

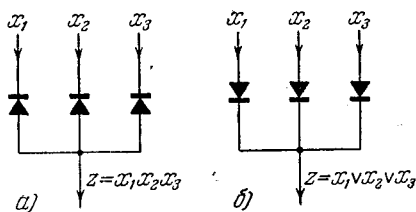


Рис. 3.32.

реализуются диодными схемами (рис. 3.32). Для операторов указанных диодных схем мы примем следующие обозначения:

$$\left. \begin{aligned} D(x_1, x_2, \dots, x_m) &= x_1 \vee x_2 \vee \dots \vee x_m, \\ K(x_1, x_2, \dots, x_m) &= x_1 x_2 \dots x_m. \end{aligned} \right\} \quad (3.15)$$

Допустимое число входных переменных в этих функциях ограничивается условиями, указанными в предыдущем разделе.

Рассмотренные полные системы элементарных функций имеют скорее теоретическое, чем практическое значение. Естественно, что схемы образования конъюнкции и дизъюнкции, приведенные на рис. 3.28 и 3.30, которые требуют для своей реализации значительного числа элементов, не находят практического применения при построении логических сетей. Последние, как указывалось выше, обычно строятся в виде комбинированных схем с использованием ламповых и полупроводниковых элементов, причем выбор тех или иных элементов, образующих полную систему, зависит от конкретных условий синтеза сети.

В логических сетях, реализуемых на электронно-ламповых и полупроводниковых элементах, выходные каналы часто объединяют с целью получения на общем выходе сети сигнала, зависящего от сигналов, идущих по объединенным каналам. В гл. II мы указывали, что объединение выходных каналов логической сети, допустимое физическими условиями ее реализации, может рассматриваться

как включение в схему «фиктивных» ячеек. В электронно-ламповых и полупроводниковых схемах «фиктивные» ячейки реализуются путем подключения элементов к общей нагрузке.

«Фиктивная» ячейка Φ_k , осуществляющая к о н т ю н к ц и ю переменных, вырабатываемых ламповыми трио-

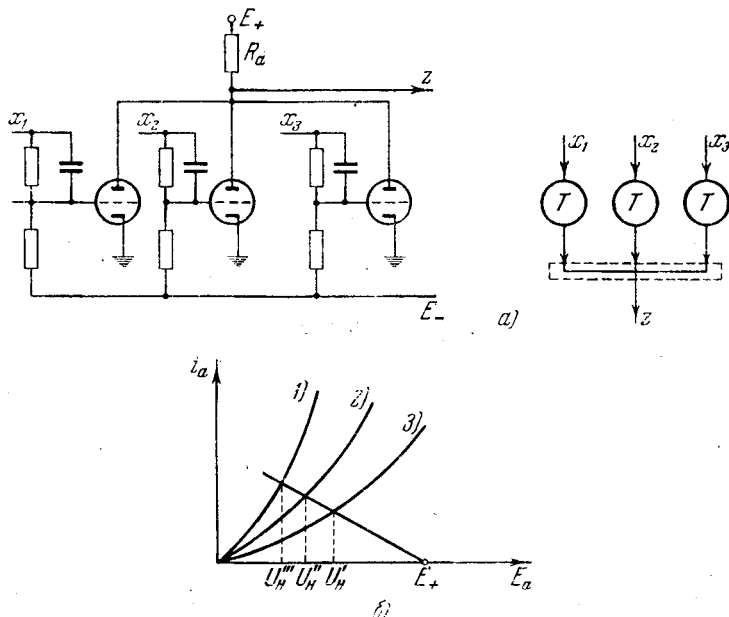


Рис. 3.33.

дами, пентодами и, как было показано выше, диодами, реализуется обобществлением их анодного сопротивления. «Фиктивная» ячейка Φ_d , осуществляющая д и з ъ ю н к ц и ю переменных, вырабатываемых катодными повторителями или диодами, реализуется обобществлением их катодной нагрузки. Такие же «фиктивные» ячейки Φ_k и Φ_d могут быть реализованы путем, соответственно, параллельного и последовательного подключения полупроводниковых триодов к общей нагрузке.

На рис. 3.33, а представлена схема (и ее условное обозначение), образованная несколькими ламповыми трио-

дами, подключенными к общей анодной нагрузке. Пунктирным прямоугольником условно показана «фиктивная» ячейка Φ_k , осуществляющая конъюнкцию сигналов, идущих по выходным каналам триодов. Как видно из схемы, на ее выходе устанавливается высокий потенциал, соответствующий $z=1$, только в том случае, когда все триоды заперты, т. е. при $x_i=0$. Если открыта хотя бы одна лампа, то через сопротивление R_a течет анодный ток и потенциал на выходе становится равным U_n , которому соответствует $z=0$.

Практически, как это видно из анодной характеристики (рис. 3.33, б), потенциал на выходе зависит от числа одновременно открытых ламп (кривая 3) соответствует случаю, когда открыта одна лампа, кривая 2) — случаю, когда открыты две лампы, а кривая 1) — случаю, когда открыты три лампы). Однако это не влияет на работу схемы, так как с увеличением числа отпертых триодов потенциал на выходе уменьшается сравнительно с его величиной, соответствующей нулевому значению z .

Таким образом, сигнал, вырабатываемый на выходе рассматриваемой схемы, можно считать конъюнкцией отрицаний входных переменных:

$$z = T(x_1) T(x_2) T(x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3. \quad (3.16)$$

Оператор таких схем мы будем обозначать через $T(x_1, x_2, \dots, x_m)$.

На рис. 3.34 приведена аналогичная схема, построенная на полупроводниковых триодах. В этой схеме потенциал точки А будет низким относительно земли, если отперт хотя бы один триод. Допустимое число входных переменных в рассматриваемой схеме определяется условиями использования выходного сигнала. Так, например, если он поступает на полупроводниковый триод для инвертирования и получения дизъюнкции ($\vee x_i$), то число

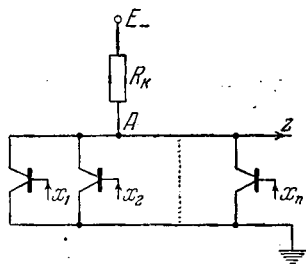


Рис. 3.34.

выходных переменных n определяется соотношением (см. предыдущий раздел)

$$E_- - i'_0 R_K n \geq U_0.$$

Оно вытекает из того условия, что при всех запертых триодах напряжение в точке A , поступающее на базу инвертора, должно быть больше напряжения (U_0), обеспечивающего его отпирание.

На рис. 3.35 дана схема, полученная путем последовательного подключения n полупроводниковых триодов к общей нагрузке. Она реализует дизъюнкцию переменных, вырабатываемых каждым из триодов, т. е.

$$z = T(x_1) \vee T(x_2) \vee \dots \vee T(x_n) = \bar{x}_1 \vee \bar{x}_2 \vee \dots \vee \bar{x}_n. \quad (3.17)$$

В ней выходной потенциал в точке A будет низким (относительно земли) только в том случае, когда все триоды отперты, т. е. когда для любого i ($i=1, 2, \dots, n$) $x_i=0$. Легко заметить, что эта схема при двух триодах реализует функцию Шеффера и аналогична ламповому пентоду. Оператор схемы, изображенной на рис. 3.35, мы обозначим через $\Pi_n(x_1, x_2, \dots, x_n)$. Максимальное число входных переменных в схеме определяется, как и в предыдущем случае, условиями использования выходного сигнала.

В заключение параграфа выясним возможности сочетания «фиктивных» ячеек и допустимые способы их включения в схему логической сети. Рассмотрим вначале сочетание однотипных ячеек. Пусть Φ'_K и Φ''_K — «фиктивные» ячейки с m входными и одним выходным полюсом каждая. На выходах ячеек вырабатываются соответственно конъюнкции

$$z' = x'_1 x'_2 \dots x'_k \dots x'_j \dots x'_m,$$

$$z'' = x''_1 x''_2 \dots x''_k \dots x''_j \dots x''_m$$

сигналов, поступающих к их входным полюсам от электронно-ламповых или диодных элементов сети (рис. 3.36, а). Найдем оператор, реализуемый сетью, полученной

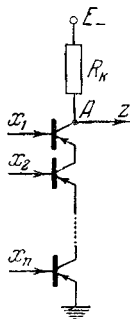


Рис. 3.35.

путем подключения какого-либо из входных каналов x'_k или выходного канала z' ячейки Φ'_K к входному полюсу x''_j ячейки Φ''_K . Отметим, что в силу физических условий реализации фиктивных ячеек подключение любого из входных или выходного канала ячейки Φ'_K к любому из полюсов ячейки Φ''_K эквивалентно отождествлению полюсов

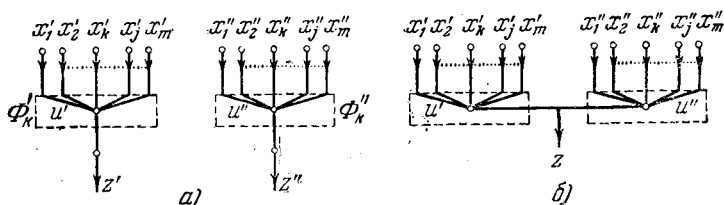


Рис. 3.36.

u' и u'' . При этом образуется сеть, представленная на рис. 3.36, б, которая вырабатывает функцию

$$z = z' z''.$$

Аналогичным образом можно показать, что сочетания «фиктивных» ячеек Φ'_K и Φ''_K , каждая из которых вырабатывает дизъюнкции

$$z' = x'_1 \vee x'_2 \vee \dots \vee x'_k \vee \dots \vee x'_j \vee \dots \vee x'_m,$$

$$z'' = x''_1 \vee x''_2 \vee \dots \vee x''_k \vee \dots \vee x''_j \vee \dots \vee x''_m$$

сигналов, поступающих к их входным узлам, реализуют функцию

$$z = z' \vee z''.$$

Таким образом, сочетание однотипных ячеек допустимо, и при этом образуется обобщенная ячейка того же типа с числом входных каналов, равным сумме входных каналов сочетаемых ячеек.

Легко установить, что при сочетании разнотипных «фиктивных» ячеек нарушается правильная работа сети. Пусть, для определенности, фиктивные ячейки Φ_K и Φ_D вырабатывают соответственно функции

$$z' = x'_1 x'_2 x'_3,$$

$$z'' = x''_1 \vee x''_2 \vee x''_3$$

сигналов, поступающих от диодных элементов. На рис. 3.37, а показаны упрощенные электрические схемы реализации конъюнкции и дизъюнкции. Подключение любого из каналов ячейки Φ_k к любому из полюсов ячейки Φ_d или наоборот равносильно, как и в предыдущих случаях, отождествлению узлов u' и u'' и приводит к схеме,

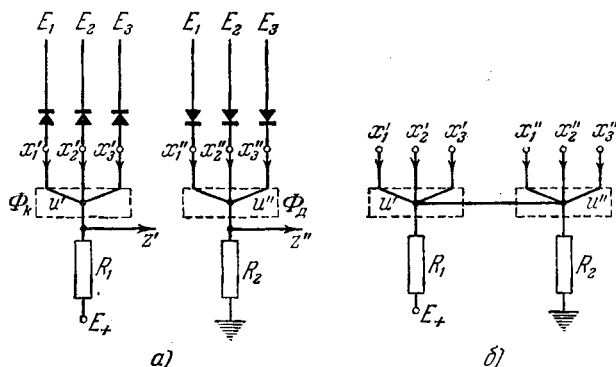


Рис. 3.37.

показанной на рис. 3.37, б. Как видно на рисунке, при этом в схеме образуется прямая электрическая цепь от источника питания к земле.

Для того чтобы исключить взаимное влияние «фиктивных» ячеек и избежать образования прямых электрических цепей, следует не допускать также подключения входных каналов «фиктивных» ячеек к внешним узлам логической сети, от которых информация обычно поступает к входным каналам многих элементов.

§ 3. Триггерные элементы

1) Физические основы работы триггера*)

Триггером обычно называют электронное реле с двумя устойчивыми состояниями равновесия, причем переход из одного равновесного состояния в другое осуществляет-

*) См. Л. А. Меерович, Л. Г. Зеличенко, Импульсная техника. Советское радио, М., 1952; А. М. Бонч-Бруевич, Применения электронных ламп в экспериментальной физике, Гостехиздат, М., 1956

ся под воздействием внешнего сигнала. Триггер может быть построен на вакуумных электронных лампах, тиристорах, полупроводниковых триодах и других электронных элементах. Схемы, действующие аналогично триггеру, могут быть реализованы также на ферромагнитных элементах.

Триггеры наряду с истинностными элементами и элементами задержки находят широкое применение в логических сетях, реализующих ограниченно-детерминированные операторы. Их использование в схемах накопления и хранения информации обеспечивает существенную экономию числа элементов и большие удобства в управлении работой этих схем.

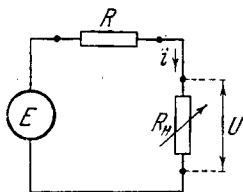


Рис. 3.38.

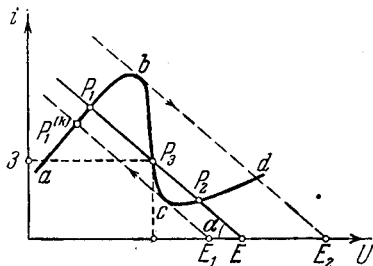


Рис. 3.39.

Рассмотрим электрическую цепь, представленную на рис. 3.38 и включающую источник напряжения E , линейное сопротивление R и нелинейное сопротивление R_n . Пусть вольтамперная характеристика этого нелинейного элемента, т. е. зависимость между током i , протекающим

через него, и разностью напряжений U на его зажимах имеет вид, представленный на рис. 3.39 (кривая $abcd$). Участок bc характеристики имеет отрицательный наклон. В этой области элемент R_n обладает отрицательным сопротивлением, так как с увеличением разности напряжений на его концах ток i уменьшается. Величина тока, протекающего в рассматриваемой электрической цепи, определяется соотношениями:

$$\begin{aligned} U &= E - iR, \\ i &= f(U) \end{aligned} \quad (3.18)$$

и легко находится графическим путем. Для этого через

точку E следует провести нагрузочную прямую под углом $\alpha = \operatorname{arctg} \frac{1}{R}$ к оси абсцисс. Точки P_1 , P_2 и P_3 пересечения этой прямой с характеристикой нелинейного элемента определяют значения токов в схеме при равновесном ее состоянии. Точкам P_1 и P_2 соответствуют состояния устойчивого равновесия, а точка P_3 является точкой неустойчивого равновесия. Мы не будем здесь проводить строгого доказательства этого утверждения, а ограничимся лишь качественным анализом работы цепи.

Предположим, что схема находится в равновесном состоянии P_3 . При этом падение напряжения на нелинейном элементе равно некоторой величине, которую мы обозначим через U_3 , а ток, протекающий через этот элемент, пусть будет равен i_3 . Тогда малое увеличение тока будет сопровождаться уменьшением падения напряжения на нелинейном элементе и соответственно увеличится доля входного напряжения, приходящаяся на сопротивление R . Последнее приведет к еще большему увеличению силы тока в цепи. В результате схема быстро придет к устойчивому равновесному состоянию, соответствующему точке P_1 . Аналогичными рассуждениями можно убедиться, что малое уменьшение тока i_3 приведет схему в другое равновесное состояние, определяемое точкой P_2 .

Переход схемы из одного равновесного состояния в другое осуществляется под воздействием изменяющегося внешнего сигнала. Пусть схема находится в равновесном состоянии, которому соответствует некоторая точка $P_1^{(h)}$, расположенная на участке ab характеристики рассматриваемого нелинейного элемента. При увеличении входного напряжения эта точка будет смещаться по направлению к точке b характеристики. Когда входное напряжение достигает значения E_2 , схема перейдет в другое устойчивое состояние, точка равновесия переместится на участок cd графика. Дальнейшее увеличение входного напряжения не изменит состояния схемы. Для ее возврата в исходное равновесное состояние необходимо уменьшить входной сигнал до значения E_1 . Таким образом, электрическая цепь с нелинейным сопротивлением, вольтамперная характеристика которого содержит участок с отрицательным наклоном, обладает двумя состояниями устойчивого

равновесия. Выходной сигнал в схеме зависит от входного сигнала и состояния схемы.

Электрическую цепь, обладающую указанными свойствами, можно реализовать в виде нелинейного усилителя при надлежащем выборе параметров и режима работы и ввода в его схему положительной обратной связи по напряжению (рис. 3.40; пояснение схемы, изображенной на этом рисунке, будет дано ниже). Последнее эквивалентно включению в электрическую цепь элемента с отрицательным сопротивлением.

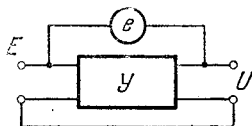


Рис. 3.40.

Пусть зависимость между входным и выходным напряжениями усилителя при разомкнутой цепи обратной связи, т. е. график $U' = f(E')$, имеет вид, представленный на рис. 3.41. Такую характеристику можно получить, например, с помощью схемы, построенной на двух триодах, работающих в режиме инверторов (рис. 3.42). Участок $a'b'$ графика соответствует большому отрицательному напряжению

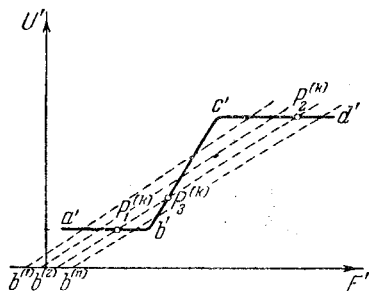


Рис. 3.41.

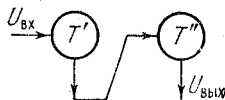


Рис. 3.42.

на входе триода T' , при котором лампа закрыта. При этом имеет место высокий уровень напряжения на ее аноде и, следовательно, на сетке триода T'' и соответственно низкий уровень выходного напряжения, снимаемого с анода второй лампы. Изменение входного напряжения в области, ограничиваемой участком $a'b'$ графика, не влияет на величину выходного напряжения, которая

остается практически постоянной. Точка b' этого участка соответствует потенциалу отпирания триода T' . Дальнейшее увеличение входного напряжения приводит к быстрому уменьшению анодного напряжения первой лампы и сеточного напряжения второго триода. Выходное напряжение U' при этом быстро возрастает (участок $b'c'$). В точке c' , соответствующей потенциалу запираания триода T'' , он достигает наибольшего значения и остается практически постоянным на всем участке $c'd'$. В этой области схема работает при отпертом триоде T' и запертом триоде T'' .

Вернемся теперь к анализу работы усилителя (J) с положительной обратной связью (рис. 3.40), условно показанной на рисунке в виде источника управляющего напряжения e . При замкнутой цепи обратной связи соотношения между напряжениями в усилителе имеют следующий вид:

$$U' = f(E'), \quad (3.19)$$

$$U' = E' - e. \quad (3.20)$$

Зависимость (3.20), как уже указывалось, имеет место для схемы с разомкнутой обратной связью и ей соответствует график $a'b'c'd'$ на рис. 3.41, а соотношение (3.19) определяет семейство прямых (по параметру e), расположенных под углом 45° к оси координат (при одинаковых масштабах по обеим координатным осям). Точки пересечения $P_1^{(k)}$, $P_2^{(k)}$ и $P_3^{(k)}$ некоторой прямой (из указанного семейства прямых) с графиком $a'b'c'd'$ определяют равновесные состояния схемы при фиксированном значении напряжения $e^{(k)}$, обусловленного обратной связью.

Выберем систему координат (U, e) и для каждого фиксированного значения $e^{(k)}$ отложим по вертикали значения U , соответствующие равновесным состояниям схемы (рис. 3.43). Геометрическое место точек равновесия образует кривую, составленную из трех участков (I, II, III). Анализ устойчивости равновесных состояний показывает, что участки I и II являются областями устойчивого равновесия, а точки, расположенные на участке III , соответствуют неустойчивым равновесным состояниям усилителя с положительной обратной связью. При изменении внешнего сигнала, вызывающего изменение сигнала в це-

при обратной связи, схема переходит скачкообразно из одного устойчивого равновесного состояния (определяемого, скажем, участком *I*) в другое устойчивое состояние (определяемое участком *II*) и обратно. Промежуточные равновесные состояния, возникающие при переходном процессе, например в моменты, когда в схеме, изображенной

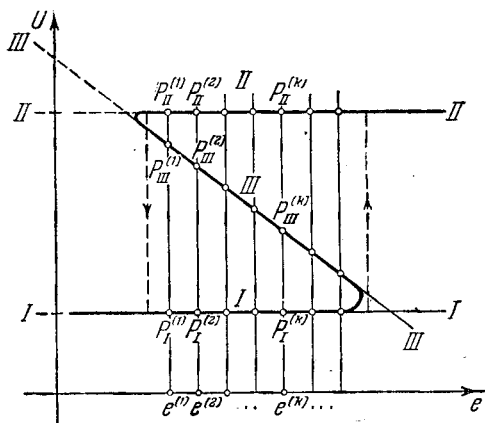


Рис. 3.43.

на рис. 3.42, обе лампы открыты или заперты, являются неустойчивыми.

Изложенные выше соображения лежат в основе построения триггерных схем. Получивший в настоящее время наиболее широкое применение в дискретных автоматах так называемый *статический триггер*, обычно реализуется в виде цепочки из двух инверторов, охваченной положительной обратной связью.

На рис. 3.44, а представлена в упрощенном виде типовая схема триггера, построенная на ламповых триодах с реостатными звеньями связи и раздельным управлением по каждой из сеток. На рис. 3.44, б дана его условная схема.

Если в рассматриваемой триггерной схеме отключить анодную нагрузку правой лампы от общей шины питания и подключить к отдельному источнику напряжения, то мы получим схему, для которой эквивалентной будет

электрическая цепь, изображенная на рис. 3.38. В ней сопротивление R соответствует анодной нагрузке R_a'' ,

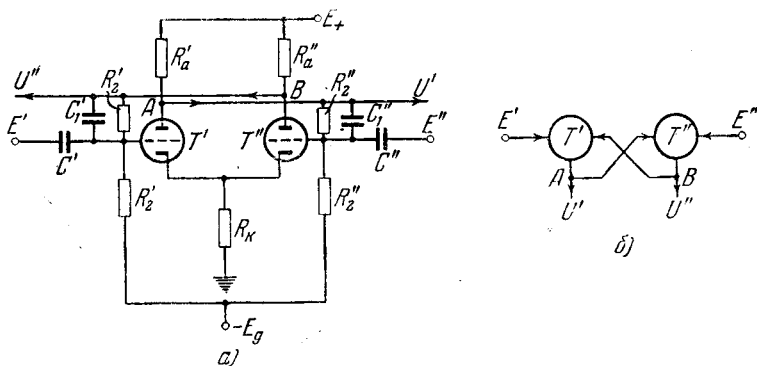


Рис. 3.44.

а нелинейное сопротивление эквивалентно всей триггерной схеме, ограниченной точкой B . Вольтамперная

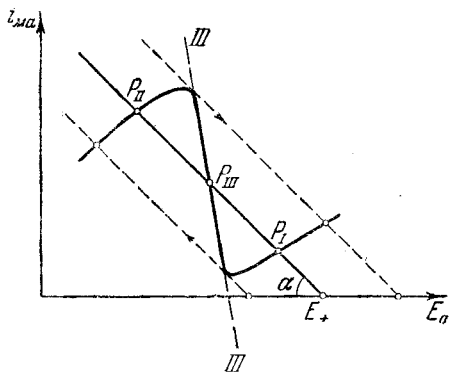


Рис. 3.45.

характеристика указанной части триггерной схемы имеет вид, представленный на рис. 3.45, и включает участок *III* с отрицательным наклоном, обусловленный наличием положительной обратной связи в цепи триггера. Нагрузочная прямая, проведенная через точку с абсциссой E_+

под углом $\alpha = \operatorname{arctg} \frac{1}{R_a''}$, пересекает вольтамперную характеристику в трех точках (P_I , P_{II} , P_{III}). Точка P_{II} соответствует состоянию устойчивого равновесия схемы при открытой правой лампе и запертой левой лампе, точка P_I — состоянию устойчивого равновесия при запертой правой и открытой левой лампе. Точка P_{III} является точкой неустойчивого равновесия, возникающего в схеме при переходном процессе.

При изменении анодного напряжения E_a или при подаче соответствующих управляющих сигналов на сетки или катоды ламп триггер переходит из одного устойчивого равновесного состояния в другое.

Оба устойчивых состояния триггера характеризуются обычно уровнями потенциалов в точках A и B схемы (рис. 3.44). Одному из них соответствует низкий уровень напряжения в точке A и высокий уровень напряжения в точке B (лампа T' открыта, лампа T'' закрыта). Другому соответствует обратное распределение уровней напряжения: высокий потенциал в точке A и низкий потенциал в точке B (лампа T' заперта, лампа T'' открыта). Переход триггера из одного устойчивого состояния в другое осуществляется подачей управляющих импульсов на входы схемы. В схеме, изображенной на рис. 3.44, ими являются сетки триодов. В некоторых схемах управление производится импульсами, поступающими на катоды или аноды ламп.

В качестве управляющего может служить как положительный, так и отрицательный импульсы. Однако вследствие большей чувствительности рассматриваемой схемы к импульсам отрицательной полярности управление обычно осуществляется отрицательными импульсами, имеющими амплитуду порядка 50 вольт и длительность порядка долей микросекунд. Интервал времени между двумя последовательными управляющими импульсами — такт работы триггера — должен превышать время срабатывания и восстановления схемы.

Управление работой триггера может осуществляться разными способами: последовательной подачей импульсов на каждый из входов схемы, изображенной на рис. 3.44, или путем подачи импульсов одновременно на оба

входа этой схемы. Последний случай равносителен объединению обоих входных каналов в один (рис. 3.46).

Рассмотрим несколько подробнее реакцию триггерной схемы на входные сигналы, реализуемые импульсами отрицательной полярности. Пусть триггер, изображенный на рис. 3.44, находится в состоянии, при котором левая лампа заперта, а правая открыта. Тогда подача отрицательного импульса на левый вход не изменит состояния схемы: точка P_I (рис. 3.45) сместится вправо. Потенциал точки

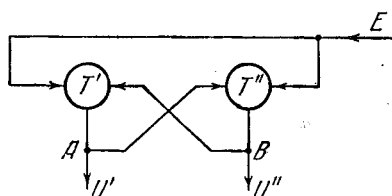


Рис. 3.46.

А сохранит высокий и практически неизменный уровень. Подача отрицательного импульса надлежащей амплитуды на сетку правой лампы запрет триод T'' , и потенциал точки В быстро возрастет. Лампа T' откроется, и уровень напряжения в точке А снизится. На

рис. 3.45 этому соответствует смещение точки P_I влево с последующим переходом точки равновесного состояния на верхнюю ветвь характеристики. По окончании переходного процесса схема придет во второе устойчивое равновесное состояние (точка P_{II}), при котором триод T' открыт, а лампа T'' заперта. Возврат схемы в исходное состояние осуществляется под воздействием отрицательного импульса, поданного на левый вход триггера.

В схеме, изображенной на рис. 3.46, отрицательный импульс поступает на оба входа триггера одновременно. Выбором параметров схемы, начальных смещений на сетках ламп и амплитуды импульса обеспечивается запирающее действие открытой лампы, вызывающее в свою очередь отпирание запертой лампы. Таким образом, подача каждого управляющего импульса вызывает изменение состояния схемы, и каждый четный импульс возвращает схему в исходное состояние. Одновходную триггерную схему называют иногда счетчиком четности (по модулю 2). Отметим, что емкости на входах схемы обеспечивают реакцию триггера только на изменение входного сигнала и исключают влияние его длительности.

Выбор параметров схемы триггера и управляющего сигнала, обеспечивающих его надежную работу в заданном режиме, осуществляется аналогично выбору пара-

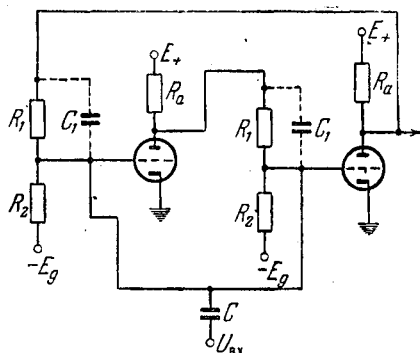


Рис. 3.47.

метров работающего на импульсах инвертора, у которого выходное напряжение равно входному (рис. 3.47).

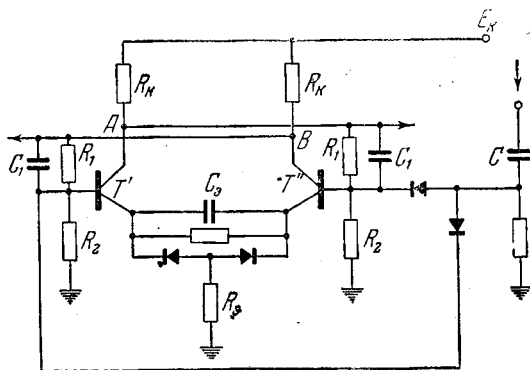


Рис. 3.48

Каждая лампа в схеме нагружена последовательно соединенными сопротивлениями R_1 и R_2 емкостью C_1 (входной емкостью лампы, паразитной емкостью монтажа и емкостью следующего каскада). Для надежной работы триггера

необходимо обеспечить полное запираание и полное отпираание ламп. При управлении отрицательным импульсом для надежного запираания ламп необходимо, чтобы амплитуда импульса была не меньше напряжения отсечки триода.

На рис. 3.48 приведена одна из распространенных триггерных схем, построенная на кристаллических триодах с автоматическим смещением. Последнее получается за счет падения напряжения на эмиттерном сопротивлении R_e , вызываемого эмиттерным током открытого триода. Сопротивление R_e в цепи эмиттера увеличивает входное сопротивление схемы и повышает устойчивость и надежность ее работы. Диоды и емкость включены в ту же цепь с целью повышения чувствительности и быстродействия триггерной схемы.

2) Логические операторы триггерных схем*)

Состояние триггера можно охарактеризовать, указав распределение потенциалов в точках A и B его схемы (рис. 3.49). Сопоставим потенциалу точки A двоичную переменную φ_A , равную единице или нулю при соответственно высоком (триод T' заперт) или низком (триод T'' открыт) уровнях напряжения в точке A . Отметим, что при

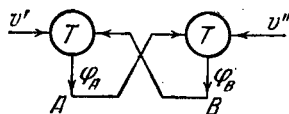


Рис. 3.49.

этом $\varphi_B = \overline{\varphi_A}$. Входные двоичные сигналы, реализуемые импульсами отрицательной полярности, поступающие на сетки триодов T' и T'' , обозначим соответственно через v' и v'' .

Из описания триггерной схемы, изложенного в предыдущем пункте, следует, что в состоянии схемы, соответствующем $\varphi_A = 1$, подача сигнала на левый вход триггера не изменяет его состояния. Сигнал, поступивший на правый триод, переводит триггер в состояние, соответствующее $\varphi_A = 0$. Аналогичным образом в состоянии 0 сигнал v'' не влияет на работу триггера, а сигнал v' пере-

*) См. М. Л. Ц е т л и н, О непримитивных схемах, Сб. «Проблемы кибернетики», вып. 1, Физматгиз, М., 1958, стр. 23—45.

водит схему в состояние 1. В рассматриваемой схеме двухвходного триггера одновременная подача сигналов на оба входа не допускается.

При переходе триггера из состояния $\varphi_A(t)=1$ в состояние $\varphi_A(t+1)=0$ на аноде левого триода возникает отрицательный перепад напряжения. Этот отрицательный импульс используется для управления последующим каскадом логической сети. Положительный импульс, возникающий на том же левом аноде при переходе триггера из состояния $\varphi_A(t)=0$ в состояние $\varphi_A(t+1)=1$, на работу последующего каскада не влияет.

В соответствии с этим составим функциональную таблицу, описывающую работу триггера. Строки таблицы, для которых одновременно $v'(t)=1$ и $v''(t)=1$ (соответствующие случаям одновременной подачи импульсов на оба входа триггера, которые, как указано выше, исключаются), дополним с таким расчетом, чтобы получить простое аналитическое представление таблицы. Таблица имеет следующий вид.

Т а б л и ц а III.2

$\varphi_A(t)$	$v'(t)$	$v''(t)$	$\varphi_A(t+1)$	$z(t)$
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

Из таблицы после упрощения получаем:

$$\varphi_A(t+1) = v'(t) \bar{\varphi}_A(t) \vee \bar{v}''(t) \varphi_A(t), \quad (3.21)$$

$$z(t) = v''(t) \varphi_A(t). \quad (3.22)$$

Выполнение соотношения (3.22) в триггере обеспечивается разными способами, в частности схемой, изображенной на рис. 3.50.

Соотношения, аналогичные (3.21) и (3.22), могут быть получены для $\varphi_B(t+1)$ и $z_B(t)$. В силу симметрии триг-

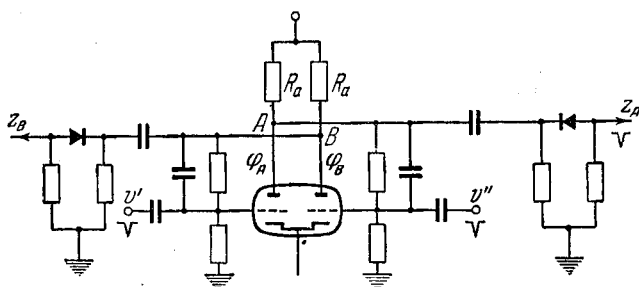


Рис. 3.50.

герной схемы они имеют следующий вид:

$$\varphi_B(t+1) = v'(t) \bar{\varphi}_B(t) \vee \bar{v}'(t) \varphi_B(t), \quad (3.23)$$

$$z_B(t) = v'(t) \varphi_B(t). \quad (3.24)$$

В одновходном триггере управляющий сигнал поступает одновременно на сетки обоих триодов. Приняв в вы-

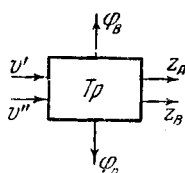


Рис. 3.51.

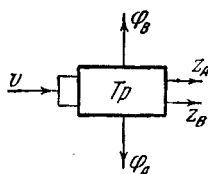


Рис. 3.52.

ражении (3.21) $v'(t) = v''(t) = v(t)$, мы получим уравнения для одновходного триггера в следующем виде:

$$\varphi_A(t+1) = v(t) \bar{\varphi}_A(t) \vee \bar{v}(t) \varphi_A(t) = v(t) \oplus \varphi_A(t), \quad (3.25)$$

$$z(t) = v(t) \varphi_A(t). \quad (3.26)$$

На рис. 3.51 и 3.52 даны условные обозначения триггеров, которыми мы в дальнейшем будем пользоваться.

Если триггер нужно включить в схему, работающую на положительных импульсах или положительных перепадах напряжения, то его входы снабжают инверторами T , обычно называемыми «тянущими» лампами (рис. 3.53). Здесь выходным сигналом положительной полярности является $z_A(t)$, возникающий при запираании триода T' . Это соответствует переходу триггера из состояния $\varphi_A(t)=0$ в состояние $\varphi_A(t+1)=1$, которое обусловлено входным сигналом $x''(t)=1$. Следовательно, в рассматриваемом случае:

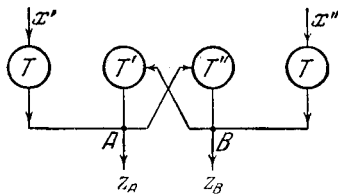


Рис. 3.53

$$\varphi_A(t+1) = \bar{\varphi}_A(t) x''(t) \vee \varphi_A(t) \bar{x}'(t), \quad (3.27)$$

$$z_A(t) = \bar{\varphi}_A(t) x''(t). \quad (3.28)$$

Входной сигнал $x'(t)$ переводит триггер в состояние $\varphi_A=0$. Если принять во внимание, что в рассматриваемой схеме

$$x''(t) = \bar{v}'(t), \quad x'(t) = \bar{v}''(t),$$

то соотношения (3.27) и (3.28) можно представить в следующем виде:

$$\varphi_A(t+1) = \bar{\varphi}_A(t) \bar{v}'(t) \vee \varphi_A(t) v''(t), \quad (3.27')$$

$$z_A(t) = \bar{\varphi}_A(t) \bar{v}'(t). \quad (3.28')$$

Полученные формулы отличаются от формул (3.21) и (3.22) тем, что здесь управляющие сигналы v' и v'' отрицательной полярности кодируются как нули.

В заключение рассмотрим схему логической сети так называемого *динамического триггера*. Это двухвходная триггерная схема отличается от статической тем, что при подаче на вход сигнала $v'(t)=1$ на выходе вырабатывается последовательность сигналов $z(t+1)=1$, $z(t+2)=1$, ..., прекращающаяся при поступлении входного сигнала

$v''(t+s)=0$. Поступление входного сигнала $v'=1$ в промежутке $t, t+s$ не меняет последовательности выходных единичных сигналов. Логическая сеть динамического триггера представлена на рис. 3.54.

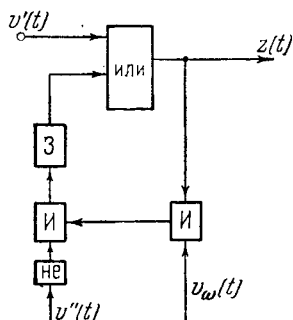


Рис. 3.54.

Как видно из схемы,

$$z(t) = v'(t) \vee \vee z(t-1) \bar{v}''(t-1) v_{\omega}(t-1), \quad (3.29)$$

где $v_{\omega}(t)$ — синхронизирующий импульс, поступающий на вход в каждом такте. Поскольку $v_{\omega}(t)=1$, выражение (3.29) можно представить в следующем виде:

$$z(t) = v'(t) \vee z(t-1) \bar{v}''(t-1).$$

Динамический триггер обладает более высоким скородействием, чем статический, его элементы работают в более легком режиме и ламповый вариант этого триггера имеет меньшие размеры, чем статическая триггерная схема.

§ 4. Ферромагнитные элементы *)

1) Физические основы **)

Ферромагнитные элементы пока уступают ламповым и полупроводниковым в отношении быстродействия, но более надежны в работе, имеют небольшие размеры и сравнительно дешевы.

Наиболее распространенными являются элементы, выполненные в виде тороидальных сердечников, несущих на себе обмотки: входные и одну выходную (рис. 3.55). В качестве материала для сердечников применяются обычно ферриты, получаемые спеканием порошкообразных окислов ферромагнитных металлов с добавлением присадок и наполнителей.

*) При изложении этого параграфа мы пользовались в основном статьей М. Л. Цетлина и Л. М. Шехтмана «Двухтактные ферротранзисторные схемы и алгебраический метод их синтеза» (сб. «Проблемы кибернетики», вып. 2, 1959, стр. 139—179).

**) См. З. И. Заволокина, Магнитные элементы в цифровых вычислительных устройствах, Госэнергоиздат, М.—Л., 1958.

Магнитные свойства сердечника характеризуются зависимостью магнитной индукции B от напряженности магнитного поля H . Эта зависимость, которую можно записать как $B = B(H)$, вообще говоря, неоднозначна, поскольку величина индукции зависит не только от заданного значения напряженности, но и от остаточной индукции сердечника до перемагничивания. Однако при достаточно больших H влияние остаточной индукции практически исчезает, и зависимость $B = B(H)$ приобретает однозначный характер, представляемый предельной петлей гистерезиса (рис. 3.56).

При отсутствии магнитного поля, т. е. в интервалах между подачей внешних сигналов, магнитная индукция сердечника может принимать либо значение $+B_r$, либо значение $-B_r$. Сопоставим этим значениям двоичную переменную и будем условно полагать, что значению магнитной индукции $+B_r$ соответствует возбужденное состояние сердечника. Ему припишем значение 1 двоичной переменной. Значению индукции $-B_r$ тогда будет соответствовать невозбужденное состояние сердечника, сопоставленное с нулевым значением двоичной переменной.

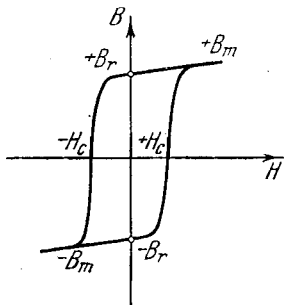


Рис. 3.56.

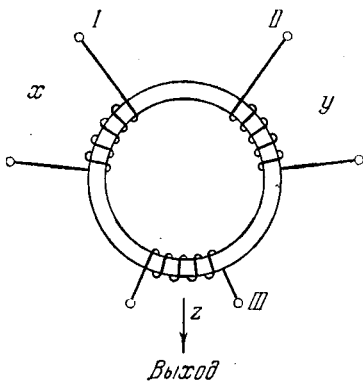


Рис. 3.55.

В схемах, построенных на ферромагнитных элементах, используются различные режимы работы сердечников. Рассмотрим вначале так называемые *трансформаторные схемы*, наиболее распространенные в современных автоматических системах дискретного действия.

При работе сердечника в трансформаторном режиме его входные обмотки, как правило, питаются импульсами тока постоянной полярности. То или иное направление магнитного поля обеспечивается различным направлением витков этих обмоток.

Удобно разбить все входные обмотки, принадлежащие одному сердечнику и предназначенные для изменения его магнитного состояния, на две группы: I) обмотки возбуждения, посредством которых сердечник переводится в состояние 1, и II) обмотки размагничивания, воздействием на которые сердечник переводится в состояние 0.

Предположим, что сердечник находится в состоянии 1, которое соответствует величине магнитной индукции $+B_r$. При подаче импульса на обмотку возбуждения индукция достигнет значения $+B_m$, а после снятия импульса вновь вернется к значению $+B_r$. При этом возникнет изменение величины индукции, равное

$$\Delta_{\Pi} B = B_m - B_r. \quad (3.30)$$

Если же импульс будет приложен к обмотке размагничивания, то индукция примет значение $-B_m$, а после снятия импульса станет равной $-B_r$. Следовательно, полное изменение индукции под воздействием размагничивающего импульса

$$\Delta_p B = B_m + B_r. \quad (3.31)$$

Аналогичное изменение величины индукции возникнет в случае, когда сердечник находится в состоянии 0 и на него будут поданы, соответственно, импульсы по размагничивающей обмотке и по обмотке возбуждения.

При изменении индукции сердечника между концами его обмоток возникает электродвижущая сила, определяемая следующим соотношением:

$$U = -WS \frac{dB}{dt} 10^{-8} \text{ (вольт)}, \quad (3.32)$$

где W — число витков обмотки, S — площадь сечения провода.

Приближенное выражение для этого соотношения можно представить следующим образом:

$$U = -WS \frac{\Delta B}{\tau} 10^{-8}, \quad (3.33)$$

где τ — время перемагничивания, зависящее от параметров сердечника и приложенного магнитного поля. Из этого соотношения следует, что при изменении состояния сердечника электродвижущая сила, наводимая в обмотках, примет следующую величину:

$$U_p = -WS \frac{\Delta p B}{\tau} 10^{-8} = -WS \frac{B_m + B_r}{\tau} 10^{-8}, \quad (3.34)$$

а паразитная электродвижущая сила, возникающая при изменении индукции в пределах одноименного участка (B_r, B_m) , выразится следующим образом:

$$U_{\Pi} = -WS \frac{B_m - B_r}{\tau} 10^{-8}. \quad (3.35)$$

Полярность наводимой электродвижущей силы зависит от направления приложенного поля.

При использовании сердечников в качестве элементов логических схем необходимо по возможности уменьшать отношение $\frac{U_{\Pi}}{U_p}$. Оно зависит от параметра $\alpha = \frac{B_r}{B_m}$, называемого *коэффициентом прямоугольности* петли гистерезиса. Для ферритов его значение лежит в пределах 0,9—0,98. При таких значениях параметра α могут быть построены вполне надежные схемы, в которых паразитные электродвижущие силы не влияют на функционирование схем.

В рассматриваемых нами трансформаторных схемах импульсы могут подаваться одновременно на входные обмотки обеих групп — возбуждения и размагничивания. При этом действующее значение напряженности магнитного поля определяется алгебраической суммой полей, наводимых каждой обмоткой. Это обуславливает возможность выполнения логических операций посредством ферромагнитных элементов. Они реализуются двумя последовательными этапами, условно называемыми *записью* и *считыванием*.

При записи импульсы, соответствующие входным двоичным переменным, поступают на входные обмотки сердечника и в соответствии с выполняемой логической операцией переводят его в состояние 0 или 1. Одна из размагничивающих обмоток используется только для считывания.

Импульс, поданный на эту обмотку, переводит сердечник в состояние 0 и возбуждает в выходной обмотке сигнал, соответствующий выходной двоичной переменной.

При построении схем на ферритовых сердечниках необходимо обеспечить надлежащую мощность выходного сигнала, снимаемого с каж-

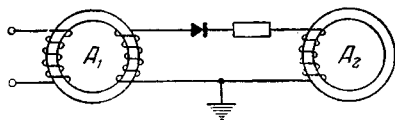


Рис. 3.57.

дого сердечника. Она должна быть достаточной для управления последующими элементами, по крайней мере двумя. Вместе с тем необходимо исключить

влияния взаимных полей в обмотках сердечников и обеспечить предписанное направление и синхронность передачи информации: только от управляющего сердечника к управляемому и при этом только во время считывания. Это означает, что подача импульсов записи на какой-либо из сердечников не должна изменять состояния управляемого им элемента, а подача импульса считывания не должна влиять на состояние управляющего элемента.

Выполнение указанных условий достигается надлежащей организацией связи между сердечниками, осуществляемой различными способами. В ферродиодных схемах в цепи связи включают кристаллические диоды (рис. 3.57). При переходе сердечника A_1 из состояния 1 в состояние 0 на выходной обмотке возникает импульс, обуславливающий появление тока в цепи связи; ток перемагничивает сердечник следующего каскада. Переход сердечника из состояния 0 в состояние 1 сопровождается возникновением в выходной обмотке импульса противоположной полярности (по отношению к первому случаю). Однако при наличии диода этот импульс не вызывает тока в цепи связи.

В ферротранзисторных схемах в цепи связи включают кристаллические триоды (рис. 3.58). При переходе сердечника A_1 из состояния 0 в состояние 1 возникающая на выходной обмотке (III) электродвижущая сила имеет положительную полярность. При этом на базу триода T подается положительный импульс, и ток

в цепи коллектора (K) не возникает. Когда при считывании сердечник A_1 переходит из состояния 1 в состояние 0, на базу (B) триода поступает импульс отрицательной полярности. Он вызывает в цепи коллектора ток, достаточный для возбуждения сердечника A_2 . Вместе с тем изменение состояния сердечника A_2 и возникающее в связи с этим изменение потенциала коллектора не приводит к появлению

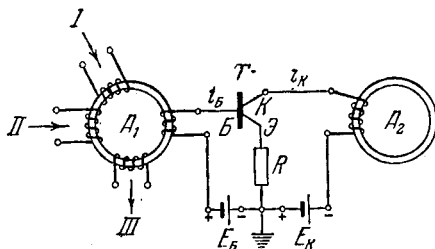


Рис. 3.58.

сколько-нибудь значительного тока в цепи коллектора или базы триода. Тем самым обеспечивается неизменность состояния сердечника A_1 и исключается передача информации в обратном направлении.

Надежность работы ферромагнитных схем существенно зависит от длительности управляющих импульсов и их синхронизации во времени. В реальной схеме вследствие изменений в параметрах сердечников и цепей связи возникают изменения в длительности выходных импульсов. При многокаскадном построении схемы это может приводить к такому уразбросу в длительности входных импульсов сердечников последующих каскадов, который повлечет за собой нарушение синхронности в их подаче, а это в свою очередь может привести к нарушениям в работе схемы. Поэтому приходится принимать специальные меры, для того чтобы возникающие отклонения параметров не выходили за допустимые пределы. Это достигается разными способами: надлежащим отбором ферритов, вводом в схему так называемых формирователей импульсов по длительности*), стробированием

*) Звеньев, обеспечивающих заданную длительность импульсов.

выходных импульсов*) и т. д. Подробное рассмотрение этого вопроса не входит в нашу задачу, и мы в дальнейшем изложении будем рассматривать функционирование схем без учета указанных отклонений.

При практической реализации схем на ферритах приходится принимать специальные меры для обеспечения правильной работы сердечников, несущих оба типа обмоток — возбуждающие и размагничивающие. Это обуславливается тем, что если импульс подан хотя бы на одну из размагничивающих обмоток сердечника, то сердечник должен остаться в состоянии 0 независимо от числа одновременно работающих обмоток возбуждения. Выполнение указанного требования достигается разными способами. Иногда, вместо одновременной подачи импульсов на обмотки возбуждения и запрета, запись разделяется на два этапа. В схемах с одновременной записью по обоим типам обмоток создают условия, при которых поле, вызываемое обмоткой запрета, превосходит суммарное поле всех обмоток возбуждения. Особенно важно, чтобы при наличии запрещающего импульса суммарное поле всех обмоток не было положительным в течение всего периода записи. Для этого запрещающему импульсу придают большую длительность, чем возбуждающему.

В связи с указанными обстоятельствами при построении схем на ферритах приходится пользоваться двумя типами элементов (G и H), отличающимися друг от друга длительностью выходного сигнала. В элементе типа G выходной сигнал служит для управления обмоткой возбуждения; в элементе типа H выходной сигнал, имеющий большую длительность, чем в элементе типа G , служит для управления обмоткой запрета. В функциональном отношении элементы G и H идентичны.

По способу считывания, обуславливающему схему коммутации считывающих обмоток, различают одно-, двух- и трехтактные трансформаторные схемы на ферритах.

*) При стробировании из выходных импульсов с помощью синхронизирующих импульсов в строго фиксированные моменты времени «вырезаются» узкие полоски. Таким способом достигается точная синхронизация импульсов, поступающих на сердечники последующих каскадов.

В *однотактных схемах* считывающие обмотки всех сердечников соединены последовательно (рис. 3.59), и, таким образом, считывающий импульс на все сердечники подается одновременно (если не принимать во внимание задержки при прохождении импульса через обмотки). При

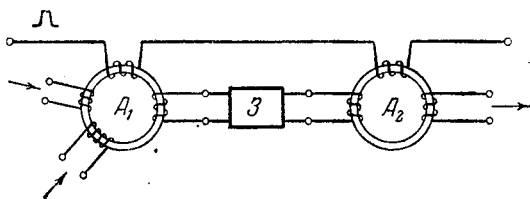


Рис. 3.59.

таким способе считывания в цепь междукаскадной связи сердечников приходится вводить элементы задержки (З)

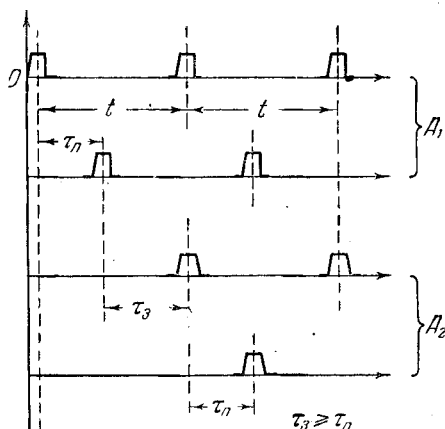


Рис. 3.60.

с тем, чтобы сдвинуть по времени поступление на сердечник входного и считывающего импульсов. Время задержки τ_z должно быть выбрано так, чтобы входной сигнал поступил на сердечник A_2 после его перемагничивания в состояние 0 под воздействием считывающего импульса. На рис. 3.60 показано распределение во времени импульсов

записи и считывания, поступающих на сердечники A_1 и A_2 . Импульс, считывающий информацию с сердечника A_1 , сдвинут относительно входных импульсов на время τ_n , необходимое для надежного перемагничивания сердечника под их воздействием. Импульс, считанный с сердечника A_1 , поступает, спустя интервал времени τ_3 , на сердечник A_2 и считывается в момент подачи следующего импульса считывания. Обычно принимают $\tau_3 \gg \tau_n$, и тогда импульсы записи поступают на сердечник последующего каскада со сдвигом относительно предыдущего каскада на один такт подачи внешних сигналов. Число считывающих импульсов, необходимое для передачи информации от внешних входных узлов сети к внешним выходным узлам, определяется общим числом каскадов в сети.

Более надежными и удобными для практической реализации являются так называемые *двухтактные схемы*. В этих схемах считывающие обмотки делят на две группы: обмотки четных и нечетных каскадов (рис. 3.61). Каждая из групп обмоток подключается к источникам считывающих импульсов, работающим со сдвигом во времени друг относительно друга. Вначале поступает считывающий импульс *СИ* в обмотки нечетных каскадов. Записанная в этих каскадах информация поступает на элементы четных каскадов. Затем со сдвигом во времени, обеспечивающим надежные условия работы схемы, поступает считывающий импульс *СИИ* на обмотки четных каскадов; информация переписывается на сердечники последующих нечетных разрядов и т. д.

Высокая помехоустойчивость с полным подавлением обратной передачи информации обеспечивается *трехтактными трансформаторными схемами* (рис. 3.62). Здесь считывающие обмотки всех каскадов объединены в три группы, каждая из которых подключается к отдельному источнику считывающих импульсов. Основные импульсы считывания поступают на группы обмоток с временным сдвигом τ , обеспечивающим надежную перезапись информации с предыдущего сердечника на последующий. Для подавления обратной передачи информации одновременно с подачей считывающего импульса на последующую группу обмоток на предыдущую группу обмоток подается добавочный импульс.

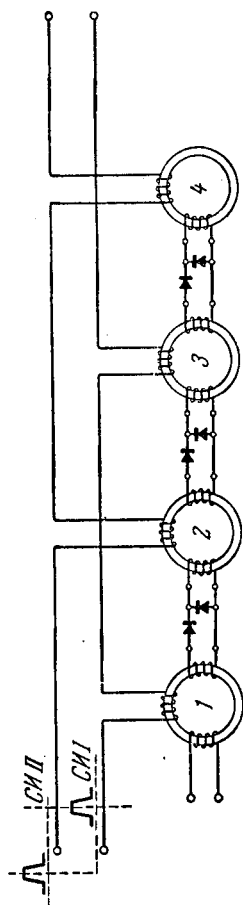


Рис. 3.61.

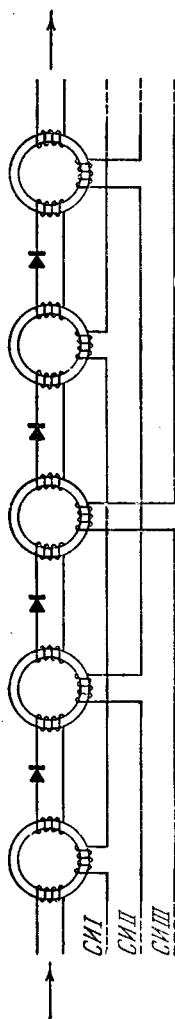


Рис. 3.62.

Трехтактные схемы на ферритах требуют значительного увеличения числа сердечников. Поэтому, несмотря на высокую надежность работы, эти схемы не нашли широкого применения. Наиболее распространенными являются двухтактные трансформаторные схемы.

В заключение приведем краткое описание *ферродиодных дроссельных элементов* *), привлекающих в последнее время внимание простотой их реализации и высокой надежностью работы.

Принципиальная схема такого элемента дана на рис. 3.63. Он состоит из сердечника C , несущего одну или

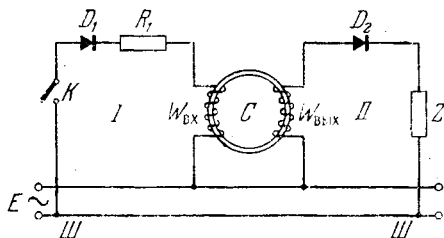


Рис. 3.63.

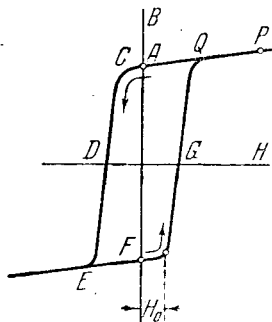


Рис. 3.64.

более входных обмоток $W_{вх}$ и выходную обмотку $W_{вых}$. В цепях этих обмоток включены диоды (D_1 и D_2). Элемент питается синусоидальным напряжением частоты f с амплитудным значением E , поступающим по шине III. Благодаря диодам во входной (I) и выходной (II) цепи токи в этих цепях протекают только в различные полупериоды питающего напряжения. Назовем полупериод, в течение которого ток может протекать во входной обмотке, *установочным*, а полупериод протекания тока в выходной обмотке — *рабочим*.

Пусть в начале установочного полупериода сердечник находился в состоянии, соответствующем точке A петли гистерезиса (рис. 3.64), и в течение этого полупериода ток

*) См. Л. М. Шехтман, Алгебраический метод синтеза ферродиодной дроссельной схемы. Сб. «Вопросы радиоэлектроники», серия 12, вып. 18, 1959.

через входную обмотку не протекал. (условно будем считать, что ключ K разомкнут, рис. 3.63). Тогда в течение рабочего полупериода происходит перемагничивание сердечника по участку $AQPA$ петли, и падение напряжения на выходной обмотке ($W_{\text{вых}}$) сердечника незначительно. При этом на нагрузке Z выделяется импульс напряжения, приблизительно равный питающему напряжению рабочего полупериода, и ток в выходной цепи выразится величиной

$$i_{\text{вых}} = \frac{E}{Z}$$

(импеданс выходной обмотки при отсутствии перемагничивания невелик).

Если в течение установочного полупериода во входной цепи протекал ток (ключ K замкнут), то сердечник перемагнитится (по участку $ACDEF$ петли гистерезиса) и перейдет в состояние, соответствующее точке F петли. Тогда в течение последующего рабочего полупериода ток, протекающий через выходную цепь, вернет сердечник в исходное состояние (по участку $EGQA$ петли). При этом в выходной обмотке сердечника возникнет импульс, амплитуда которого приблизительно равна амплитуде питающего напряжения, а импульс напряжения, выделяющийся на нагрузке Z , будет незначительным.

Итак, если в течение установочного полупериода, к которому мы отнесем момент времени t во входной цепи, ток не протекает, то в последующем за ним рабочем полупериоде ($t+1$) в выходной цепи протекает большой ток. Если же в течение установочного полупериода (t) во входной цепи протекает ток, достаточный для перемагничивания сердечника, то в момент ($t+1$) ток в выходной цепи будет незначительным.

Соединим теперь два элемента так, как показано на рис. 3.65 (число витков выходной обмотки каждого сердечника должно превышать число витков его входной обмотки). В этой схеме нагрузкой для первого элемента служит входная цепь второго элемента. Пусть в течение установочного полупериода сердечник C_1 перемагнитился из состояния A в состояние F (рис. 3.64). Как видно из схемы, в том же полупериоде ток будет протекать также в выходной цепи сердечника C_2 , и к концу этого полупе-

риода он придет в состояние, соответствующее точке A петли, независимо от предыдущего магнитного состояния. В течение следующего полупериода сердечник C_1 также вернется в исходное состояние (точку A петли); состояние сердечника C_2 при этом не изменится (так как по причинам, указанным выше, ток в выходной цепи сердечника C_1 будет мал). Если в течение установочного полупериода ток во входной цепи первого сердечника не

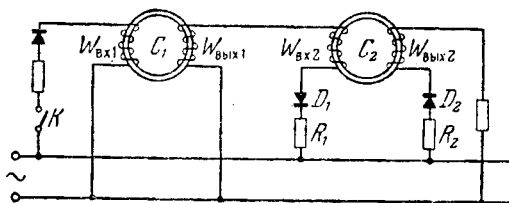


Рис. 3.65.

протекает, то к началу рабочего полупериода сердечники C_1 и C_2 будут находиться в состоянии, соответствующем точке A петли гистерезиса. В течение последующего рабочего полупериода сердечник C_2 перемагнитится в состояние, определяемое точкой F петли. Таким образом, в многокаскадной схеме, составленной из ферродиадных дроссельных элементов, при перемагничивании в момент времени t сердечника предыдущего каскада, в последующий момент времени $(t+1)$ сердечник последующего каскада не перемагничивается, и наоборот.

Снабдим теперь сердечник двумя входными обмотками, направленными встречно друг другу. Входную обмотку, с помощью которой обеспечивается описанный режим работы сердечника, будем называть перемагничивающей (W_n), а обмотку, создающую противоположное по направлению магнитное поле, назовем запрещающей (W_a). Протекание тока через запрещающую обмотку в течение установочного полупериода приводит сердечник в состояние, соответствующее точке A петли гистерезиса (независимо от предшествующего состояния). При этом в рабочем полупериоде в выходной цепи сердечника будет протекать ток i_n . Обмотки выберем таким образом, чтобы при одном и том

же токе i_b для каждой обмотки было справедливо

$$H_{пв} - H_{зв} < H_0,$$

где $H_{пв}$ и $H_{зв}$ — поля, создаваемые соответственно перемagnичивающей и запрещающей обмотками, H_0 — поле старта (необходимое для перемagnичивания сердечника). Тогда при протекании тока i_b одновременно по обеим обмоткам сердечник независимо от исходного состояния переводится в состояние, соответствующее точке A петли гистерезиса. Выполнение указанного условия необходимо также для обеспечения надежной работы элемента.

Описываемый ферромагнитный элемент, называемый в дальнейшем элементом типа R , может управлять несколькими (от 4 до 10) последующими сердечниками, входные обмотки которых могут соединяться последовательно или параллельно; однако по ряду причин предпочтительным является параллельное соединение управляемых обмоток.

2) Операторы ферромагнитных элементов

Предположим, что в схеме ферромагнитного трансформаторного элемента, представленной на рис. 3.55, обмотка I является возбуждающей, а обмотка II — размагничивающей. Условные обозначения обмотки того и другого типа показаны на рис. 3.66, a и b соответственно, а условная

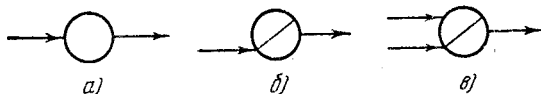


Рис. 3.66.

схема элемента с обмотками обоих типов и с выходной обмоткой дана на рис. 3.66, $в$. Найдем оператор, реализуемый этим элементом.

Предварительно напомним, что подача сигнала на обмотку возбуждения приводит элемент в состояние, сопоставленное со значением 1 двоичной переменной. Подача сигнала только на размагничивающую обмотку или одновременная подача сигналов на обе обмотки приводит

сердечник в состояние, которому сопоставлено нулевое значение двоичной переменной. При переходе сердечника из состояния 1 в состояние 0 в выходной обмотке возникает сигнал, сопоставляемый со значением 1 выходной переменной, а сигналу, возбуждаемому в той же обмотке при переводе феррита из состояния 0 в состояние 1, припишем нулевое значение выходной переменной.

Обозначим входные сигналы, поступающие на возбуждающую и размагничивающую обмотки, соответственно, через x и y , состояние элемента через q , а выходной сигнал обозначим через z . Тогда работа рассматриваемого элемента описывается таблицей III.3.

Таблица III.3

$q(t)$	$x(t)$	$y(t)$	$q(t+1)$	$z(t)$
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

Из таблицы III.3 мы получаем следующие канонические уравнения ограниченно-детерминированного оператора, реализуемого рассматриваемым элементом:

$$\left. \begin{aligned} z(t) &= q(t) y(t), \\ q(t+1) &= [q(t) \vee x(t)] \bar{y}(t). \end{aligned} \right\} \quad (3.36)$$

В общем случае, когда сердечник несет m возбуждающих и l размагничивающих обмоток, канонические уравнения этого оператора будут иметь следующий вид:

$$\left. \begin{aligned} z(t) &= q(t) [y_1(t) \vee y_2(t) \vee \dots \vee y_m(t)], \\ q(t+1) &= [q(t) \vee x_1(t) \vee x_2(t) \vee \dots \\ &\quad \dots \vee x_m(t)] \bar{y}_1(t) \bar{y}_2(t) \dots \bar{y}_l(t). \end{aligned} \right\} \quad (3.37)$$

Как было указано в предыдущем разделе этого параграфа, при реализации логических операций на трансформаторных ферромагнитных элементах запись информации на сердечнике и считывание с него осуществляются двумя последовательными этапами. При этом запись может производиться одновременно с подачей входных сигналов как на возбуждающие, так и на размагничивающие обмотки. Сигнал считывания подается на специальную размагничивающую обмотку, называемую считывающей.

Покажем теперь, что при таком режиме работы ферромагнитных элементов с помощью приведенного выше оператора можно получить полную систему элементарных функций, но в растянутом масштабе времени.

Будем полагать, что элемент, представленный на рис. 3.67, помимо указанных обмоток, снабжен еще дополнительной размагничивающей обмоткой (на схеме не изображенной), которая служит только для считывания информации сердечника. Пусть в момент времени t (который соответствует такту записи информации) сигнал $q(t)$ равен 0. Подадим на возбуждающие и размагничивающие обмотки соответственно сигналы 1 и y , а на обмотку считывания сигнал $v(t) = 0$. Тогда, в соответствии с приведенными выше каноническими уравнениями,

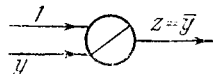


Рис. 3.67.

$$z(t) = 0,$$

$$q(t+1) = \bar{y}(t).$$

В момент $(t+1)$ соответствующий такту считывания сигнал $v=1$ поступает только на считывающую обмотку. При этом

$$z(t+1) = q(t+1)v(t+1) = \bar{y}(t),$$

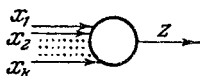
$$q(t+2) = 0.$$

Мы видим, что рассмотренный ферромагнитный элемент реализует оператор отрицания с задержкой на один такт. В зависимости от дальнейшего использования выходного сигнала, снимаемого с элемента, он может быть

выполнен в варианте G или H . Их операторы мы будем обозначать через $G(1, y)$ и $H(1, y)$; таким образом,

$$G(1, y) = H(1, y) = \bar{y}. \quad (3.38)$$

Ясно вместе с тем, что если при тех же начальных условиях и при той же организации считывания $y(t) = 0$, а $x(t) = 1$, то



$$z(t+1) = x(t), \quad \text{или} \quad G(x, 0) = x,$$

и элемент реализует задержку входного сигнала на один такт.

С помощью одного сердечника может быть реализован оператор дизъюнкции двух или более переменных (см. схему на рис. 3.68). Действительно, пусть $q(t) = 0$; тогда (при $v(t) = 0$)

$$q(t+1) = x_1(t) \vee x_2(t) \vee \dots \vee x_k(t)$$

и при подаче считывающего импульса

$$z(t+1) = q(t+1) = x_1(t) \vee x_2(t) \vee \dots \vee x_k(t).$$

Следовательно, оператор дизъюнкции, как и оператор отрицания, реализуется с задержкой на один такт.

В один такт и с помощью одного сердечника, несущего m возбуждающих и l размагничивающих обмоток, можно реализовать также функцию

$$z(t+1) = [x_1(t) \vee x_2(t) \vee \dots \vee x_m(t)] \bar{y}_1(t) \bar{y}_2(t) \dots \bar{y}_l(t).$$

В частном случае при $m = l = 1$ мы имеем:

$$z(t+1) = x(t) \bar{y}(t). \quad (3.39)$$

Оператор, реализующий функцию (3.39), мы будем в дальнейшем обозначать через $G(x, y)$ или $H(x, y)$.

Получение оператора конъюнкции двух или более переменных с помощью ферромагнитных трансформаторных элементов осуществляется на основе известного соотношения

$$z = x_1 x_2 \dots x_k = \overline{x_1 \vee x_2 \vee \dots \vee x_k}.$$

В самом общем случае его реализация требует трехкаскадной схемы (рис. 3.69). В первом каскаде образуются отрицания входных переменных, во втором каскаде образуется дизъюнкция выходных сигналов первого каскада

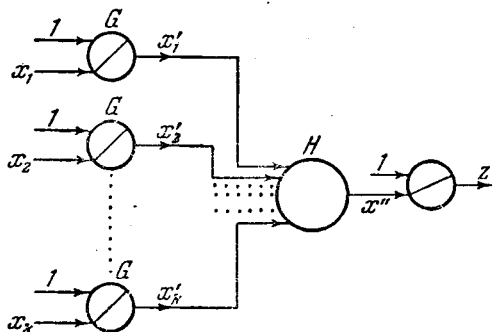


Рис. 3.69.

и, наконец, выход третьего каскада, образующий отрицание указанной дизъюнкции, дает требуемую конъюнкцию переменных x_1, x_2, \dots, x_k . Имея в виду, что

$$x'_j(t+1) = \bar{x}_j(t),$$

$$x''(t+2) = \bigvee_{j=1}^k x'_j(t+1) = \bigvee_{j=1}^k \bar{x}_j(t),$$

мы получаем:

$$z(t+3) = \overline{\bigvee_{j=1}^k \bar{x}_j(t)} = x_1(t) x_2(t) \dots x_k(t).$$

Следовательно, оператор конъюнкции в общем случае реализуется в масштабе $\sigma=3$. Очень часто, однако, в схемах, построенных на ферритах, дизъюнкцию выходных переменных, вырабатываемых элементами схемы, образуют путем объединения выходных полюсов сетей, доставляющих члены дизъюнкции (рис. 3.70). Это приводит к экономии сердечников и обмоток и вместе с тем уменьшает число тактов, необходимых для реализации полной системы элементарных функций. В частности, при этих

условиях упрощается образование конъюнкции, которая становится реализуемой двумя каскадами на основе

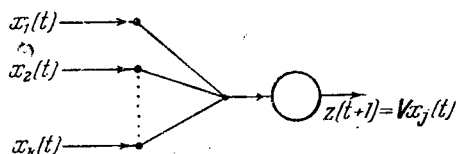


Рис. 3.70.

следующего соотношения (рис. 3.71):

$$\begin{aligned} z(t+2) &= x_1(t) x_2(t) \dots x_n(t) = \\ &= G\{1, H[1, x_1(t)] \vee H[1, x_2(t)] \vee \dots \\ &\quad \dots \vee H[1, x_n(t)]\}. \end{aligned} \quad (3.40)$$

На рис. 3.72 дана схема получения конъюнкции ($k=2$)

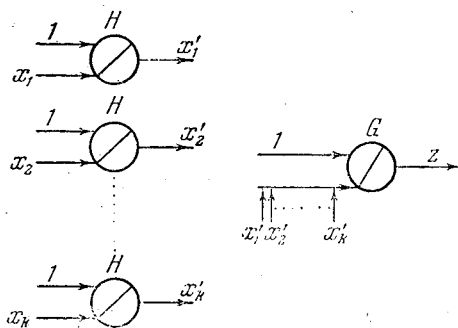


Рис. 3.71.

по формуле, несколько отличной от (3.40); эта формула имеет вид *)

$$z(t+2) = x_1(t) x_2(t) = G[G(x_1, 0), H(1, x_2)]. \quad (3.41)$$

*) Здесь, а в некоторых случаях также и в дальнейшем, в целях лучшей обзримости формул мы опускаем параметр времени (t) в обозначениях операторов.

Здесь сердечник 1 выполняет функцию задержки входного сигнала с тем, чтобы обеспечить одновременную запись и считывание входных и выходных сигналов в каждом каскаде сети.

Из рассмотрения приведенных операторов видно, что полная система элементарных функций может быть реализована ферромагнитными схемами в масштабе времени $\sigma = 2$. Отметим попутно, что, поскольку дизъюнкция может быть получена без увеличения числа каскадов в сети, любая совершенная дизъюнктивная нормальная форма может быть реализована двухкаскадной схемой, т. е. в масштабе $\sigma = 2$, с одноктактной или двухтактной системой считывания, в зависимости от принятого способа коммутации считывающих обмоток.

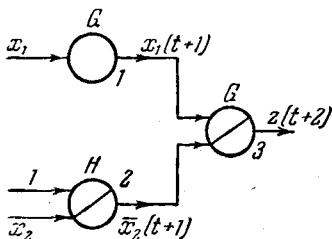


Рис. 3.72.

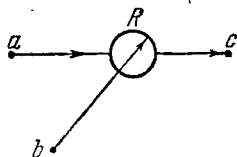


Рис. 3.73.

Перейдем теперь к описанию операторов ферромагнитных дроссельных элементов *). Снабдим сердечник двумя входными обмотками — перемещающей и запрещающей — и выходной обмоткой. Условная схема такого элемента представлена на рис. 3.73; здесь a является входным полюсом перемещающей обмотки, b — входной полюс запрещающей обмотки, c — выходной полюс. Состоянию q сердечника, соответствующему точке A петли гистерезиса (см. рис. 3.64), сопоставим значение 1 двоичной переменной, а состоянию, соответствующему точке F , сопоставим 0. Входные сигналы, поступающие на перемещающую и запрещающую обмотки, обозначим соответственно буквами x и y , выходной сигнал — буквой z (токам i_x в соответствующих цепях сопоставим значения 1). Тогда, пользуясь приведенным в предыдущем разделе данного пара-

*) См. статью Л. М. Шехтмана, упомянутую в примечании стр. 178.

графа описанием работы ферромагнитного дроссельного элемента, мы можем составить следующую таблицу.

4 Таблица III.4

$q(t)$	$x(t)$	$y(t)$	$q(t+1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Из таблицы III.4 мы находим, что

$$q(t+1) = q(t) \bar{x}(t) \vee y(t). \quad (3.42)$$

Выходная функция $z(t) = 0$, а

$$\left. \begin{aligned} z(t+1) &= q(t+1), \\ q(t+2) &= q(t). \end{aligned} \right\} \quad (3.43)$$

Покажем теперь, что с помощью элементов рассматриваемого типа можно реализовать полную систему функций в растянутом масштабе времени. Пусть в момент времени t (который соответствует некоторому установочному полупериоду) сигнал $q(t) = 1$, а $y(t) = 0$. Тогда, согласно (3.42) и (3.43),

$$z(t+1) = \bar{x}(t).$$

Таким образом, сердечник с одной входной перемагничивающей обмоткой реализует оператор отрицания с задержкой в один такт:

$$R(x, 0) = \bar{x}. \quad (3.44)$$

Отметим попутно, что при $q(t) = 1$ и $x(t) = 0$

$$z(t+1) = y(t) \quad \text{или} \quad R(1, y) = y.$$

т. е. что при указанных условиях сердечник работает как обычный элемент задержки входного сигнала на один такт.

Оператор дизъюнкции реализуется так же, как в схемах, построенных на ферромагнитных трансформаторных элементах, т. е. чаще всего путем объединения выходных

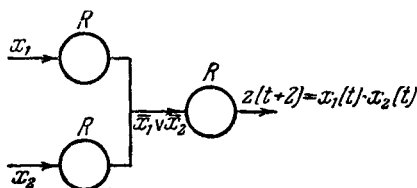


Рис. 3.74.

каналов элементов, доставляющих члены дизъюнкции. Оператор конъюнкции может быть получен на основе соотношения

$$x_1 x_2 \dots x_k = \overline{\overline{x_1} \vee \overline{x_2} \vee \dots \vee \overline{x_k}}$$

и так же, как и в ферротрансформаторных схемах, реализуется в масштабе $\sigma = 2$:

$$\begin{aligned} z(t+2) &= x_1(t) x_2(t) \dots x_k(t) = \\ &= R[R(x_1, 0) \vee R(x_2, 0) \vee \dots \vee R(x_k, 0), 0]. \end{aligned} \quad (3.45)$$

На рис. 3.74 приведена схема, построенная в соответствии с соотношением (3.45) для образования конъюнкции двух переменных.

Полезно указать, что с помощью одного элемента можно непосредственно реализовать функцию импликации с задержкой в один такт. Действительно, из (3.42) и (3.43) следует, что при $q(t) = 1$ мы будем иметь:

$$z(t+1) = \bar{x}(t) \vee y(t) = R(x, 0) \vee R(1, y).$$

Заметим, наконец, что посредством ферродиодных дроссельных элементов совершенная дизъюнктивная нормальная форма реализуется (как и на ферротрансформаторных элементах) за два такта.

ГЛАВА IV

АНАЛИЗ АВТОМАТОВ

§ 1. Общие замечания

В настоящей главе рассматривается задача анализа автоматов, кратко сформулированная в конце гл. II, а также некоторые примыкающие к ней вопросы.

Возможны различные постановки задачи анализа автомата в зависимости от того, какой мы располагаем информацией о его конструкции, и от того, что необходимо выяснить о его работе. Зачастую дело обстоит так, что задана схема автомата (обычно изображенная графически) и нужно установить, какой оператор реализуется в ней. Рассмотрим несколько подробнее эту ситуацию.

Задание схемы предполагает задание системы элементов $\{M_i\}$, из которых она построена, с указанием для каждого элемента его канонических уравнений, а также отношения подчиненности выходов входам. Не всякая схема над $\{M_i\}$ является логической сетью; так, например, в схеме, изображенной на рис. 2.16, б, два выходных канала соединены в полюсе z_1 , в то время как правила образования логической сети исключают возможность такого соединения. Задача анализа предполагает поэтому выяснение того, является ли предложенная схема логической сетью, или более общо, удовлетворяет ли она принятым требованиям «правильной организованности» (например, в том смысле, как это было оговорено для ламповых элементов).

Пусть уже установлено, что данная схема является логической сетью. Тогда, естественно, возникает задача определения того ограниченно-детерминированного оператора, который в ней реализуется. При этом речь может идти как об операторе сети *) в целом, так и об операторе,

*) Здесь и часто в дальнейшем изложении под сетью понимается логическая сеть.

отнесенном лишь к некоторым ее полюсам (см. гл. II, § 6). В этой главе ограниченно-детерминированный оператор считается заданным, если найдены его канонические уравнения (или соответствующие им таблицы).

Наконец, напомним, что если для ограниченно-детерминированного оператора получены каким-то образом канонические уравнения с K -значным внутренним алфавитом, то это еще не означает, что его вес также равен K ; на самом деле он может оказаться меньше K . В связи с этим возникает задача *минимизации* внутреннего алфавита с целью получения более компактного и обозримого описания рассматриваемого оператора.

Таким образом, мы можем отметить следующие три вопроса, подлежащих рассмотрению при указанной постановке задачи анализа автоматов:

1. Распознавание правильной организации схемы.
2. Составление канонических уравнений оператора, который реализуется схемой.
3. Минимизация алфавита состояний.

Все эти вопросы рассматриваются в § 5.

Прежде чем перейти к дальнейшему изложению, разъясним несколько подробнее причины, вызывающие обычно избыточность внутреннего алфавита. В соответствии с определениями гл. II объемом внутренней памяти для данного элемента M_i называется логарифм числа внутренних букв в системе канонических уравнений, которая отнесена этому элементу. Избыточность внутреннего алфавита может здесь возникнуть, например, за счет специфики кодирования. Чаще всего применяется двоичное кодирование, которому сопутствуют внутренние алфавиты с числом букв, являющимся степенью двойки, хотя вес оператора и не является числом такого вида. Объяснение этого явления заключается в том, что физически различные внутренние состояния элементарного автомата являются функционально неразличимыми. Легко понять, что если логическая сеть построена из таких элементов, то объем ее внутренней памяти (равный сумме объемов памяти составных элементов) будет также больше логарифма веса оператора, реализуемого ею. Однако избыточность внутренней памяти может иметь место в логической сети и в том случае, когда она не присуща ее элементам. Но даже

и в тех случаях, когда объем внутренней памяти данной логической сети в точности равен весу ее оператора, отнесенного ко всем выходным полюсам, он может быть существенно больше веса оператора, отнесенного лишь к некоторым выходным полюсам. Таким образом, если объем внутренней памяти известен, то мы располагаем лишь верхней оценкой для веса оператора, точное значение которого может быть найдено предлагаемым приемом минимизации.

П р и м е р. Пусть в состав какой-нибудь сети, кроме истинностных элементов (например, пентодов и триодов), входят лишь три задерживающих элемента (типа \mathfrak{M}_3). Тогда объем внутренней памяти сети равен 3; значит, каков бы ни был оператор, отнесенный к некоторым ее выходным полюсам, его вес ≤ 8 . В частности, оператор может оказаться даже истинностным. Вместе с тем полученная оценка для веса оператора позволяет сделать и некоторые дальнейшие заключения, например о том, как он будет перерабатывать входные периодические последовательности (см. § 4).

Этот тривиальный пример является поучительным в том смысле, что уже частичная информация о конструкции автомата позволяет делать определенные выводы о свойствах реализуемого им оператора. Поэтому оправдана и такая постановка задачи анализа, когда конструкция автомата неизвестна (или частично известна) и требуется установить лишь некоторые общие свойства оператора. В связи с этим мы рассмотрим ряд свойств ограниченно-детерминированных операторов (*достижимость, различимость, периодичность*) и установим соответствующие этим свойствам количественные характеристики, которые тесно связаны с весом оператора и, следовательно, с объемом памяти реализующей его сети.

Предположим, что имеются один или несколько экземпляров исследуемого автомата с неизвестной конструкцией, над которыми можно проводить эксперименты путем подачи входных последовательностей и наблюдения соответствующих выходных последовательностей. Пользуясь указанными понятиями (различимость и т. п.), можно посредством таких экспериментов установить важные свойства операторов, реализуемых автоматами, а в некоторых

случаях удастся даже точно определить эти операторы. Такой подход может оказаться полезным также для выяснения того, одинаково ли работают два автомата с различной конструкцией. Этот круг вопросов рассматривается нами в § 2—4 этой главы, но его завершением является содержание § 2 гл. V.

В заключение этого параграфа отметим следующие видоизменения в задании операторов посредством канонических таблиц и деревьев, которые окажутся удобными в этой главе:

1°. Канонические таблицы можно записывать с двойным входом. Например, вместо таблицы II.4 можно пользоваться таблицей IV.1 для выхода и таблицей IV.2 для состояния или же объединенной таблицей IV.3, в которой на пересечениях соответствующих строк помещены пары букв: $z(t)$ (в числителе) и $q(t+1)$ (в знаменателе).

Таблица IV.1

		q	
		0	1
x_1	x_2		
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Таблица IV.2

		q	
		0	1
x_1	x_2		
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Таблица IV.3

		q	
		0	1
x_1	x_2		
0	0	0/0	1/0
0	1	1/0	0/1
1	0	1/0	0/1
1	1	0/1	1/1

2°. «Свертывание» дерева оператора путем отождествления вершин, являющихся носителями одной и той же

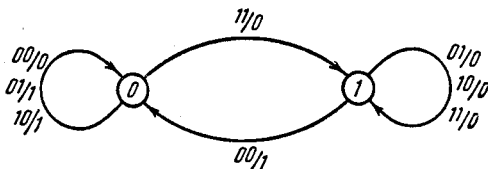


Рис. 4.1.

внутренней буквы, порождает так называемую *диаграмму переходов*. На рис. 4.1 изображена диаграмма переходов для оператора поразрядного сложения (ср. с деревом,

изображенным на рис. 2.10). Вершины диаграммы изображены в виде кружков, в которых помещены соответствующие внутренние буквы. Если существует входная буква, переводящая состояние q , в состояние q_j , то на диаграмме это зафиксировано стрелкой, ведущей из кружка q_i в кружок q_j , причем на этой стрелке указаны все такие входные буквы (в числителе), а также соответствующие выходные буквы (в знаменателе).

§ 2. Достижимость

В этом и последующем параграфах рассматриваются понятия *достижимости* и *различимости*, которые удобны для выяснения ряда специфических свойств ограниченно-детерминированных операторов.

Рассмотрим общий ограниченно-детерминированный оператор θ , задаваемый каноническими уравнениями

$$\begin{aligned} z(t) &= \Phi[x(t), q(t)], \\ q(t+1) &= \Psi[x(t), q(t)]. \end{aligned}$$

Будем считать пока, что все K букв внутреннего алфавита попарно различимы, а следовательно, путем надлежащей фиксации начального состояния $q(1)$ можно задать K частных ограниченно-детерминированных операторов. Допуская определенную вольность в формулировках, мы не будем четко отличать частные операторы от определяющих их начальных состояний, а в случае графического изображения в виде деревьев и диаграмм — от соответствующих вершин.

Пусть для входного слова длины S (т. е. слова, имеющего вид $x_1 x_2 \dots x_S$) и для начального состояния $p = q_1$ мы имеем:

$$\psi[x_1, q_1] = q_2, \quad \psi[x_2, q_2] = q_3, \quad \dots, \quad \psi[x_S, q_S] = q_{S+1} = r. \quad (4.1)$$

В таком случае мы будем говорить, что слово $x_1 x_2 \dots x_S$ переводит состояние (вершину, оператор) p в состояние (вершину, оператор) r или что r достижимо из p посредством слова $x_1 x_2 \dots x_S$. Это означает, что из вершины диаграммы переходов можно попасть в вершину r по ориентированной ломаной из S стрелок.

В терминах достижимости возникает такая же классификация ограниченно-детерминированных операторов и их состояний, как и в теории цепей Маркова. Применительно к ограниченно-детерминированным операторам эта классификация подробно рассмотрена в работе Ю. Я. Базилевского*). Мы ограничимся здесь рассмотрением лишь некоторых понятий, связанных с указанной классификацией.

О п р е д е л е н и е. Состояние q называется *возвратным* при выполнении следующего условия: если какое-

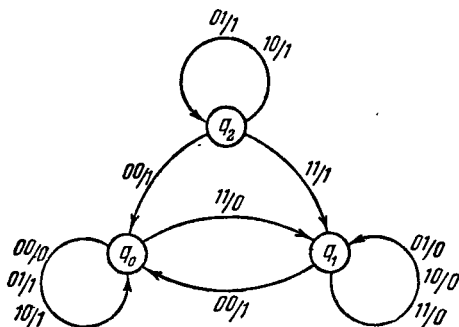


Рис. 4.2.

нибудь состояние p достижимо из q , то и q достижимо из p ; в противном случае оно называется *невозвратным*.

П р и м е р. Рассмотрим оператор, заданный диаграммой, изображенной на рис. 4.2. Из рисунка видно, что состояние q_2 невозвратно; подача однобуквенного слова 00 переводит его в состояние q_0 , из которого q_2 уже недостижимо; можно сказать, что состояние q_2 *забывается* и не может быть восстановлено в процессе работы оператора **).

Пусть S — какое-либо множество состояний оператора θ .

О п р е д е л е н и е. Множество состояний называется *замкнутым*, если невозможен переход из состояния,

*) Ю. Я. Базилевский, Вопросы теории временных логических функций, Сб. «Вопросы теории математических машин», Физматгиз, М., 1958.

**) Забегая вперед, отметим, что в тех случаях, когда такой оператор осуществляется в реальном физическом устройстве, возвращение к невозвратному состоянию иногда достигается путем надлежащего вмешательства извне.

входящего в S , к состоянию, не входящему в S . Если не существует никаких замкнутых множеств, кроме множества всех состояний, то оператор θ называется *неприводимым* (в противном случае он *приводим*).

Замкнутое подмножество состояний задает оператор, который является в известном смысле частью исходного оператора θ . Таковым является, например, множество, включающее какое-нибудь данное состояние q_0 и все те состояния, которые достижимы из q_0 ; иными словами, это есть частный оператор вместе со всеми своими остаточными операторами.

Оператор θ из предыдущего примера (рис. 4.2) приводим, ибо, как легко видеть, множество, состоящее из q_0 и q_1 , замкнуто; оно определяет оператор последовательного поразрядного суммирования (сравнить с диаграммой рис. 4.1).

Неприводимый оператор характеризуется тем, что всякое его состояние достижимо из любого другого его состояния. Таковым является, например, оператор диаграммы, изображенной на рис. 4.1.

Все состояния неприводимого оператора возвратны. Однако обратное положение не всегда имеет место. Оператор, все состояния которого возвратны, может оказаться приводимым. В этом случае множество всех состояний распадается на несколько замкнутых и попарно пересекающихся подмножеств состояний таких, что в каждом из них возможен переход от любого его состояния к любому другому его состоянию. Иными словами, такой оператор разлагается на несколько составляющих операторов (причем в данном случае все эти составляющие операторы неприводимы). В этом случае диаграмма распадается на несколько поддиаграмм, не связанных между собой стрелками.

О п р е д е л е н и е. Назовем *степенью достижимости оператора* θ (обозначение $\rho(\theta)$) наименьшее число ρ такое, что посредством входного слова длины $\leq \rho$ можно перевести любое состояние q во всякое другое такое состояние, в которое оно вообще переводимо.

Теперь нетрудно доказать следующую теорему.

Т е о р е м а 1. *Степень достижимости любого ограничено-детерминированного оператора строго меньше его*

веса, то есть $q(\theta)$ не больше веса оператора θ , уменьшенного на единицу.

Д о к а з а т е л ь с т в о. Действительно, если среди состояний q_1, q_2, \dots, q_{S+1} из уравнений (4.1) имеются повторяющиеся, например $q_i = q_j$, то, удалив из слова $x_1 x_2 \dots x_S$ буквы $x_{i+1}, x_{i+2}, \dots, x_j$, мы получили бы более короткое слово, переводящее p в r . Следовательно, можно считать, что все q_i попарно различны, но тогда $S+1$ не больше веса оператора θ , что и требовалось доказать.

В частности, перефразируя эту теорему применительно к тому случаю, когда оператор θ с весом K определяется своим деревом, можно сказать, что базисные вершины этого дерева удастся разместить ниже K -го яруса.

В дальнейшем, при отсутствии особых оговорок, под базисом дерева V , обозначаемым буквой \mathfrak{B} (возможно с индексами), мы будем понимать базис специального вида, определяемый следующим образом. Занумеруем все вершины дерева V ранга $< K$ в порядке возрастания их рангов:

$$0_1, 0_2, \dots, 0_n. \quad (4.2)$$

Для вершин одного и того же ранга порядок может быть любым; для определенности мы будем дальше пересчитывать вершины данного фиксированного ранга слева направо. Отнесем теперь к \mathfrak{B} всякую такую вершину, которая различима от любой другой вершины, предшествующей ей в принятой нумерации (4.2). Очевидно, что \mathfrak{B} является базисом.

П р и м е р. На рис. 4.3 изображено дерево константного оператора веса K , вырабатывающего последовательность (периодическую) $00\dots 0100\dots 01\dots$ ($K-1$ нулей). Его базис состоит из K вершин, расположенных на входном пути $00\dots 0$ (K нулей).

Отсюда видно, между прочим, что оценка, содержащаяся в теореме, не может быть улучшена, ибо в данном случае $q(\theta)$ в точности равно весу оператора θ , уменьшенному на единицу.

Однако нетрудно привести примеры операторов, у которых степень достижимости значительно меньше веса этого оператора. Это обстоятельство должно быть принято во внимание в тех случаях, когда приходится

устанавливать верхнюю оценку для степени достижимости (см. об этом гл. V, § 5).

С другой стороны, понятие степени достижимости может оказаться полезным при оценке веса оператора с н и з у. Пусть, например, в дереве оператора θ какая-нибудь выходная буква z встречается впервые в μ -м ярусе и не встречается в более низких ярусах; тогда очевидно, что

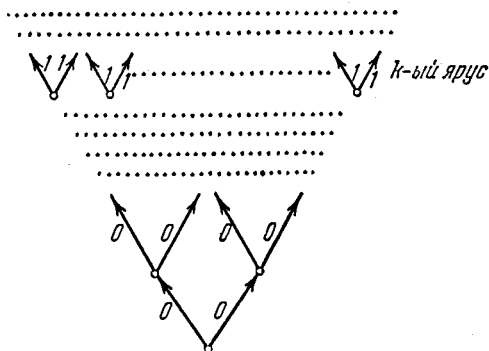


Рис. 4.3.

корень дерева (и его состояние) различим от той вершины γ ранга μ , из которой выходит ребро—носитель буквы z . Вместе с тем состояние, приписываемое вершине γ , недостижимо из начального состояния словом, которое короче, чем $\mu-1$; следовательно, вес такого оператора $\geq \mu$. Подобные рассуждения позволяют иногда установить, что некоторый детерминированный оператор не имеет конечного веса.

Рассмотрим, например, оператор последовательного умножения, который (наподобие оператора последовательного сложения) воспринимает на i -м такте i -е справа цифры исходных данных (множителя и множимого) и выдает i -ю цифру результата (произведения). Этот оператор является детерминированным; следующее рассуждение показывает, что он не имеет конечного веса.

Рассмотрим состояние p , в которое переходит начальное состояние после подачи (справа налево) пар цифр,

образующих следующие последовательности:

$$\dots 100\dots 0100\dots 0100\dots 0\dots$$

$$\dots \underbrace{100\dots 0100\dots 0100\dots 0}_{l \text{ нулей}} \dots$$

Каковы бы ни были входные цифры, подаваемые после пары единиц, следующие $l-1$ входных цифр будут сплошь нулями; первая ненулевая выходная цифра может появиться лишь l ярусами выше вершины p . Таким образом, в соответствии с предыдущим замечанием, степень достижимости — а значит, и вес — этого оператора $\geq l$. В силу произвольности l оператор не имеет конечного веса.

§ 3. Различимость

Задача, которая будет рассмотрена ниже, заключается в следующем. Пусть имеются два частных ограниченно-детерминированных оператора θ_1 и θ_2 (или, быть может, два состояния одного и того же общего оператора); предполагается, что наши сведения об этих операторах сводятся к тому, что для любого входного слова можно указать, в какие выходные слова оно перерабатывается. При этих условиях требуется выяснить: различимы эти операторы или нет?

Ситуацию, оправдывающую такую постановку вопроса, Мур*) изображает следующим образом. Предположим, что мы имеем дело с двумя автоматами, внутренние устройства которых нам неизвестны (т. е., как говорят в кибернетике, автоматы выступают в качестве «черных ящиков»); пусть эти автоматы реализуют ограниченно-детерминированные операторы θ_1 и θ_2 . Над нашими автоматами можно производить эксперименты, которые заключаются в том, что на их входы подаются какие-то слова и сравнивается та «продукция», которую выдают автоматы в ответ на то или иное слово. Путем надлежащей постановки эксперимента (желательно как можно более простого и экономного) нужно выяснить, одинаково ли

*) Э. Ф. Мур, Умозрительные эксперименты с последовательностными машинами, Сб. «Автоматы», ИЛ, М., 1956

функционируют эти автоматы (то есть реализуют ли они, по существу, один и тот же оператор или нет?).

В свете такой интерпретации поставленной выше задачи можно рассматривать эксперименты двух видов:

1) Простой эксперимент: каждый из исследуемых автоматов представлен одним экземпляром, а эксперимент заключается в подаче одного входного слова некоторой длины γ (длина эксперимента) на оба автомата и в сравнении результатов, выдаваемых автоматами.

2) Кратный эксперимент: экспериментатор располагает множеством экземпляров автоматов, причем каждый экземпляр принадлежит только к одному из двух изучаемых типов автоматов; экспериментатор вводит в каждый экземпляр автомата различные входные слова и сравнивает результаты, выдаваемые в ответ на это автоматами.

Введем теперь некоторые вспомогательные понятия.

Пусть частные ограниченно-детерминированные операторы θ_1 и θ_2 перерабатывают одни и те же входные слова $x = x(1) x(2) \dots x(S)$ в различные выходные слова $y = y(1) y(2) \dots y(S)$ и $y' = y'(1) y'(2) \dots y'(S)$. В таком случае мы будем говорить, что входное слово x различает операторы θ_1 и θ_2 или что операторы θ_1 и θ_2 различимы словом x . Если θ_1 и θ_2 различные ограниченно-детерминированные операторы, то всегда существуют слова конечной длины, различающие их *).

О п р е д е л е н и е 1. Операторы θ_1 и θ_2 k -различимы (обозначение: $\theta_1 \not\sim_k \theta_2$), если они различимы входным словом длины k ; в противном случае они k -неразличимы (обозначение: $\theta_1 \sim_k \theta_2$). Очевидно, если $\theta_1 \not\sim_k \theta_2$, то тем более $\theta_1 \not\sim_\gamma \theta_2$ при любом $\gamma > k$.

О п р е д е л е н и е 2. Операторы θ_1 и θ_2 строго k -различимы (обозначение: $\theta_1 \not\sim_k^s \theta_2$), если $\theta_1 \not\sim_k \theta_2$, но

*) Это утверждение справедливо вообще для детерминированных операторов.

не $\theta_1 \not\sim_{k-1} \theta_2$. О слове длины k , различающем операторы θ_1 и θ_2 такие, что $\theta_1 \not\sim \theta_2$, будем говорить, что оно *строго k -различает* эти операторы.

Термины и обозначения, только что введенные применительно к двум частным ограниченно-детерминированным операторам, естественным образом распространяются и на двойки состояний q_i и q_j общего ограниченно-детерминированного оператора θ ; это следует из того, что каждому состоянию однозначным образом сопоставлен свой остаточный оператор. Ниже мы будем применять эту терминологию — и соответствующие ей обозначения — также и к внутренним буквам алфавита состояний, встречающимся в канонических таблицах оператора θ .

Следующие далее утверждения А, Б, В легко вытекают из приведенных определений.

А. Для всякого ограниченно-детерминированного оператора существует натуральное число k такое, что всякие два его различных состояния уже k -различимы. Это утверждение вытекает из того, что ограниченно-детерминированный оператор имеет лишь конечное число пар различных состояний.

О п р е д е л е н и е 3. Наименьшее из натуральных чисел, обладающих свойством А, называется *степенью различимости* оператора θ (обозначение: $\text{ст}(\theta)$).

Б. Пусть входное слово $x_1 x_2 \dots x_k$ строго k -различает состояния q_i и q_j ; тогда слово $x_2 x_3 \dots x_k$ строго $(k-1)$ -различает состояния q'_i и q'_j , в которые переводятся состояния q_i и q_j соответственно, входной буквой x_1 .

Действительно, с одной стороны, слово $x_2 x_3 \dots x_k$ различает q'_i и q'_j , ибо в противном случае q_i и q_j были бы уже 1-различимы однобуквенным словом x_1 . С другой стороны, если бы q'_i и q'_j были γ -различимы каким-нибудь словом $\xi_{i_1} \dots \xi_{i_\gamma}$ длины $\gamma < k-1$, то слово $x_1 \xi_{i_1} \dots \xi_{i_\gamma}$ длины $\gamma+1 < k$ уже различало бы q_i и q_j .

Из утверждения Б вытекает, в частности, следующее следствие. Если ограниченно-детерминированный оператор θ имеет два строго k -различимых состояния, то для любого $\gamma < k$ он имеет и строго γ -различимые состояния. Другими словами, если среди состояний оператора θ существуют строго k -различимые, но не существует строго

$(k+1)$ -различимых, то $\text{ст}(\theta) = k$ (обратное утверждение непосредственно вытекает из определения).

В. Отношение k -неразличимости является симметричным (то есть из $q_i \sim_k q_j$ следует $q_j \sim_k q_i$) рефлексивным (то есть $q_j \sim_k q_j$) и транзитивным (то есть из $q_1 \sim_k q_2$ и $q_2 \sim_k q_3$ следует $q_1 \sim_k q_3$). Поэтому при любом p множество всех $\{q_n\}$ разбивается на систему классов $\{\mathfrak{B}_j^{(p)}\}$:

$$\mathfrak{B}_1^{(p)}, \mathfrak{B}_2^{(p)}, \dots, \mathfrak{B}_{j_p}^{(p)} \quad (4.3)$$

таких, что состояния из разных классов p -различимы, а состояния из одного и того же класса попарно p -неразличимы. Очевидно, разбиение (4.3) является полным*) в том и только в том случае, когда $p \geq \text{ст}(\theta)$.

Оценки длины экспериментов, которыми мы будем пользоваться в дальнейшем, основаны на следующей теореме.

Теорема 2. *Степень различимости любого ограничено-детерминированного оператора, не являющегося истинностным, строго меньше его веса. Для истинностного оператора очевидно, что $\text{ст}(\theta) = \text{вес}(\theta) = 1$ (где $\text{вес}(\theta)$ означает: «вес оператора θ »).*

Доказательство. Если в разбиении (4.3) $p < \text{ст}(\theta)$, то при переходе от p к $p+1$ элементы из различных классов $\{\mathfrak{B}_j^{(p)}\}$, будучи p -различимыми, — а подавно и $(p+1)$ -различимыми, — не могут оказаться в одном классе $\mathfrak{B}_s^{(p+1)}$. Поэтому для любого класса \mathfrak{B}_j^p имеется альтернатива: а) либо он является одним из классов $\mathfrak{B}_s^{(p+1)}$, б) либо он разбивается на два, или более, классов $\mathfrak{B}_s^{(p+1)}$. Последнее имеет место в том и только в том случае, когда в рассматриваемом классе \mathfrak{B}_j^p имеется хотя бы одна пара строго $(p+1)$ -различимых состояний.

Заметим теперь, что по крайней мере для одного из классов $\mathfrak{B}_j^{(p)}$ выполняется вторая альтернатива, коль скоро $p < \text{ст}(\theta)$, ибо в соответствии с пунктом В для каждого $\gamma \leq \text{ст}(\theta)$ (в том числе и для $\gamma = p+1$) должны существовать строго γ -различимые состояния.

*) То есть любые два различных состояния попадают в различные классы.

Итак, в каждом из разбиений

$$\{\mathfrak{B}_j^{(1)}\}, \{\mathfrak{B}_j^{(2)}\}, \dots, \{\mathfrak{B}_j^{\text{ст}(\theta)}\}$$

число классов по крайней мере на единицу больше, чем в предыдущем. Вместе с тем $\{\mathfrak{B}_j^{(1)}\}$ состоит по крайней мере из двух классов (ведь оператор θ не является истинностным). Отсюда следует вывод, что число p классов в полном разбиении, равное весу оператора, по крайней мере на единицу больше его степени различимости.

Теорема, таким образом, доказана.

Приведем еще без доказательства следующее утверждение:

Г. Два состояния q_i и q_j , принадлежащие какому-либо классу $\mathfrak{B}_j^{(p)}$, являются $(p+1)$ -различимыми (а значит, и строго $(p+1)$ -различимыми) в том и только в том случае, когда существует входная буква x , переводящая их соответственно в состояния q'_i и q'_j , принадлежащие различным классам $\{\mathfrak{B}_j^{(p)}\}$.

Простым следствием теоремы о степени различимости является следующая теорема.

Теорема 3*). *Для того чтобы два частных ограниченно-детерминированных оператора θ_1 и θ_2 веса $\leq K$ совпадали, необходимо и достаточно, чтобы они не были различимы никаким простым экспериментом длины $\leq 2K - 1$.*

Необходимость тривиальна. Для проверки достаточности рассмотрим общий ограниченно-детерминированный оператор θ , состоящий из частных операторов θ_1 и θ_2 и из всех их остаточных операторов. Вес θ не больше суммы весов θ_1 и θ_2 , т. е. вес $(\theta) \leq 2K$; следовательно, в соответствии с теоремой, степень неразличимости оператора θ не больше $2K - 1$. Итак, если θ_1 и θ_2 не являются $(2K - 1)$ -различимыми, то они уже вообще неразличимы.

Приведем пример, показывающий, что оценки, содержащиеся в теореме и ее следствии, не могут быть улучшены.

*) См. Э. Ф. Мур, Умозрительные эксперименты с последовательностными машинами, Сб. «Автоматы», ИЛ, М., 1956; Б. А. Трахтенброт, Об операторах, реализуемых в логических сетях, ДАН СССР, 1957, т. 112, № 6.

Рассмотрим константные операторы θ_1 и θ_2 , вырабатывающие соответственно периодические последовательности:

11... 1011... 10... ($K-1$ единица в периоде)

11... 101... 101... ($K-2$ единицы в периоде и одна в предпериоде).

Непосредственной проверкой убеждаемся, что вес каждого из этих операторов в точности равен K . Эти операторы неразличимы экспериментом длины $2K-2$, но различимы экспериментом длины $2K-1$.

Вместе с тем легко привести примеры, когда степень различимости оператора значительно меньше его веса. Это следует принимать во внимание при установлении верхней оценки для степени различимости (см. ниже, гл. V, § 2).

П р и м е р. Пусть из каких-либо соображений известно, что для каждого из двух исследуемых автоматов объем внутренней памяти не более 3 (например, известно, что это логические сети, в состав каждой из которых, кроме истинностных элементов, входят не более чем по 3 задерживающих элемента). Предположим далее, что зафиксированы какие-нибудь начальные состояния этих автоматов q' и q'' , которые неразличимы никаким экспериментом длины $\leq 2 \cdot 2^3 - 1 = 15$. При этих условиях первый автомат при начальном состоянии q' реализует тот же частный оператор, что и другой автомат при начальном состоянии q'' .

§ 4. Периодичность

В этом параграфе будут установлены характерные свойства ограниченно-детерминированных операторов, связанные с переработкой *периодических* входных последовательностей. Рассмотрение этого вопроса вызвано тем; что на практике автоматические устройства часто предназначаются именно для переработки такого рода последовательностей. При этом, естественно, возникает вопрос о том, будет ли и выходная информация периодической, и — в случае, если это будет иметь место, — вопрос о том, какая существует связь между периодами входных и выходных последовательностей.

Предварительно напомним и уточним соответствующие терминологию и обозначения.

Последовательность $\{U(n)\}$ называется *чисто периодической*, если существует такое целое число $\omega \geq 1$ (*период*), что $U(n) = U(n + \omega)$ при любом n . В этом случае существует бесчисленное множество периодов, совпадающих со всевозможными кратными минимального периода последовательности. Последовательность $\{U(n)\}$ называется *периодической*, если существует такое $\mu \geq 0$, что остаточная последовательность $U(\mu + 1) U(\mu + 2) \dots U(\mu + \xi)$ является чисто периодической.

Среди всевозможных таких μ существует наименьшее, которое мы назовем *длиной переходного процесса (предпериода)*. Кроме того, все чисто периодические остаточные последовательности имеют одни и те же периоды, поэтому можно говорить о минимальном периоде ω периодической последовательности.

Сохраняя обозначения μ и ω для минимальных значений указанных параметров, зададим периодическую последовательность в следующей форме:

$$U(1) U(2) \dots U(\mu) [U(\mu + 1) \dots U(\mu + \omega)],$$

где до квадратных скобок выписан переходный процесс (предпериод), а в квадратных скобках — периодическая часть (сокращенно — период); число $\mu + \omega$ назовем *приведенной длиной* последовательности $\{U(n)\}$.

Если $\mu < s < t$, причем $s = t \pmod{\omega}$, то остаточные последовательности

$$\begin{aligned} U(s) \quad U(s+1) \dots U(s+m) \dots, \\ U(t) \quad U(t+1) \dots U(t+m) \dots \end{aligned}$$

совпадают, т. е. $U(s+m) = U(t+m)$ при любом $m = 0, 1, 2, \dots$ Обратно, если для какой-либо последовательности $\{V(n)\}$ совпадают остаточные последовательности, начинающиеся соответственно с членов $V(s)$ и $V(t)$, то эта последовательность периодична, причем $\mu \leq s$, а $t - s$ является одним из ее периодов (не обязательно минимальным), в частности, приведенная длина такой последовательности меньше, чем $s + (t - s)$, то есть меньше t .

Т е о р е м а 4. *Ограниченно-детерминированный оператор θ с весом K преобразует любую периодическую*

последовательность $\{f(n)\}$ с периодом ω и с длиной μ преобразуется в периодическую последовательность $\{g(n)\}$ с периодом $\Omega = K\omega$ и с приведенной длиной $\leq \mu + K\omega$.

Доказательство. Пусть $\{q(n)\}$ — последовательность состояний, соответствующая входной последовательности $\{f(n)\}$. В последовательности пар

$$\langle f(1), q(1) \rangle \langle f(2), q(2) \rangle \dots \langle f(i), q(i) \rangle \dots \langle f(i+1), q(i+1) \rangle \dots \quad (4.4)$$

i -й член (пара) однозначно определяет i -й элемент последовательности $\{g(n)\}$. Поэтому для доказательства теоремы достаточно установить периодичность последовательности (4.4) и справедливость неравенства, приведенного в формулировке теоремы. Для этого рассмотрим следующую систему $K\omega + 1$ членов последовательности (4.4):

$$\langle f(\mu + 1), q(\mu + 1) \rangle \dots \langle f(\mu + K\omega + 1), q(\mu + K\omega + 1) \rangle \dots$$

Поскольку по условию теоремы различных состояний не больше K (так как вес оператора θ равен K), то по крайней мере одно состояние встречается в этой системе не меньше чем $(\omega + 1)$ раз. Среди соответствующих $(\omega + 1)$ пар системы найдутся по крайней мере две такие, для которых $t \equiv s \pmod{\omega}$ (обозначим их через $\langle f(s), q(s) \rangle$ и $\langle f(t), q(t) \rangle$); следовательно, остаточные последовательности

$$\begin{array}{cccc} g(s) & g(s+1) & \dots & g(s+m), \\ g(t) & g(t+1) & \dots & g(s+m) \end{array}$$

совпадают. Поскольку $g(s) = g(t)$, то и остаточные последовательности

$$\begin{array}{cccc} q(s) & q(s+1) & \dots & q(s+m), \\ q(t) & q(t+1) & \dots & q(s+m) \end{array}$$

совпадают. Таким образом, и в последовательности пар (4.4) совпадают соответствующие остаточные последовательности. Поскольку $\mu < s < t \leq \mu + K\omega + 1$, то это как раз означает, что последовательность (4.4) периодична, причем ее период $\Omega = t - s \leq K\omega$, а приведенная длина $t - 1 \leq \mu + K\omega$. Теорема, таким образом, доказана.

Пример. Оператор, заданный канонической таблицей IV.4, при начальном условии $q(1)=0$ перерабатывает последовательность 111... (с периодом единичной длины) в последовательность 01 [110] (где 110— период).

Таблица IV.4

$x \backslash q$	0	1	2	3	4	5	6	7
0	0/0	1/2	0/3	1/1	0/1	1/7	0/2	1/4
1	0/1	1/7	0/2	1/4	0/4	1/6	0/7	1/5

Вход $x(t)$: 1 1 1 1 1 1 . . .
 Состояние $q(t)$: 0 1 7 5 6 7 . . .
 Выход $z(t)$: 0 1 1 1 0 1 . . .

⏟
период

Оператор, заданный канонической таблицей IV.5, перерабатывает входную последовательность 101010... в последовательность 0111 [00100].

Таблица IV.5

$x \backslash q$	0	1	2	3	4	5	6	7
0	0/0	1/2	1/3	0/1	0/1	1/7	1/2	0/4
1	0/1	1/7	1/2	0/4	0/4	1/6	1/7	0/5

Вход: 1 0 1 0 1 0 1 0 1 0 . . .
 Состояние: 0 1 2 2 3 4 1 7 4 4 . . .
 Выход: 0 1 1 1 0 0 1 0 0 0 . . .

⏟
период

Пусть θ — константный оператор с весом K . Применяя предыдущую теорему к входной последовательности 0, 0, 0, ..., для которой $\mu=0$, $\omega=1$, мы устанавливаем, что оператор θ вырабатывает периодическую последовательность

$$\beta(1) \dots \beta(\mu) [\beta(\mu+1) \dots \beta(\mu+\Omega)], \quad (4.5)$$

для которой $\mu + \Omega \leq 0 + K \cdot 1 = K$. С другой стороны, для константного оператора, вырабатывающего последовательность (4.5), имеет место и обратное неравенство $\mu + \Omega \geq K$, ибо θ можно задать каноническими уравнениями с $(\mu + \Omega)$ -значным алфавитом состояний $q_1, q_2, \dots, q_\mu, q_{\mu+1}, \dots, q_{\mu+\Omega}$:

$$\left. \begin{aligned} \Phi(a, q_i) &= \beta(i), \\ \Psi(a, q_i) &= q_{i+1 \pmod{\mu+\Omega}}, \end{aligned} \right\} \quad (4.6)$$

где a — произвольная входная буква. Итак, справедлива следующая теорема.

Теорема 5*). *Константный оператор вырабатывает периодическую последовательность, приведенная длина которой в точности равняется его весу. Всякая периодическая последовательность с приведенной длиной μ вырабатывается некоторым константным оператором веса $K = \mu$.*

П р и м е ч а н и е. Теорема 4 не дает для длины предпериода выходной последовательности оценки, зависящей от длины μ предпериода входной последовательности и от веса оператора θ . В частности, чисто периодическая последовательность может перерабатываться в последовательность с непустым предпериодом.

Следующий пример показывает, что оценки, содержащиеся в теореме 4, в известном смысле не поддаются большому уточнению. Для любого k определим счетчик k -кратности как оператор $g(n) = \theta[f(n)]$ с двузначными входным и выходным алфавитами, работающий следующим образом: если в конечной последовательности $f(1), f(2), \dots, f(n)$ число единиц кратно k , то $g(n) = 1$, в противном случае $g(n) = 0$. Легко видеть, что чисто периодическая последовательность

$$(00 \dots 0, 1)$$

с периодом ω , содержащим $(\omega - 1)$ нулей, перерабатывается счетчиком k -кратности в чисто периодическую последовательность

$$\overbrace{[11 \dots 10]}^{(\omega-1) \text{ единиц}} \quad \overbrace{00 \dots 00}^{\omega \text{ нулей}} \dots \overbrace{00 \dots 01}^{(\omega-1) \text{ нулей}} \quad (k \text{ групп по } \omega \text{ букв})$$

*) См. упоминавшуюся неоднократно статью Бёркса и Райта «Theory of logical nets».

с минимальным периодом $\Omega = \omega k$. Вместе с тем этот оператор может быть описан каноническими уравнениями с j -значным алфавитом состояний $(q_0, q_1, \dots, q_{k-1})$:

$$\left. \begin{aligned} \Phi(1, q_{k-1}) &= 1, & \Phi(0, q_0) &= 1, & \Phi(0, q_i) &= 0; \\ \text{в прочих случаях:} & & & & & \\ \Psi(0, q_i) &= q_i, & \Psi(1, q_i) &= q_j & [j = (i+1) \pmod j]. \end{aligned} \right\} \quad (4.7)$$

Таким образом, для любых k и ω существует ограниченно-детерминированный оператор веса j , увеличивающий входной период ω в точности в j раз.

Свойство периодичности ограниченно-детерминированных операторов, указанное в теореме 4, и сам процесс доказательства этой теоремы могут оказаться полезными при анализе работы устройств, реализующих эти операторы. Именно, если даны канонические уравнения оператора, то для любой входной периодической последовательности, заданной в приведенной форме, можно эффективно указать приведенную форму соответствующей выходной последовательности.

Может возникнуть вопрос о том, имеет ли место обратное утверждение, а именно: если задан детерминированный оператор и известно, что он перерабатывает всякую периодическую последовательность в периодическую последовательность с удлинением не более чем в k раз и с соблюдением соответствующей оценки для приведенной длины, то можно ли утверждать, что он является ограниченно-детерминированным оператором? Можно ли дать оценку веса оператора посредством этого числа k ? Отрицательный ответ на поставленные вопросы может быть обоснован путем разбора следующего примера.

Предположим, что в дереве со входным алфавитом $<0,1>$ вдоль пути $101001000100001\dots$ ребрам приписаны выходные буквы, одноименные с входными, а всем остальным ребрам приписаны нули. Полученное дерево задает детерминированный оператор θ , перерабатывающий указанную входную последовательность $g(n)$ в ее самое. Любая другая последовательность, начиная с того места, где возникает отклонение от указанной, порождает на выходе только нули, т. е. перерабатывается в последо-

1) Распознавание правильной организации схемы

Как уже отмечалось, не всякая схема над системой элементов $\{M_i\}$ является логической сетью. Введенные в § 4—5 гл. II понятия позволяют сформулировать следующий критерий.

Т е о р е м а 6).* Для того чтобы схема над системой $\{M_i\}$ была логической сетью, необходимо и достаточно выполнение двух условий: а) ни в какой полюс не втекает два или более выходных каналов и б) в схеме отсутствуют неправильные петли.

Д о к а з а т е л ь с т в о. Необходимость выполнения условий а) и б) устанавливается индукцией по пунктам (правилам) 2, 3 и 4 определения понятия «логическая сеть» (см. стр. 122), если учесть, что для элементов $\{M_i\}$ эти условия выполнены. Фактически отождествление полюсов при построении сети регламентируется правилами 3 и 4, которые не предусматривают отождествления выходных полюсов (условие а)). Применение правила 3 вообще не порождает новых петель; если же в результате применения правила 4 и возникает новая петля (которая должна обязательно проходить через отождествленные полюсы z_j и x_i), то ее правильность обеспечивается соблюдением условия неподчиненности z_j полюсу x_i .

Перейдем к доказательству достаточности условий а) и б). Пусть схема M над $\{M_i\}$ удовлетворяет условиям а) и б). В таком случае ее построение может быть осуществлено, исходя из заданных элементов, следующим образом:

1) многократным применением правила 2 строится логическая сеть N_1 , имеющая тот же состав элементов, что и M , причем в N_1 никакие полюсы элементов не отождествлены;

2) последовательным применением правила 3 отождествляются те входные полюсы в N_1 , которые отождествлены в M ; полученную логическую сеть обозначим через N_2 ;

3) поскольку для M выполнено условие а), то для того, чтобы превратить N_2 в M , остается еще осуществить ряд актов отождествления выходного полюса с входным.

*) См. статью Бёркса и Райта «Theory of logical nets».

Поэтому достаточно показать, что, в каком бы порядке эти акты ни осуществлялись, ни при одном из них выходной полюс α_1 не будет отождествлен с входным полюсом α_n , которому он подчинен. Это действительно так, ибо в противном случае в результате такого отождествления получилась бы как раз неправильная петля, которая и после остальных актов отождествления, завершающих превращение N_2 в M , оставалась бы неправильной петлей в M . По условию же теоремы в M отсутствуют такие петли. Теорема таким образом доказана.

Применение этого критерия к схеме рис. 2.15 показывает, что она не является логической сетью, ибо, как было показано в § 4 гл. II, в ней имеется неправильная петля. Схема на рис. 2.17 является логической сетью; она отличается от схемы на рис. 2.15 тем, что в нее входит задерживающий элемент \mathfrak{M}_3 , который превращает неправильную петлю схемы рис. 2.15 в правильную петлю, отмеченную утолщенной линией. На рис. 2.16, а и 2.18 изображены логические сети, соответствующие некоторым промежуточным стадиям построения сети на рис. 2.17.

Условимся схему называть *деревообразной*, если в ней вообще отсутствуют петли. Из теоремы непосредственно вытекает следующее следствие.

С л е д с т в и е. Всякая деревообразная схема над $\{\mathfrak{M}_i\}$, в каждый полюс которой втекает не более одного выходного канала, является логической сетью над $\{\mathfrak{M}_i\}$.

В тех случаях, когда в схеме содержится много петель (или длинные петли), применение критерия теоремы требует большого перебора. Мы укажем в этой связи одно видоизменение этого критерия, которое хотя и не элиминирует громоздкости перебора, все же может оказаться полезным, в особенности для составления системы канонических уравнений ограниченно-детерминированного оператора логической сети, исходя из канонических уравнений операторов отдельных элементов.

Поскольку проверка условия а) теоремы не представляет труда, мы ограничимся теперь рассмотрением лишь таких схем, для которых это условие выполнено.

Назовем *правильной нумерацией схемы* такую нумерацию всех ее полюсов $\xi_1, \xi_2, \dots, \xi_i, \dots, \xi_n$, которая обладает следующим свойством: полюс с номером i не подчи-

нен полюсам с номерами $j \geq i$. Если схема допускает правильную нумерацию, то она является логической сетью.

Действительно, наличие неправильной петли повлекло бы за собой существование полюса ξ_i , подчиненного самому себе, т. е. $i < i$. Легко устанавливается и обратное утверждение: всякая логическая сеть допускает правильную нумерацию *). Такую нумерацию доставляет следующая процедура отнесения рангов к полюсам логической сети.

Отнесем ранг 0 ко всякому полюсу, который не подчинен никакому полюсу схемы (в том числе не подчинен самому себе). В частности, ранг 0 сопоставляется любому входному полюсу логической сети. Полюсы нулевого ранга обязательно существуют, ибо всякая цепь полюсов z_1, z_2, \dots такая, что $z_1 > z_2 > z_3 > \dots$, в логической сети обрывается как раз на таком полюсе (иначе возникла бы неправильная петля).

Пусть уже определены все полюсы рангов, меньших или равных n . Тогда ранг $n + 1$ приписывается каждому такому полюсу, который подчинен только полюсам ранга не выше n , причем в числе полюсов, которым он подчинен, имеется по крайней мере один полюс ранга n . После конечного числа шагов процесс отнесения рангов обрывается.

Покажем еще, что таким образом всем полюсам схемы будут отнесены ранги. Рассуждая от противного, предположим, что существует некоторый полюс α_1 , не получивший ранга. Тогда α_1 не является полюсом нулевого ранга, а также не подчинен самому себе (неправильных петель нет), поэтому он подчинен каким-то другим полюсам $\beta, \gamma, \dots, \delta$. Среди этих последних имеется по крайней мере один (обозначим его через α_2), которому тоже не был приписан ранг, ибо в противном случае полюсу α_1 был бы приписан ранг на единицу больше максимума рангов полюсов $\beta, \gamma, \dots, \delta$. Повторение этих рассуждений приводит к противоречию, ибо порождает бесконечную последовательность полюсов $\alpha_1, \alpha_2, \alpha_3, \dots$ (к которым не отнесены ранги) таких, что $\alpha_1 < \alpha_2, \alpha_2 < \alpha_3$, чего быть не может в логической сети.

*) Множество полюсов в логической сети частично упорядочено отношением подчиненности; далее следует развернутое описание нумерации частично упорядоченного множества.

После того как к полюсам отнесены ранги, нетрудно получить правильную нумерацию, если приписать номера (в порядке естественного возрастания) сначала полюсам нулевого ранга, потом полюсам первого ранга и т. д.

Пример. В логической сети, изображенной на рис. 2.17, полюсы x , z_4 имеют нулевой ранг, z_1 и z_2 — ранг 1 и z_3 — ранг 2. Условимся называть рангом логической сети максимальный из рангов ее полюсов. Тогда ранг логической сети, изображенной на рис. 2.17, равен 2.

2) Составление канонических уравнений

При анализе схемы отнесение рангов к полюсам удобно также для составления канонических уравнений ограниченно-детерминированного оператора, реализуемого в логической сети. Проиллюстрируем это на примере логической сети N , изображенной на рис. 2.17. Система уравнений $E(N)$, получаемая путем объединения уравнений составных элементов в сети N , приведена на стр. 116 (система 2.12); при этом входные и выходные переменные переименованы в соответствии с наименованиями полюсов, которым они сопоставлены.

Отнесем теперь к уравнениям выходных переменных те же ранги, что и к сопоставляемым им полюсам, и выпишем их в порядке, соответствующем правильной нумерации полюсов. Затем выпишем уравнения состояний. Система $E(N)$ примет вид:

$$\left. \begin{aligned} z_4(t) &= r(t), \\ z_1(t) &= z_4(t) \oplus x(t) \oplus q(t), \\ z_2(t) &= z_4(t) \oplus p(t), \\ z_3(t) &= z_1(t) \oplus z_2(t), \\ r(t+1) &= z_3(t), \\ q(t+1) &= z_4(t) x(t) \vee q(t) z_4(t) \vee q(t) x(t), \\ p(t+1) &= z_4(t) \oplus p(t), \\ r(0) &= q(0) = p(0) = 0. \end{aligned} \right\} \quad (4.10)$$

Для получения системы канонических уравнений достаточно исключить из правых частей уравнений системы

(4.10) выходные переменные, встречающиеся в них. В правой части уравнения выходного переменного ранга r могут встречаться выходные переменные лишь меньшего ранга s . Поскольку эти уравнения расположены в порядке возрастания их рангов, то нужное исключение переменных непосредственно достигается путем последовательной подстановки правой части каждого уравнения вместо левой его части всюду, где эта левая часть встречается в следующих за ним уравнениях.

В нашем случае в результате таких подстановок уравнения выходных переменных принимают вид:

$$\left. \begin{aligned} z_4(t) &= r(t), \\ z_1(t) &= r(t) \oplus x(t) \oplus q(t), \\ z_2(t) &= r(t) \oplus p(t), \\ z_3(t) &= r(t) \oplus x(t) \oplus q(t) \oplus r(t) \oplus p(t) = \\ &= x(t) \oplus q(t) \oplus p(t). \end{aligned} \right\} \quad (4.11)$$

Теперь можно исключить выходные переменные из правых частей уравнений состояний:

$$\left. \begin{aligned} r(t+1) &= x(t) \oplus q(t) \oplus p(t), \\ q(t+1) &= r(t)x(t) \vee q(t)r(t) \vee q(t)x(t), \\ p(t+1) &= r(t) \oplus p(t). \end{aligned} \right\} \quad (4.12)$$

Система (4.11) вместе с системой (4.12) и образует систему канонических уравнений для оператора рассматриваемой сети.

Канонические таблицы значений (в двоичном коде) выходных переменных z_1, z_2, z_3, z_4 и значений состояний, составленные в соответствии с этой системой, имеют вид, показанный в таблице IV.6.

В соответствии с обобщением понятия реализуемости ограниченно-детерминированного оператора данной логической сетью (см. § 6 гл. II) рассмотрим ограниченно-детерминированный оператор, реализуемый логической сетью, изображенной на рис. 2.17; в одном лишь полюсе z_2 . Его система канонических уравнений включает лишь

Таблица IV.6

$x \backslash qpr$								
	000	001	010	011	100	101	110	111
Выходная таблица								
0	0000	1101	0110	1011	1010	0111	1100	0001
1	1010	0111	1100	0001	0000	1101	0110	1011
Таблица состояний								
0	000	010	011	001	001	111	010	100
1	001	111	010	100	100	110	111	101

одно следующее уравнение из подсистемы (4.11):

$$z_2(t) = r(t) \oplus p(t).$$

Канонической таблицей этого оператора является рассмотренная выше таблица IV.5 (буквы состояний для большей обозримости записаны в десятичной системе). Если же рассматривать, например, полюс z_4 , то соответствующий ему ограниченно-детерминированный оператор имеет каноническую таблицу IV.4.

3) Минимизация алфавита состояний

Ниже будет описана процедура, позволяющая по заданным каноническим таблицам осуществить разбиение соответствующего алфавита состояния Q на систему подмножеств (классов)

$$Q_1 Q_2 \dots Q_s \quad (4.13)$$

таких, что две буквы попадают в один класс в том и только в том случае, когда они задают неразличимые состояния. Этим, собственно, и решается задача минимизации.

Разбиение (4.13) алфавита Q мы будем называть *п о л н ы м* его разбиением. Названная процедура заключается в последовательном построении разбиений $\{\mathcal{B}_r^{(p)}\}$ для $p = 1, 2, \dots$, ст (6) так, как это было сделано при доказательстве теоремы из § 3 о степени различимости.

Предварительно обратим внимание на условие, согласно которому два состояния q_i и q_j , принадлежащие какому-либо классу $\mathfrak{B}_j^{(p)}$, являются $(p+1)$ -различимыми (а значит, и строго $(p+1)$ -различимыми). Для выполнения этого условия необходимо и достаточно существование такой входной буквы x , которая переводила бы их в p -различимые состояния q'_i и q'_j (принадлежащие различным классам $\{\mathfrak{B}_j^{(p)}\}$). Будем, далее, считать, что оператор задан каноническими таблицами с двойным входом.

Рассмотрим случай, когда $p=1$. В этом случае в один класс объединяются все такие буквы состояний, которым в выходной канонической таблице соответствуют одинаковые столбцы (выходных букв).

Осуществим теперь переход от p к $p+1$. Пусть уже построено разбиение $\{\mathfrak{B}_j^{(p)}\}$. Два столбца внутренних букв канонической таблицы состояний назовем *эквивалентными относительно разбиения* $\{\mathfrak{B}_j^{(p)}\}$, если любая строка пересекает их в элементах, принадлежащих одному его классу. Две буквы q_i и q_j , попавшие в один класс $\mathfrak{B}_s^{(p)}$, останутся также в одном классе $\mathfrak{B}_s^{(p+1)}$ в том и только в том случае, когда выполнено условие: в канонической таблице состояний столбцы, занумерованные буквами q_i и q_j , эквивалентны относительно разбиения $\{\mathfrak{B}_j^{(p)}\}$.

Процесс обрывается тогда, когда для любых двух элементов любого класса $\{\mathfrak{B}_j^{(p)}\}$ соответствующие им столбцы эквивалентны относительно этого разбиения. В таком случае $\{\mathfrak{B}_j^{(p)}\}$ является уже полным разбиением. Очевидно, это произойдет при p , равном степени различимости оператора.

П р и м е р. Рассмотрим канонические таблицы, соответствующие оператору, отнесенному к полюсу z_2 сети, изображенной на рис. 2.17, и в соответствии с описанной процедурой построим разбиения (см. канонические таблицы IV.5):

$$p = 1: \{0, 3, 4, 7\}, \{1, 2, 5, 6\},$$

$$p = 2: \{0, 7\}, \{3, 4\}, \{1, 6\}, \{2, 5\},$$

$$p = 3: \{0\}, \{7\}, \{3, 4\}, \{1, 6\}, \{2\}, \{5\}.$$

Классы $\{3, 4\}$ и $\{1, 6\}$ больше не разбиваются, что же касается остальных классов, то они являются уже одноэлементными. Итак, вес рассматриваемого оператора равен 6, а его степень различимости равна 3. Минимизированные таблицы получаются вычеркиванием из заданных таблиц столбцов, соответствующих, например, буквам 4 и 6, с заменой в уцелевших столбцах таблицы состояний букв 4 и 6 соответственно буквами 3 и 1 (см. таблицу IV.7).

Таблица IV.7

	0	1	2	3	5	7
0	0/0	1/2	1/3	0/1	1/7	0/3
1	0/1	1/1	1/2	0/3	1/1	0/5

Аналогичная проверка оператора, отнесенного к полюсу z_4 , а также таблицы IV.6 оператора, отнесенного ко всем выходным полюсам сети, показывает, что в первом случае все 8 букв попарно 3-различимы, а во втором случае — даже 1-различимы. Для z_4 мы имеем следующие разбиения:

$$p = 1: \{0, 2, 4, 6\}, \{1, 3, 5, 7\},$$

$$p = 2: \{0, 6\}, \{2, 4\}, \{1, 7\}, \{3, 5\},$$

$$p = 3: \{0\}, \{6\}, \{2\}, \{4\}, \{1\}, \{7\}, \{3\}, \{5\}.$$

ГЛАВА V

СПОСОБЫ ЗАДАНИЯ ОПЕРАТОРОВ

§ 1. Общие замечания

Настоящая и последующие главы посвящены задаче синтеза автомата, предварительная формулировка которой была дана в главе II. Однако в этой формулировке необходимо уточнить, каков должен быть исходный материал (исходная информация), к которому будут применяться методы, дающие как результат соответствующие логические сети. Уточнение может быть достигнуто, например, путем объявления канонических таблиц (уравнений) оператора такой стандартной формой его задания, исходя из которой и проводится дальнейшее исследование задачи синтеза. Именно этой точки зрения мы и будем придерживаться в последующих двух главах. При таком подходе синтез логической сети можно рассматривать как перевод со стандартного языка канонических уравнений на язык логической схемы («схемный» язык), зависящий от предписанного набора базисных элементов. Вместе с тем анализ логической сети оказывается переводом со «схемного» языка на стандартный.

Заметим, однако, что любое соглашение о стандартной форме задания оператора сразу порождает другие вопросы, решение которых, по существу, представляет собой составную часть решения общих задач синтеза и анализа. Дело в том, что то преобразование информации (то есть тот оператор), которое должно быть реализовано в логической сети, зачастую сообщается в виде предписания, совсем непохожего на избранную стандартную форму (в нашем случае—на канонические таблицы) задания оператора. Таким образом, естественно, возникает задача перехода

от первоначального предписания к стандартной форме. Точно так же при анализе схемы описание ее работы может оказаться более обозримым и естественным не в терминах стандартного языка, а в терминах какого-либо другого, более удобного (с некоторой точки зрения) языка. Тогда может возникнуть задача перевода описания работы схемы со «схемного» языка на этот более «удобный» язык, и этот перевод может осуществляться либо прямо, либо при помощи стандартного языка «посредника».

В настоящей главе рассматривается первый этап синтеза автомата, связанный с получением канонических уравнений оператора, исходя из различных способов его задания.

Начнем с рассмотрения некоторых простых примеров, иллюстрирующих ситуацию, которая здесь может возникнуть.

Ниже формулируются четыре предписания о переработке входных последовательностей в выходные; для каждого из них ставится вопрос о построении соответствующей логической сети.

1. Входной алфавит: 000, 001, ..., 110, 111 (двоичные записи чисел 0, 1, ..., 7); выходной алфавит тот же. Преобразование: любое x ($0 \leq x \leq 7$) переходит в любом такте в $x+1 \pmod{7}$.

2. Входной алфавит: 0, 1; выходной тот же. Преобразование: $z(t)=1$ тогда и только тогда, когда $x(t)=1$ и число единиц среди $x(1), x(2), \dots, x(t)$ четное.

3. Входной алфавит: 000, 001, ..., 110, 111 (двоичные записи чисел 0, 1, ..., 7); выходной алфавит: 0, 1. Преобразование: $z(t)=1$ в том и только в том случае, если существует такой такт σ , предшествующий текущему такту t , в котором старший разряд входного числа равен 1, причем если между σ и t еще имеются промежуточные такты, то для каждого такого такта τ имеет место по крайней мере один из следующих двух случаев: 1) в такте τ второй разряд равен 1; 2) существует такт λ ($\sigma < \lambda < \tau$), в котором младший разряд равен 1.

На рис. 5.1 показано соотношение указанных в предписании тактов времени.

Отметим, что если в этом примере разрядам двоичного числа, взятым слева направо, сопоставить свойства: «быть прописной буквой», «быть согласной буквой» и

«быть типично русской буквой» (а выполнению соответствующего свойства приписать значение 1, а невыполнению — значение 0), то мы приходим к естественной интерпретации примера, упомянутого во введении.

4. Входной алфавит 0,1; выходной тот же. Преобразование:

$z(t) = 1$ в том, и только в том случае, когда $x(t) = 1$ и число единиц среди $x(1), x(2), \dots, x(t)$ является простым.

5. Входной алфавит: a, b, c ; выходной: 0,1. Преобразование: последовательности

$abcbcbca, abacaba$

перерабатываются соответственно в последовательности

0 0 1 0 1 0 1 и 0 0 1 1 0 1 1.

Для остальных последовательностей в данном алфавите характер преобразования безразличен.

Допустим теперь, что для каждого из приведенных предписаний ставится задача построить хотя бы одну какую-либо логическую сеть, реализующую соответствующее преобразование информации. При ближайшем рассмотрении оказывается, что предписание 4 задает детерминированный оператор, который не является ограниченным, а следовательно, не может быть вообще реализовано в логической сети. Что же касается первых трех предписаний, то они задают ограниченно-детерминированные операторы $\theta_1, \theta_2, \theta_3$, причем θ_1 является истинностным оператором; приводимая ниже таблица V.1 является канонической для оператора θ_3 .

При наличии же канонических таблиц операторов логические сети, реализующие их, как мы знаем, могут быть всегда построены (если только исходный набор элементов является полным).

Таким образом, когда ставится задача о построении сети (напоминаем, что имеется в виду к а к а я-н и б у д ь сеть, не обязательно оптимальная) по некоторому предписанию, то вся трудность сводится, по существу, к распознаванию того, задает ли это предписание ограниченно-детерминированный оператор (*распознавание возможности реализации*), и в случае, если это так, то еще и в переходе

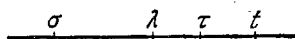


Рис. 5.1.

Таблица V. 1

$x_1x_2x_3$	00	01	10	11
000	0/00	0/11	1/00	1/11
001	0/00	0/11	1/00	1/11
010	0/00	0/11	1/10	1/11
011	0/00	0/11	1/11	1/11
100	0/10	0/11	1/10	1/11
101	0/10	0/11	1/10	1/11
110	0/10	0/11	1/10	1/11
111	0/10	0/11	1/11	1/11

к соответствующей канонической таблице. Очевидно, такое исследование существенно зависит от того, в каких терминах допустимы первоначальные предписания.

Рассмотрим сначала тот случай, когда на первоначальное описание не наложено никаких специальных требований, кроме следующего: для любой входной последовательности (конечной, но произвольной длины) соответствующая выходная последовательность должна быть четко и эффективно указана этим предписанием.

Представим эту ситуацию следующим образом: «заказчик» логической сети затрудняется описать интересующий его оператор в терминах, обозримых для «исполнителя», но по требованию последнего может сообщить ему для любой конкретной входной последовательности, в какую именно она должна быть переработана. Так, например, в соответствии с предписанием 3 для входных последовательностей

000, 001, 010, 011, 100, 101, 110, 111, 011, 111;
000, 101, 001, 111, 010, 111

будут указаны выходные последовательности

0, 0, 0, 0, 0, 1, 1, 1, 1, 1;
0, 0, 1, 0, 1, 1.

Иными словами, это означает, что мы располагаем возможностью строить любые конечные части деревьев рассматриваемых операторов. На деревьях, изображенных

веса, степени различимости, степени достижимости и т. п.). Эти вопросы и некоторые связанные с ними количественные оценки будут рассмотрены в § 2.

Возможна и другая постановка вопроса, при которой считается, что первоначальные предписания сформулированы на определенном формальном языке (т. е. выражены в виде формул специального вида) и задача заключается в их преобразовании (т. е. в преобразовании формул) в соответствующие канонические таблицы. В этом случае возникает своеобразная задача перевода с одного формального языка на другой, ибо канонические таблицы тоже можно рассматривать как формальный язык описания операторов. Такая постановка задачи представляет интерес тогда, когда исходный язык является (в рассматриваемых случаях) более естественным, чем язык канонических таблиц.

В § 3 настоящей главы излагается способ задания оператора с помощью матриц, предложенных М. Л. Цетлиным *), и рассматривается алгоритм перевода с матричного языка на язык канонических уравнений и обратно.

Трудности, которые иногда возникают при попытке описать оператор непосредственно в виде канонических уравнений, могут быть зачастую уменьшены посредством некоторой их естественной модификации (осуществляемой средствами логики суждений). При этом оказывается возможным выразить в форме, более близкой к обычному языку, временные соотношения и логические условия. Связанная с этим типичная ситуация рассматривается в § 4, где одновременно выясняется целесообразность дальнейшего расширения логических средств, привлекаемых для формального описания оператора. В качестве такого языка предлагается язык логики предикатов, рассматриваемый в § 5.

§ 2. Фрагменты деревьев

Пусть детерминированный оператор θ перерабатывает всякое входное слово длины μ в выходное слово такой же длины. Рассмотрим какое-либо конечное множество вход-

*) М. Л. Цетлин, О непримитивных схемах, Сб. «Проблемы кибернетики», № 2, 1959.

ных слов (вообще говоря, неодинаковой длины). Переработка этих слов оператором θ будет изображаться соответствующим конечным деревом, являющимся фрагментом бесконечного дерева оператора θ . В частности, если рассматриваемое множество состоит из всех входных слов заданной длины, то такое конечное дерево мы условимся называть *полным деревом высоты μ* .

Задание операторов посредством конечных деревьев может оказаться целесообразным в следующих двух случаях:

1. Первоначальное описание оператора дано в таких терминах, которые затрудняют непосредственный переход к схеме автомата или хотя бы к каноническим уравнениям. При этом мы надеемся получить канонические уравнения путем анализа некоторого конечного фрагмента дерева. Эта ситуация была разъяснена в § 1.

2. Первоначальное предписание предусматривает лишь преобразование конечного числа конечных входных слов. При этом ставится задача определения такого ограниченно-детерминированного оператора, который реализует нужное преобразование указанных слов; что же касается других возможных входных слов или продолжений заданных слов, то на оператор специальных ограничений не накладывается. В этом случае само первоначальное задание предстает в виде конечного дерева, для которого нужно найти бесконечное продолжение с конечным весом. Разумеется, среди возможных продолжений предпочтение отдается оптимальному (в смысле того или иного принятого критерия). На практике такие случаи встречаются, например, тогда, когда проектируемый автомат предназначается для работы в течение заранее ограниченного промежутка времени (ограниченного числа тактов), после которого он приводится опять в начальное состояние путем вмешательства извне *).

В связи с вышеизложенным полезно рассмотреть подробнее конечные деревья и те преобразования, которые ими описываются (для этих преобразований мы сохраним

*) Например, секретный замок открывается или подает тревогу после заранее ограниченного числа тактов. Из состояния тревоги (или открытия) его переводят в начальное состояние.

термин «детерминированный оператор»). Прежде всего укажем, как перенести на такие конечные деревья (и операторы) некоторые понятия и предложения, установленные ранее.

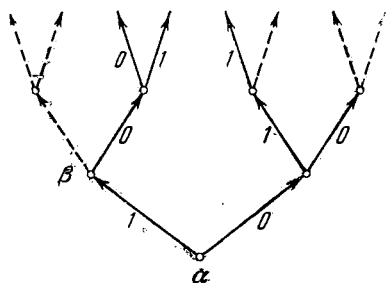


Рис. 5.4.

Понятия k -различимости и строгой k -различимости для двух конечных деревьев определяются по-прежнему: деревья считаются различимыми, если существует такое входное слово, которое их различает. Например, деревья, изображенные на рис. 5.4 и 5.5, 2-различимы (и даже строго 2-различимы) входным словом 11. Никакие другие входные слова вообще их не различают.

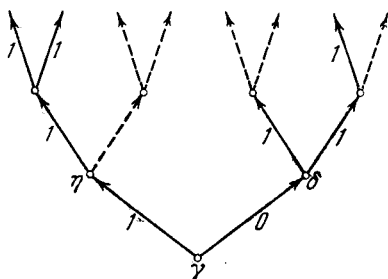


Рис. 5.5.

Понятия k -различимости и строгой k -различимости мы будем применять и к ветвям одного и того же конечного дерева, а также к корням этих ветвей. Так, например, в дереве, изображенном на

рис. 5.4, вершины α и β различимы входным словом 10. В дереве, изображенном на рис. 5.5, вершина γ 1-различима от вершины δ , но неразличима от вершины η .

Сохраняют свой прежний смысл также понятие *веса* и понятие *степени различимости*. Для дерева, изображенного на рис. 5.4, вес равен 2 и степень различимости тоже равна 2; для дерева, изображенного на рис. 5.5, вес равен 2, а степень различимости равна 1.

Из приведенного примера видно, что теорема 2 гл. IV об оценке степени различимости теряет силу применительно к конечным деревьям. Более того, для любого сколь угодно большого μ можно построить дерево с весом, равным 2, и степенью различимости большей чем μ . Например, на

рис. 5.6 слово $00\dots 01$ (μ нулей) преобразуется в самого себя, а слово $10\dots 01$ (μ нулей) — в $10\dots 00$ ($\mu+1$ нуль); вершины α , β здесь строго $(\mu+1)$ -различимы.

Отметим также, что отношение неразличимости или k -неразличимости, оставаясь по-прежнему симметричным, не является транзитивным, когда сравнению подлежат деревья с неодинаковыми высотами. Так, например, на рис. 5.7 каждое из деревьев V' и V'' неразлично от дерева V , но вместе с тем V' и V'' различимы друг от друга. Поэтому, в отличие от бесконечных деревьев, вообще говоря, невозможно разбиение множества всех ветвей конечного дерева на попарно непересекающиеся классы таким образом, чтобы две ветви были неразличимы тогда и только тогда, когда они принадлежат одному и тому же классу.

В дальнейшем среди конечных деревьев особый интерес будут представлять полные деревья. Понятие

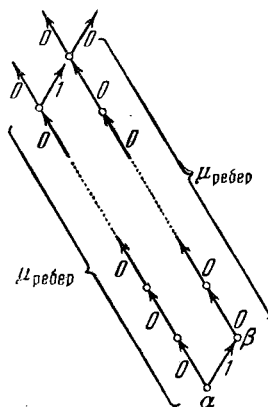


Рис. 5.6.

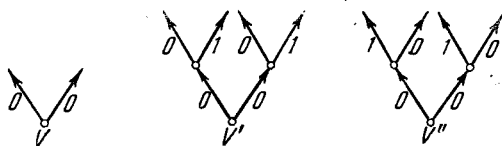


Рис. 5.7.

базиса определяется для них по-прежнему; однако тот факт, что число ветвей (вершин) в любом базисе одно и то же (и равно весу дерева), требует в данном случае другого обоснования, чем раньше. Причина заключается в том, что отношение неразличимости не является более транзитивным. Условимся также при отсутствии особых оговорок под базисом \mathfrak{B} полного дерева понимать базис, определяемый той процедурой, которая была принята в § 2 гл. IV для дерева ограниченно-детерминированного оператора.

вес и, следовательно, не задает ограниченно-детерминированного оператора. Если же ребрам из ярусов $h+1, \dots, h+k-1$ приписать нули, а всем ребрам из ярусов $h+k, h+k+1, \dots$ приписать единицы, то получится дерево V_k с конечным весом, но этот вес будет больше k (где k — произвольно взятое натуральное число).

Итак, любое заданное конечное дерево V может быть продолжено как в дерево с конечным, но сколь угодно большим весом, так и в дерево с бесконечным весом. Таким образом, становится ясным, что для содержательности дальнейших рассмотрений нужно исходить из некоторой верхней оценки для веса того оператора, который намереваются исследовать по его конечным деревьям.

Пусть заданы: конечное дерево V и натуральное число K . Возникают вопросы:

а) Можно ли продолжить V в бесконечное дерево ограниченно-детерминированного оператора веса, не большего чем K (иначе говоря, обладает ли дерево V свойством так называемой *K-продолжаемости*)?

б) Единственным ли образом осуществима K -продолжаемость?

В случае утвердительного решения вопросов а) или б) возникает задача:

в) Эффективно построить каноническую таблицу (или каноническую диаграмму) оператора, имеющего вес, не превышающий K , для которого V является начальным конечным деревом.

Понятие *веса конечного дерева* позволяет сформулировать тривиальную оценку для веса любого продолжения данного конечного дерева V высоты h и, в частности, для веса ограниченно-детерминированного оператора, совпадающего с V на нижних h ярусах. Для этого достаточно заметить, что K вершин базиса \mathfrak{B} дерева V останутся попарно различимыми и при любом его продолжении. Следовательно, если вес V равен K , то вес любого продолжения будет не меньше K .

Вместе с тем из приводимых ниже соображений видно, что всегда возможно продолжение дерева веса K в ограниченно-детерминированный оператор того же веса.

Каноническая таблица ограниченно-детерминированного оператора θ веса K , продолжающего данное конечное

дерево веса K , строится в результате последовательной реализации следующих операций:

1°. Выделяются K вершин конечного дерева V , образующие его базис \mathfrak{B} . Этим вершинам приписываются K различных букв, образующих алфавит состояний искомого оператора θ .

2°. Выделяется множество \mathfrak{B}' тех вершин, в которые втекают ребра, вытекающие из \mathfrak{B} ; в \mathfrak{B}' могут оказаться и некоторые вершины из \mathfrak{B} , к которым уже отнесены состояния. Что касается тех вершин из \mathfrak{B}' , которые не вошли в \mathfrak{B} , то каждой из таких вершин γ приписывается такая буква состояния, которая уже была отнесена какой-либо вершине β из \mathfrak{B} , неразличимой от γ .

Примечание. Для каждой вершины конечного дерева V и, в частности, для каждой вершины из \mathfrak{B}' найдется по крайней мере одна неразличимая от нее вершина в \mathfrak{B} . Если такая вершина \mathfrak{B} единственна, то и отнесение буквы состояния происходит однозначно; в противном же случае операция 2° неоднозначна.

3°. В канонической таблице искомого оператора θ столбец q_i заполняется в соответствии с выходными буквами, которые отнесены ребрам, выходящим из базисной вершины с состоянием q_i и с буквами состояний, отнесенными вершинам из \mathfrak{B}' , в которые втекают эти ребра.

Пример. На рис. 5.9 изображено дерево с весом $K=5$; в базис \mathfrak{B} входят вершины, которым отнесены состояния q_1, q_2, q_3, q_4, q_5 . Рядом с вершинами из \mathfrak{B}' , не

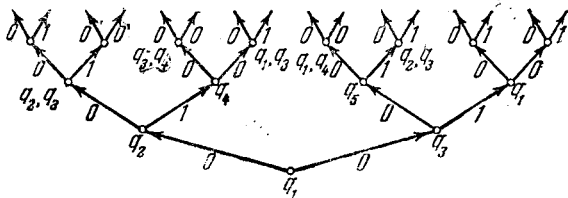


Рис. 5.9.

вошедшими в \mathfrak{B} , выписаны те состояния, которые могут быть к ним отнесены в соответствии с пунктом 2°. Ниже приведена каноническая таблица (таблица V.2), составлен-

ная в предположении, что вершинам из \mathfrak{B}' отнесены те буквы, которые стоят на первых местах.

Таблица V.2

$x \backslash q$	q_1	q_2	q_3	q_4	q_5
0	$0/q_2$	$0/q_2$	$0/q_5$	$0/q_1$	$0/q_1$
1	$0/q_3$	$1/q_4$	$1/q_1$	$0/q_2$	$1/q_2$

Мы не приводим формального доказательства того, что каноническая таблица, составленная согласно пунктам 1°—3°, действительно задает оператор, продолжающий заданное конечное дерево. Дополнительно отметим следующий просто проверяемый факт: всякий ограниченно-детерминированный оператор, K -продолжающий данное конечное дерево V веса K , может быть получен указанным путем.

Ответ на вопрос б), сформулированный выше на стр. 229, частично дает следующая теорема.

Теорема 1. *Для того чтобы дерево V высоты h допускало единственное продолжение с сохранением веса, достаточно выполнения условия*

$$h \geq q(V) + \text{ст}(V) + 1.$$

Доказательство.

При условиях, указанных в теореме, любая вершина δ из базиса \mathfrak{B} является корнем ветви высоты $> \text{ст}(V)$, а любая вершина γ из \mathfrak{B}' — корнем ветви высоты $\geq \text{ст}(V)$ (рис. 5.10). Следовательно, для всякой вершины γ из \mathfrak{B}' существует лишь единственная вершина базиса, неразличимая от вершины γ .

Следствие (см. также теорему 3 гл. IV). Для того чтобы дерево V высоты h и веса K допускало единственное продолжение, достаточно выполнения условия:

$$h \geq 2K - 1.$$

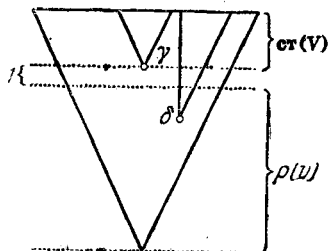


Рис. 5.10.

Вернемся теперь к задаче о получении канонических уравнений для ограниченно-детерминированного оператора θ путем построения такого его начального дерева (желательно с минимально возможной высотой), которое содержит полную информацию об операторе. Из предыдущих рассмотрений ясно, что для этого, помимо первоначального предписания, позволяющего эффективно строить начальное дерево любой высоты, необходимо располагать еще какой-либо предварительной верхней оценкой его параметров (веса, степени различимости, степени достижимости).

Заметим сначала, что если для высоты h начального дерева V справедливо, что $h \geq \text{ст}(\theta) + \varrho(\theta) + 1$, то все базисные вершины бесконечного дерева оператора θ уже различимы в V , вес которого, следовательно, равен уже весу дерева θ . Одновременно имеют место соотношения: $\text{ст}(V) = \text{ст}(\theta)$ и $\varrho(V) \leq \varrho(\theta)$; поэтому в соответствии с теоремой 1 дерево V допускает единственное продолжение с сохранением веса, которое и обязано совпадать с θ .

Отсюда вытекает следующая процедура для получения канонических уравнений оператора θ при наличии оценок $\text{ст}(\theta) \leq s$, $\varrho(\theta) \leq r$:

Э т а п 1. Строится начальное дерево V высоты $s + r + 1$; V допускает единственное продолжение с сохранением веса.

Э т а п 2. Составляются канонические уравнения в соответствии с пунктами 1°—3° стр. 230.

Если мы располагаем лишь оценкой следующего вида: $\text{вес}(\theta) \leq K$, то дерево V приходится брать с высотой $h = 2K - 1$. Уже при сравнительно небольших K это связано с очень громоздкими построением, перебором и сравнением ветвей этого дерева. Иногда бывает, что для $\text{ст}(\theta)$ и $\varrho(\theta)$ (или для одной из них при наличии оценки: $\text{вес}(\theta) \leq K$) имеются лучшие оценки, чем $\text{ст}(\theta) \leq K - 1$, $\varrho(\theta) \leq K - 1$; тогда соответственно и высоту h можно брать меньше.

Приведем еще некоторые соображения, позволяющие уменьшить отмеченную громоздкость.

Во-первых, отметим, что на самом деле вовсе не обязательно осуществлять описанную процедуру строго в указанные два этапа. Зачастую удобнее вести составление

канонических таблиц по ходу построения дерева, причем так, что отпадает необходимость в рассмотрении полного дерева и можно ограничиться построением менее обширных его частей. Покажем, например, как можно проводить указанный процесс при наличии оценки вида $st(\theta) \leq s$.

Идея заключается в том, что вершины базиса \mathfrak{B} , а также вершины из \mathfrak{B}' (и состояния, которые им следует приписывать) можно обнаруживать постепенно; по мере обнаружения указанных вершин ведется и заполнение соответствующих столбцов канонической таблицы. Процесс осуществляется следующим образом.

Строим дерево высоты $s+1$; его корень входит в базис, и ему приписывается состояние q_0 . Далее устанавливается попарная различимость или неразличимость для $(m+1)^*$ — вершин ранга ≤ 2 ; это позволяет выделить среди вершин ранга 2 те, которые должны быть отнесены к базису. Одновременно этим вершинам приписываются состояния, и составляется первый столбец в канонической таблице; этот столбец показывает, в какие состояния может переходить состояние q_0 . Далее, с каждой из выявленных вершин второго ранга в базисе поступают так же, как с корнем q_0 . Для этого приходится «поднимать» еще по одному ярусу, насчитывающему m^s ребер над уже имеющимися ветвями высоты s , выходящими из этих вершин. Путем сравнения полученных вершин третьего ранга между собой, а также с уже ранее построенными вершинами базиса определяется, какие из них следует отнести к базису (и приписать им новые буквы состояний) и какие состояния следует приписать тем вершинам третьего ранга, которые не входят в базис. Естественный обрыв этого процесса происходит тогда, когда впервые среди вершин данного ранга не обнаружено вершин, отличимых от ранее выявленных вершин базиса. Тогда завершается и построение канонических таблиц.

Оценим число v ребер дерева, которые будут фактически построены. Легко видеть, что эти ребра заполняют:

$$1) \text{ одно дерево высоты } s; \text{ в нем } m + m^2 + \dots + m^s = \\ = \frac{m^{s+1} - 1}{m - 1} - 1 \text{ ребер;}$$

*) m — число входных букв.

2) столько ярусов из m^s ребер, сколько вершин в базе.

Если вес $\theta \leq K$, то для числа v фактически построенных ребер имеем:

$$v \leq Km^s + \frac{m^{s+1}-1}{m-1} - 1. \quad (5.1)$$

В частности, когда $s = K - 1$ (оценка для $st(\theta)$), при заданной оценке веса получается:

$$v < Km^{K-1} + m^K, \quad (5.2)$$

что существенно меньше числа ребер в полном дереве высоты $2K - 1$, равного

$$m + m^2 + \dots + m^{2K-1} = \frac{m^{2K}-1}{m-1} - 1. \quad (5.3)$$

Пример. На рис. 5.11, а—5.11, в изображены этапы построения дерева для оператора, о котором известно,

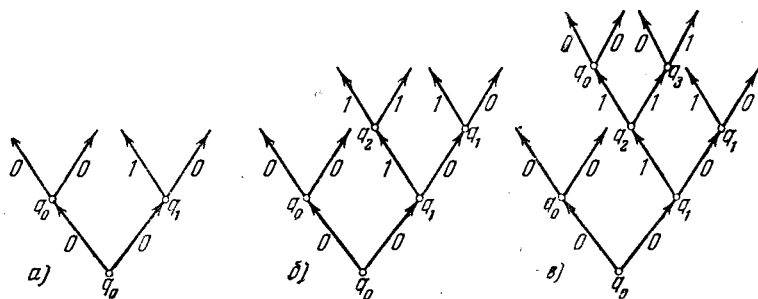


Рис. 5.11.

что его степень различимости равна 1. Построение завершается на пятом ярусе, когда никаких новых состояний не обнаружено. Соответствующая каноническая таблица дана на следующей странице (см. табл. V. 3).

Вес оператора θ равен 4; поэтому по предварительной оценке веса пришлось бы строить полное дерево высо-

входят и другие вершины, кроме вершин, ранее уже вошедших в базис \mathfrak{B} дерева V (все вершины из \mathfrak{B} входят в \mathfrak{B}_1 , так как в результате продолжения дерева V вершины, бывшие попарно различимыми, остаются таковыми). Пусть O_γ — такая вершина из \mathfrak{B}_1 , ранг которой γ является минимальным среди рангов новых вершин, вошедших в \mathfrak{B}_1 ;

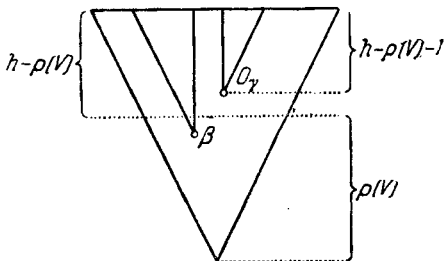


Рис. 5.13.

очевидно, что $\gamma \leq \rho(V) + 2$, ибо, по свойству базиса, если в \mathfrak{B}_1 содержится вершина ранга $\chi > \rho(V) + 2$, то в нем содержатся и вершины всех рангов, меньших чем χ , в том числе и вершины ранга $\rho(V) + 2$, не входящие в \mathfrak{B} .

В дереве V вершина O_γ была неразличима от какой-нибудь базисной вершины β , ранг которой меньше или равен $\rho(V) + 1$ (рис. 5.13). Поскольку вершины β и O_γ являются в V основаниями неразличимых ветвей высоты $\geq h - [\rho(V) + 1]$, а в V_∞ эти вершины уже различимы, то они в V_∞ строго λ -различимы при некотором $\lambda > h - \rho(V) - 1 = \text{ст}(V) + \delta(V)$. Итак, $\text{ст}(V_\infty) > \text{ст}(V) + \delta(V)$.

Рассмотрим теперь разбиение множества вершин дерева V_∞ на системы классов $\{\mathfrak{B}_j^p\}$ (где $p = 1, 2, \dots, \text{ст}(V_\infty)$). В системе $\{\mathfrak{B}_j^{\text{ст}(V)}\}$ число классов не меньше K (веса дерева V), ибо K базисных вершин V , будучи попарно $\text{ст}(V)$ -различимы, попадут в различные классы. Тогда, повторяя рассуждения, имевшие место при доказательстве теоремы из § 3 гл. IV, мы заключаем, что в системах $\{\mathfrak{B}_j^{\text{ст}(V)+1}\}$, $\{\mathfrak{B}_j^{\text{ст}(V)+2}\}$, ... число классов не меньше $K + 1$, $K + 2$ и т. д. И, наконец, в системе $\{\mathfrak{B}_j^{\text{ст}(V_\infty)}\}$ число классов больше чем

$K + \delta$. Следовательно, вес дерева V_∞ больше чем $K + \delta$. Теорема доказана.

Практическое значение этой теоремы заключается в следующем. Пусть для искомого оператора θ , вес которого по некоторой грубой оценке меньше чем \tilde{K} , построено конечное дерево V с высотой $h < 2\tilde{K} - 1$ и весом K (разумеется, $K \leq \tilde{K}$).

Вообще говоря, этого недостаточно для однозначного восстановления оператора. Однако если дерево имеет положительную избыточность δ , причем $K + \delta \geq \tilde{K}$, то, не привлекая дополнительных ярусов выше h , можно утверждать, что данное дерево высоты h однозначным образом определяет искомый оператор (который фактически совпадает с единственным оператором, продолжающим дерево V и имеющим вес K).

П р и м е р. Пусть для веса оператора θ имеется предварительная оценка: вес $K \leq 3$. В соответствии с теоремой 1 для восстановления канонических таблиц достаточно иметь пятиярусное дерево оператора.

Предположим теперь, что для θ нами уже построено четырехярусное дерево V , изображенное на рис. 5.14.

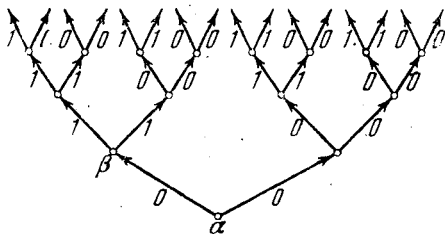


Рис. 5.14.

Вес K этого дерева равен 2; в качестве базиса можно взять вершины α и β . Очевидно, что $\text{ст}(V) = 1$ и $\text{q}(V) = 1$. Избыточность дерева тогда определяется так: $\delta(V) = h - \text{ст}(V) - \text{q}(V) - 1 = 4 - 3 = 1$.

Таким образом, в классе операторов с весом $\leq K + \delta = 3$ имеется единственное продолжение с весом, равным

в данном случае 2. Соответствующая каноническая таблица имеет вид:

Таблица V.4

$x \backslash q$	q_1	q_2
0	$0/q_2$	$1/q_2$
1	$0/q_1$	$1/q_0$

§ 3. Матрицы перехода

Наряду с каноническими таблицами (уравнениями) для задания ограниченно-детерминированных операторов удобно пользоваться таблицами другого рода — таблицами с двойным входом, получившими название *матриц перехода*.

В этом параграфе будет дано определение *матрицы перехода состояний* и *матрицы перехода реакций* и описаны два процесса:

- восстановление канонических уравнений по заданным матрицам перехода;
- вычисление элементов матриц перехода по заданным каноническим уравнениям.

Этот материал заимствован из работы М. Л. Цетлина*), который интерпретирует процесс а) как *синтез схемы автомата* и процесс б) как *анализ схемы автомата*. Такая интерпретация мотивирована тем, что матрицы перехода удобны для описания работы автомата во времени, в то время как канонические таблицы (уравнения) удобнее для построения по ним схемы автомата.

Матрицы перехода состояний имеют вид, показанный в таблице V.5.

В этой таблице X_{ij} есть подмножество входного алфавита X , состоящее в точности из тех входных букв X_λ ,

*) М. Л. Цетлин, О непримитивных схемах, Сб. «Проблемы кибернетики», № 2, 1959.

Т а б л и ц а V.5

$q(t+1) \backslash q(t)$	1	2	...	k
1	X_{11}	X_{12}	...	X_{1k}
2	X_{21}	X_{22}	...	X_{2k}
...
k	X_{k1}	X_{k2}	...	X_{kk}

для которых (ср. соответствующие канонические уравнения)

$$\Psi(X_{\lambda}, q_i) = q_j,$$

то есть из тех букв, которые переводят состояние q_i в состояние q_j .

П р и м е р. Для оператора θ (см. таблицу II.4) последовательного поразрядного суммирования матрица перехода состояний имеет вид, показанный в таблице V.6 ($q(t)$ обозначено через q , а $q(t+1)$ — через q').

Т а б л и ц а V.6

$q' \backslash q$	0	1
0	00, 01, 10	11
1	00	01, 10, 11

Очевидно, что диаграммы перехода, рассмотренные в гл. IV, являются графическими изображениями матриц перехода.

Из определения матрицы перехода состояний непосредственно вытекают (для любого $i \leq k$) следующие утверждения:

а) Множества $X_{i1}, X_{i2}, \dots, X_{ik}$ попарно не имеют общих элементов.

б) Объединение множеств X_{i1}, \dots, X_{ik} образует весь входной алфавит

$$\sum_{j=1}^k X_{ij} = X.$$

Совершенно аналогично можно определить и другие матрицы перехода, элементами которых являются различные подмножества алфавита состояний. Так, например, матрица перехода для выходных букв (сокращенно: *матрица реакций*) имеет вид

$$\|Q_{SL}\| \quad (S \leq m, L \leq k),$$

где Q_{SL} есть множество тех состояний q , для которых (см. канонические уравнения)

$$\Phi(x_S, q) = z_L.$$

Для множеств Q_{SL} имеют место соотношения типа а), б) (см. выше), где вместо входного алфавита X фигурирует алфавит состояний Q .

Иногда в матрицах перехода вместо множества X_{ij} или Q_{SL} мы будем брать (сохраняя при этом прежние обозначения X_{ij} , Q_{SL}) множества пар букв *), где первая буква пары имеет прежний смысл, а в качестве второй буквы взята соответствующая выходная буква (для матриц перехода состояний) или соответствующая буква состояний (для матриц перехода выходных букв).

Пример. Для того же оператора θ , рассмотренного в предыдущем примере, матрица, построенная указанным способом, имеет вид, показанный в таблице V.7.

Таблица V.7

$X \backslash z$	0	1
00	0/0	1/0
01	1/1	0/0
10	1/1	0/0
11	0/1	1/1

*) Эти пары мы будем записывать в виде дробей

канонических уравнений (5.4) в матрицу состояний. Именно, формулы для $X_{\eta_1, \dots, \eta_k, \eta'_1, \dots, \eta'_k}(x_1, \dots, x_m)$ можно составить исходя из тождества

$$\begin{aligned} X_{\eta_1, \dots, \eta_k, \eta'_1, \dots, \eta'_k}(x_1, \dots, x_m) &= \\ &= \Psi^{\eta_1}(x_1, \dots, x_m, \eta_1, \dots, \eta_k) \cdot \\ &\Psi_2^{\eta'_1}(x_1, \dots, x_m, \eta_1, \dots, \eta_k) \cdot \dots \\ &\dots \cdot \Psi_k^{\eta'_k}(x_1, \dots, x_m, \eta_1, \dots, \eta_k). \end{aligned} \quad (5.5)$$

Пример. Даны канонические уравнения:

$$\left. \begin{aligned} z(t) &= q_1(t) q_2(t) [x_1(t) \vee \bar{x}_2(t)], \\ q_1(t+1) &= [\bar{x}_1(t) \vee x_2(t) \vee \bar{q}_2(t)] \cdot \\ &\quad \cdot [q_1(t) \vee x_1(t) \bar{x}_2(t)] = \Psi_1, \\ q_2'(t+1) &= [x_1(t) \vee \bar{x}_2(t) \vee q_1(t)] \cdot \\ &\quad \cdot [q_2(t) \vee \bar{x}_1(t) x_2(t)] = \Psi_2. \end{aligned} \right\} \quad (5.6)$$

Восстановим по ним матрицу состояний. Найдем, например, матричный элемент $X_{00, 00}$:

$$\begin{aligned} X_{00, 00} &= [\bar{x}_1 \vee x_2 \vee 0] [0 \vee x_1 \bar{x}_2] \cdot \\ &\quad \cdot [x_1 \vee \bar{x}_2 \vee 0] [0 \vee \bar{x}_1 x_2] = \bar{x}_1 \vee x_2. \end{aligned}$$

Найдя аналогично остальные матричные элементы, мы получаем матрицу состояний, приведенную в таблице V.8.

Таблица V.8

	0 0	0 1	1 0	1 1
0 0	$\bar{x}_1 \vee x_2$	0	$x_1 \bar{x}_2$	0
0 1	$\bar{x}_1 x_2$	$x_1 \vee \bar{x}_2$	0	0
1 0	0	0	$x_1 \vee \bar{x}_2$	$\bar{x}_1 x_2$
1 1	0	$x_1 \bar{x}_2$	0	$\bar{x}_1 \vee x_2$

Аналогично обстоит дело с матрицей реакций, элементами которой объявляются характеристические функции множеств наборов $Q_{\alpha_1, \dots, \alpha_m, \beta_1, \dots, \beta_n}$, обозначаемые как $Q_{\alpha_1, \dots, \alpha_m, \beta_1, \dots, \beta_n} (q_1, \dots, q_k)$. Составление матрицы реакции по каноническим уравнениям производится в соответствии с формулами

$$\begin{aligned} Q_{\alpha_1, \dots, \alpha_m, \beta_1, \dots, \beta_n} (q_1, \dots, q_k) = \\ = \Phi_1^{\beta_1} (\alpha_1, \dots, \alpha_m, q_1, \dots, q_k) \& \dots \\ \dots \& \Phi_n^{\beta_n} (\alpha_1, \dots, \alpha_m, q_1, \dots, q_k). \end{aligned} \quad (5.7)$$

Пример. Пользуясь формулами (5.7), по уравнениям (5.6) восстанавливается следующая матрица реакций (табл. V.9).

Таблица V.9

	0	1
0 0	$q_1 \vee \bar{q}_2$	$\bar{q}_1 q_2$
0 1	$q_1 \vee \bar{q}_2$	$\bar{q}_1 q_2$
1 0	1	0
1 1	$q_1 \vee \bar{q}_2$	$\bar{q}_1 q_2$

Условия а) и б), которым удовлетворяют элементы матриц состояний, рассматриваемые как множества, для случая, когда элементами матрицы объявлены функции алгебры логики, должны быть перефразированы следующим образом:

а) Любые два элемента (две функции) в одной строке попарно ортогональны (то есть их конъюнкция обращается в тождественный нуль).

б) Дизъюнкция всех элементов одной строки тождественно равна 1.

Вместе с тем по всякой паре матриц, элементы которых являются функциями алгебры логики и для которых выполняются условия а) и б) (при соблюдении надлежащих условий относительно размерности матриц и числа аргументов), могут быть восстановлены соответствующие канонические уравнения. При этом по матрице реакций

восстанавливается первая группа уравнений (функции Φ_1, \dots, Φ_n), а по матрице состояний — вторая группа (функции Ψ_1, \dots, Ψ_k).

Можно было бы указать формулы, дающие явное выражение функций Φ_i и Ψ_j через матричные элементы, подобно тому как формулы (5.5) и (5.7) выражают матричные элементы через функции Φ_i и Ψ_j . Однако эти формулы слишком громоздки, и нам удобнее будет пользоваться словесной формулировкой соответствующего алгоритма, сопровождая ее иллюстрацией на конкретном примере.

Пусть, для определенности, нужно восстановить функции Ψ_i ($i=1, 2$) по матрице состояний, заданной таблицей V.8. Тогда мы поступаем следующим образом:

1) В матрице состояний отмечаем столбцы, в нумерации которых $q_i=1$ (для Ψ_1 — это третий и четвертый столбцы).

2) Для строки, занумерованной парой η_1, η_2 , составляем конъюнкцию $q_1^{\eta_1} q_2^{\eta_2}$ и дизъюнкцию $\mathcal{A}_{\eta_1 \eta_2}$ матричных элементов, лежащих на пересечении этой строки с отмеченными столбцами. В нашем случае (для Ψ_1) мы имеем:

$$\mathcal{A}_{00} = x_1 \bar{x}_2,$$

$$\mathcal{A}_{01} = 0,$$

$$\mathcal{A}_{10} = x_1 \vee \bar{x}_2 \vee \bar{x}_1 x_2 = 1,$$

$$\mathcal{A}_{11} = \bar{x}_1 \vee x_2.$$

3) Учитываем, что для искомой функции имеет место равенство:

$$\Psi_1 = \bar{q}_1 \bar{q}_2 \mathcal{A}_{00} \vee \bar{q}_1 q_2 \mathcal{A}_{01} \vee q_1 \bar{q}_2 \mathcal{A}_{10} \vee q_1 q_2 \mathcal{A}_{11}. \quad (5.8)$$

В нашем случае мы будем иметь:

$$\Psi_1 = \bar{q}_1 \bar{q}_2 x_1 \bar{x}_2 \vee \bar{q}_1 q_2 \vee q_1 q_2 (\bar{x}_1 \vee x_2).$$

Хотя по виду это выражение и не совпадает с формулой для $q_1(t+1)$ в уравнениях (5.6), однако легко показать, что посредством тождественных преобразований оно может быть приведено к этому же виду.

Для восстановления функции Φ_i по матрице реакции поступают аналогично: отмеченными в этом случае будут

этом как задание оператора, преобразующего систему входных предикатов $\{x_j(t)\}$ в систему выходных предикатов $\{z_i(t)\}$. Переход от канонических таблиц к уравнениям (5.10) сводится к обычному составлению формул алгебры логики по соответствующим таблицам истинности (см. главу I). Подчеркнем, что хотя в этих уравнениях и встречаются предикаты, привлекаемые логические операции относятся только к логике суждений.

Ранее (гл. I, § 5) уже было разъяснено, почему первоначальные предписания для преобразования информации зачастую возникают в терминах логики суждений (или в близких к ней терминах). Если предписания записаны на языке логики суждений, то тогда с самого начала мы имеем дело с системой логических уравнений, имеющих вид (5.10) (или легко приводимых к такому виду). Такая ситуация является особенно типичной для истинностных операторов. Рассмотрим несколько примеров, в которых (для большей простоты) мы ограничимся случаем единственного выходного предиката.

1°. Входными предикатами являются предикаты $x_1(t)$, $x_2(t)$, $x_3(t)$. Пусть z равно 1 в некоторый момент t в том и только в том случае, когда x_1 равно 0 и в то же время x_2 и x_3 принимают разные значения. Соответствующая алгебροлогическая запись имеет вид:

$$z(t) = \bar{x}_1(t) [x_2(t) \oplus x_3(t)].$$

2°. Входной предикат $x(t)$ единственный; при $t > 1$ справедливо, что $z(t) = 1$ в том и только в том случае, когда значение x в данный момент t отлично от его значения в непосредственно предшествующий момент. При $t = 1$ значение z задано дополнительно начальным условием $z(1) = w$. Тогда мы имеем:

$$z(t) = (t > 1) [x(t) \oplus x(t-1)] \vee (t = 1) w. \quad (5.11)$$

Уравнение (5.11) может быть приведено к каноническому виду следующим образом. Введем внутренний предикат

$$\left. \begin{aligned} q_1(t+1) &= x(t), \\ q_1(1) &= w \end{aligned} \right\} \quad (5.12)$$

(равенство $q_1(1)=u$ означает, что $q_1(1)$ принимает произвольное значение 0,1). При наличии этого предиката уравнение (5.11) принимает вид

$$z(t) = (t > 1) [x(t) \oplus q_1(t)] \vee (t = 1) \omega.$$

Нам остается еще элиминировать выражения $t > 1$ и $t = 1$. Для этого достаточно ввести внутренний предикат $q_2(t)$, имеющий смысл $t > 1$ и определяемый соотношениями

$$q_2(t+1) = 1,$$

$$q_2(1) = 0.$$

Окончательно вместо (5.11) получим следующую систему канонических уравнений (с начальными условиями):

$$z(t) = q_2(t) [x(t) \oplus q_1(t)] \vee \bar{q}_2(t) \omega,$$

$$q_1(t+1) = x(t),$$

$$q_2(t+1) = 1,$$

$$q_1(1) = u,$$

$$q_2(1) = 0.$$

Для составления соответствующей канонической таблицы (табл. V.10) достаточно подсчитать значения $z(t)$, $q_1(t+1)$, $q_2(t+1)$ для каждого из 2^3 возможных распределений значений $x(t)$, $q_1(t)$, $q_2(t)$ (например, при $x=q_1=q_2=0$ мы получаем: $z(t)=\omega$, $q_1(t+1)=0$, $q_2(t+1)=1$).

Таблица V.10

$x \backslash q_1 q_2$	00	01	10	11
0	$\omega/01$	$0/01$	$\omega/01$	$1/01$
1	$\omega/11$	$1/11$	$\omega/11$	$0/11$

Из таблицы V.10 видно, что вес оператора, рассматриваемого в данном примере, равен 3, ибо состояния 00 и 01 неразличимы, каково бы ни было начальное значение ω .

3°. Рассмотрим еще естественное обобщение предыдущего примера, заключающееся в том, что начальные условия предписаны для нескольких значений t .

Пусть дано:

$$z(t) = (t > 3) [x(t) \oplus x(t-1)] \vee (t=1) w_1 \vee \\ \vee (t=2) w_2 \vee (t=3) w_3, \quad (5.13)$$

где w_i — некоторые константы, или условия, выраженные через $x(t)$ при $t \leq i$ средствами логики высказываний. Тогда вместо одного предиката q_2 из предыдущего примера для распознавания времени $t=1, 2, 3$ или >3 можно воспользоваться двумя предикатами q_2, q_3 , определенными, например, так, как указано в таблице V.11.

Таблица V.11

t	$q_2(t)$	$q_3(t)$
1	0	1
2	1	0
3	1	1
4	0	0
5	0	0
6	0	0
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮

В соответствии с этой таблицей пара цифр $q_2(t), q_3(t)$ есть двоичная запись числа t при $t \leq 3$ и двоичная запись числа 0 при $t > 3$. Зависимость $q_2(t+1)$ и $q_3(t+1)$ от $q_2(t), q_3(t)$ указана в таблице V.12 и задана уравнениями (5.13').

Таблица V.12

$q_2(t)$	$q_3(t)$	$q_2(t+1)$	$q_3(t+1)$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	0

Уравнения, задающие зависимость значений $q_2(t+1)$ и $q_3(t+1)$ от значений $q_2(t)$ и $q_3(t)$, имеют вид

$$q_2(t+1) = q_2(t) \oplus q_3(t), \quad q_3(t+1) = q_2(t) \bar{q}_3(t) \quad (5.13')$$

при начальных условиях

$$q_2(1) = 0, \quad q_3(1) \stackrel{\Delta}{=} 1.$$

Канонические уравнения, соответствующие уравнениям (5.13), таковы

$$\begin{aligned} z(t) = & \bar{q}_2(t) \bar{q}_3(t) [x(t) \oplus q_1(t)] \vee \bar{q}_2(t) q_3(t) \omega_1 \vee \\ & \vee q_2(t) \bar{q}_3(t) \omega_2 \vee q_2(t) q_3(t) \omega_3, \\ q_1(t+1) = & x(t), \\ q_2(t+1) = & q_2(t) \oplus q_3(t), \\ q_3(t+1) = & q_2(t) \bar{q}_3(t) \end{aligned}$$

при начальных условиях $q_1(1) = \omega^*$, $q_2(1) = 0$ и $q_3(1) = 1$.

Из рассмотренных примеров ясно, как осуществляется переход к каноническим уравнениям любого оператора, первоначально заданного уравнениями типа

$$z(t) = \Phi[x_1(t), x_1(t-1), x_1(t-2), \dots, x_m(t), x_m(t-1), \dots]$$

для $t > v$, и начальными условиями для $z(t)$ при $t = 1, 2, \dots, v$ (где v — некоторое фиксированное число). Общее число внутренних предикатов, вводимых для выражения условий $t=1, t=2, \dots, t=v, t>v$, не превосходит $]g_2(v+1)[^{**})$. Кроме того, если какой-нибудь входной предикат x_i встречается с аргументом $t-s$, то следует ввести s внутренних предикатов:

$$\begin{aligned} q_1(t+1) &= x(t), \\ q_2(t+1) &= q_1(t), \\ &\dots \dots \dots \\ q_s(t+1) &= q_{s-1}(t). \end{aligned}$$

П р и м е ч а н и е. Начальные условия для всех указанных внутренних предикатов можно считать нулевыми, что следует из того, что:

*) ω — произвольная константа (0 или 1).

**) $]a[$ означает наименьшее из таких целых чисел m , для которых $a \leq m$.

1) для предиката типа $q(t+1)=x(t)$ начальное условие можно выбрать любым;

2) предикаты, распознающие времена $t=1, 2, \dots, v$, можно определить так, чтобы счет времени начинался не с 1 (как это имело место в таблице V.9), а с нуля.

4°. Процедура, описанная выше, применима и в том случае, когда оператор задан уравнением вида

$$z(t) = \Phi[x_1(t), x_1(t-1), \dots, z(t-1), z(t-2), \dots] \quad (5.14)$$

при соответствующих начальных условиях. Для этого достаточно ввести наряду с внутренними предикатами $q_1(t+1)=x(t)$, $q_2(t+1)=q_1(t)$, ... внутренние предикаты $p_1(t+1)=z(t)$, $p_2(t+1)=p_1(t)$.

Тогда уравнение (5.14) перейдет в систему уравнений

$$z(t) = \Phi[x_1(t), q_1(t), q_2(t), \dots, p_1(t), p_2(t), \dots],$$

$$q_1(t+1) = x(t),$$

$$q_2(t+1) = q_1(t),$$

$$\dots \dots \dots$$

$$p_1(t+1) = \Phi[x_1(t), q_1(t), q_2(t), \dots, p_1(t), p_2(t), \dots],$$

$$p_2(t+1) = p_1(t),$$

$$\dots \dots \dots$$

с соответствующими начальными условиями.

Так, например, работа двоичного счетчика описывается условием

$$z(t) = [z(t-1) \oplus x(t)] (t > 1) \vee x(t) (t = 1).$$

Положив $p_1(t+1)=z(t)$ и $p_1(1)=0$, мы убеждаемся, что $p_1(1) \oplus x(1) = x(1)$. Следовательно, канонические уравнения можно написать в виде

$$z(t) = p_1(t) \oplus x(t),$$

$$p_1(t+1) = p_1(t) \oplus x(t)$$

при начальном условии $p_1(1)=0$.

Во всех изложенных выше случаях переход от первоначальных описаний осуществленных средствами логики суждений к каноническим уравнениям возможен и происходит очень просто. Однако это вовсе не означает, что такой переход всегда необходим при анализе или синтезе

схемы автомата. Уравнения указанных типов бывают удобными для непосредственного построения по ним соответствующих логических сетей. Наличие предикатов, зависящих от аргументов $t-1$, $t-2$ и т. д., указывает на то, что следует применять надлежащие задерживающие элементы; в частности, присутствие в уравнении (5.14) выражений $z(t-1)$, $z(t-2)$, ... приводит к образованию замкнутых петель обратной связи через задерживающие элементы. Все эти вопросы будут более подробно рассмотрены нами в главе VI.

До сих пор мы рассматривали такие случаи, когда первоначальное предписание для выходного предиката $z(t)$ сформулировано посредством операций логики суждений: конъюнкции, дизъюнкции, отрицания и т. д. Язык логики предикатов (то есть применение к в а н т о р о в) расширяет возможности формулировки таких предписаний. Прежде чем перейти к общим рассмотрениям, приведем примеры, показывающие, что формулировка первоначального предписания с привлечением кванторов естественно напрашивается во многих случаях.

П р и м е р *). Автоматическая железнодорожная переездная сигнализация функционирует следующим образом. При вступлении поезда на участок переезда AB в направлении от A к B шлагбаум закрывается на весь период времени $\tau_1 \leq \sigma < \tau_2$, длящийся до тех пор, пока колеса поезда касаются рельса в точке A . Считается, что длина поезда \geq длине участка AB , и тогда для некоторого q между τ_1 и τ_2 колеса касаются одновременно рельса как в A , так и в B . Касание рельса в A или в B закодировано как подача нуля по входному каналу A (соответственно по входному каналу B) системы сигнализации, а закрытие шлагбаума — как выдача единицы по выходному каналу z . Шлагбаум закрывается при выполнении условий, описываемых в нижеследующих пунктах:

1) всегда, когда одновременно $A(t)=0$ и $B(t)=0$, т. е. поезд, вступивший на участок AB , занял его полностью. Это условие записывается так: $\overline{A}(t) \cdot \overline{B}(t)$;

*) Пример заимствован из книги М. А. Гаврилова «Теория релейно-контактных схем», где содержится другой способ решения этой задачи.

В правой части этого равенства дизъюнктивный член $(\exists \tau) x(t)$ задает как раз предикат $z(t)$; следовательно, $\tau < 1$

$$z(t+1) = z(t) \vee x(t).$$

Для $t=1$ непосредственно из формулы (5.16) имеем:

$$z(1) = (\exists \tau) x(\tau) = 0$$

(ибо для параметра τ не существует значений, которые были бы меньше 1).

Тем самым для оператора, описываемого формулой (5.16), получаются уже почти канонические уравнения

$$\begin{aligned} z(t+1) &= z(t) \vee x(t), \\ z(1) &= 0. \end{aligned}$$

Переход от этих уравнений к каноническим путем введения внутренних предикатов может быть осуществлен (в случае надобности) так, как это было показано раньше. В результате мы получим систему

$$\begin{aligned} z(t) &= q_1(t) [q_2(t) \vee q_3(t)], \\ q_1(t+1) &= 1, \\ q_2(t+1) &= q_1(t). \end{aligned}$$

Еще проще, положив $z(t) = q(t)$, получаются уравнения

$$\begin{aligned} z(t) &= q(t), \\ q(t+1) &= q(t) \vee x(t) \end{aligned}$$

при начальном условии $q(1)=0$.

Совершенно аналогично осуществляется переход от формулы типа

$$z(t) = (\tau) x(\tau) \quad (5.17)$$

к уравнениям

$$\left. \begin{aligned} z(t+1) &= z(t) \cdot x(t), \\ z(1) &= 1, \end{aligned} \right\} \quad (5.17')$$

а затем к системе канонических уравнений

$$\left. \begin{aligned} z(t) &= q(t), \\ q(t+1) &= x(t) q(t) \end{aligned} \right\} \quad (5.17'')$$

при начальном условии $q(1)=1$.

§ 5. Язык логики предикатов

1) Спуск кванторов

В этом параграфе будет дано описание формального языка, построенного на базе исчисления одноместных предикатов с ограниченными кванторами (см. гл. I, § 5), посредством которого можно определять ограниченно-детерминированные операторы. Сначала проследим, как преобразуется выражение

$$z(t) = (\tau) [x_1(\tau) \vee (\mathbf{E} q) x_2(q)] \quad (5.18)$$

$\tau < t$ $\tau < q < t$

при спуске кванторов; одновременно эта процедура будет обоснована применительно ко всем формулам рассматриваемого языка. Правую часть соотношения (5.18), определяющую предикат $z(t)$, мы обозначим через \mathfrak{A} . Отметим сначала следующие особенности выражения \mathfrak{A} :

1. \mathfrak{A} есть формула исчисления одноместных предикатов, все предметные кванторы которой ограничены.

2. В формуле \mathfrak{A} имеется единственная свободная переменная, а именно t .

Формулу \mathfrak{A} , характеризруемую свойствами 1 и 2, мы будем называть *t-формулой* и обозначать через $\mathfrak{A}(t)$ (эта запись содержит указание на свободную предметную переменную t). Иногда мы будем эту формулу обозначать еще и так: $\mathfrak{A}(x_1, x_2, \dots, x_m, t)$; такая запись будет служить для указания, помимо переменной t , свободных предикатных переменных, которые в ней участвуют. Очевидно, что в (5.18) таких предикатных переменных две: x_1 и x_2 , а в (5.17) содержится единственная предикатная переменная x .

При любом фиксированном выборе предикатов x_1, x_2, \dots, x_m формула \mathfrak{A} задает вполне определенный предикат $z(t)$; иными словами, $\mathfrak{A}(x_1, x_2, \dots, x_m, t)$ задает оператор, перерабатывающий систему входных предикатов x_1, x_2, \dots, x_m в выходной предикат $z(t)$. Рассмотрим теперь связанные предметные переменные τ и q , фигурирующие в (5.18). Характер ограничения кванторов здесь таков, что при любой фиксации значения t истинность или ложность формулы зависит лишь от того, каковы значения x_1, x_2 при значениях аргумента, не превосходящих t .

Содержательно это имеет тот смысл, что значение выходного предиката $z(t)$ в момент t зависит только от значений

входных предикатов в предыдущие моменты; тем самым гарантируется детерминированность оператора, описываемого формулой $\mathcal{A}(t)$.

Введем теперь понятие *t-контролируемости*, которое позволяет, в частности, усмотреть аналогичную ситуацию и в случае произвольной *t*-формулы.

Определение. Предметная переменная τ является *t-контролируемой* (соответственно, *строго t-контролируемой*) в подформуле \mathcal{B} формулы \mathcal{A} , если формула \mathcal{A} переходит в эквивалентную ей формулу при замене в \mathcal{B} любого вхождения неравенства $\tau \leq t$ (соответственно, $\tau < t$) на \mathcal{I} или при замене \mathcal{I} на $\tau \leq t$ (соответственно, на $\tau < t$).

Примечание. Строгая *t-контролируемость* влечет за собой и просто *t-контролируемость*.

Далее, справедлива следующая лемма.

Лемма. Во всякой подформуле вида $\underset{\lambda < \tau < \mu}{[\tau]} \mathcal{B}$ *t-формулы* $\mathcal{A}(t)$ предметная переменная τ является *t-контролируемой* в области действия квантора $[\tau]$, т. е. в формуле \mathcal{B} *).

Доказательство этой леммы основано на некоторых эквивалентностях, справедливых вообще для логики предикатов. Пусть $\mathcal{B}(\dots \mathcal{M} \dots)$ — формула, в которой выделена подформула \mathcal{M} ; результат замены этой подформулы на \mathcal{I} (соответственно, на \mathcal{L}) обозначается в виде $\mathcal{B}(\dots \mathcal{I} \dots)$ (соответственно, в виде $\mathcal{B}(\dots \mathcal{L} \dots)$). Как нетрудно показать, справедливы следующие эквивалентности**), которые мы будем называть *эквивалентностями с заменой*:

$$\mathcal{M} \cdot \mathcal{B}(\dots \mathcal{M} \dots) \text{ экв. } \mathcal{M} \cdot \mathcal{B}(\dots \mathcal{I} \dots), \quad (a)$$

$$\overline{\mathcal{M}} \cdot \mathcal{B}(\dots \mathcal{M} \dots) \text{ экв. } \overline{\mathcal{M}} \cdot \mathcal{B}(\dots \mathcal{L} \dots), \quad (б)$$

$$\mathcal{M} \rightarrow \mathcal{B}(\dots \mathcal{M} \dots) \text{ экв. } \mathcal{M} \rightarrow \mathcal{B}(\dots \mathcal{I} \dots), \quad (в)$$

$$\overline{\mathcal{M}} \rightarrow \mathcal{B}(\dots \mathcal{M} \dots) \text{ экв. } \overline{\mathcal{M}} \rightarrow \mathcal{B}(\dots \mathcal{L} \dots). \quad (г)$$

Проверим, например, эквивалентность (a). Если при некоторой интерпретации подформула \mathcal{M} ложна, то лож-

*) Запись $[\tau]$ служит для выражения того, что речь идет о кванторе любого из двух видов: кванторе общности (τ) или кванторе существования ($\exists \tau$). О значении знака « \prec » см. стр. 81.

**) Некоторыми из этих эквивалентностей мы будем пользоваться и в дальнейшем, вне данной леммы.

ными становятся и обе части эквивалентности (а); если же подформула \mathfrak{M} истинна, то каждая из частей эквивалентности имеет то же значение истинности, что и $\mathfrak{B}(\dots II\dots)$.

Из (а) и (в) непосредственно следует t -контролируемость (строгая t -контролируемость) переменной τ в \mathfrak{B} в подформулах вида $[\tau] \mathfrak{B}$ (соответственно, в подформулах вида $[\tau] \mathfrak{B}$). В самом деле, взяв, для определенности, квантор существования: $(\exists \tau) \mathfrak{B}$, можно это выражение записать как $(\exists \tau) (\lambda < \tau \& \tau < t \& \mathfrak{B})$ и применять (а) для $\tau < t$, взятого в качестве \mathfrak{M} . Аналогично рассматриваются и другие возможные случаи.

Пусть теперь в t -формуле $\mathfrak{A}(t)$ выделен квантор $[\tau] \mathfrak{B}$, где τ_1 отлично от t ; тогда, очевидно, существует квантор старшего ранга $[\tau_1]$. Если τ_2 тоже отлично от t , то процесс продолжается, выделяя кванторы $[\tau_2], [\tau_3], \dots$ до обязательного обнаружения квантора вида $[\tau_k]$ (назовем его ведущим для исходного квантора). Нетрудно теперь показать, что если в цепочке неравенств $\tau < \tau_1 < \tau_2 < \dots < t$ знак « $<$ » хотя бы один раз означает строгое неравенство, то переменная τ является строго контролируемой в подформуле $[\tau] \mathfrak{B}$; в противном случае можно гарантировать лишь контролируемость этой переменной. Лемма, таким образом, доказана.

Примечание. Если ведущий квантор имеет вид $[\tau_k]$, то тем самым уже можно гарантировать строгую t -контролируемость.

Проследим теперь за процессом спуска квантора, применяемым к t -формуле (5.18).

Для этого заметим, что в правой части*) равенства

$$z(t+1) = (\tau) [x_1(\tau) \vee (\exists q) x_2(q)];$$

кванторы, ограниченные сверху условием $\leq t$, встречаются

*) Для которой мы сохраним обозначение $\mathfrak{A}(t)$ вместо напрашивающегося обозначения $\mathfrak{A}(t+1)$.

дважды (в формуле (5.16) имелся только один такой квантор). Спуск кванторов мы начинаем производить с кванторов наивысшего ранга; в данном случае таковым является квантор (τ) , имеющий ранг 2, в то время как $(E q)$ имеет ранг 1. Мы получаем: *)

$$z(t+1) = (\tau)_{\tau < t} [x_1(\tau) \vee (E q)_{\tau < q \leq t} x_2(q)] \& [x_1(t) \vee (E q)_{t < q \leq t} x_2(q)]. \quad (5.18')$$

Вообще, если в формуле $\mathfrak{A}(t)$ имеются несколько кванторов наивысшего ранга R типа $[\omega]$, то области их действия попарно не пересекаются; поэтому одновременно и повсеместно можно применять эквивалентности 3'—4' (см. стр. 83) для элиминации ограничения $\omega \leq t$. Так, например, допустима замена $(E \omega) K$ на

$$(E \omega) K \vee q < t \cdot K_{\omega}^t, \quad (5.19)$$

где K_{ω}^t получается из K заменой ω на t . На самом же деле в наших условиях переменная q является строго t -контролируемой в области действия квантора $[q]$, который ее связывает, ибо квантор, являющийся ведущим для $[q]$ (обозначим его $[\sigma]$), имеет более высокий ранг, чем $(E \omega)$, и в нем ограничение $\sigma \leq t$ уже элиминировано (см. примечание к сформулированной выше лемме). Поэтому, вместо замены на (5.19), выражение $(E \omega) K$ можно заменить на

$$(E \omega) K \vee K_{\omega}^t, \quad (5.20)$$

и, аналогично, выражение $(\omega) K$ можно заменить на

$(\omega) K \cdot K_{\omega}^t$. В результате такого преобразования формула $\mathfrak{A}(t)$ перейдет в формулу $\mathfrak{A}'(t)$, причем:

1. $\mathfrak{A}'(t)$ есть t -формула, эквивалентная $\mathfrak{A}(t)$.
2. В формуле $\mathfrak{A}'(t)$ наивысший ранг кванторов вида $[\omega]$ меньше R .

3. Всякий предметный или предикатный символ, встречающийся в $\mathfrak{A}'(t)$, имеется также и в $\mathfrak{A}(t)$, причем его вхож-

*) Применяя к правой части уравнения (5.18) эквивалентность 3, приведенную на стр. 83.

дения в $\mathfrak{A}'(t)$ и $\mathfrak{A}(t)$ одновременно либо свободные, либо связанные; кроме того, кванторная глубина формулы $\mathfrak{A}'(t)$ не превосходит кванторной глубины формулы $\mathfrak{A}(t)$ (в предыдущем примере кванторные глубины были одинаковы).

Условимся называть *спутником* формулы \mathfrak{A} всякую формулу K , которая характеризуется условиями, высказанными в пункте 3 относительно \mathfrak{A}' . Итак, рассматриваемое нами преобразование переводит заданную t -формулу \mathfrak{A} с ограниченными кванторами в t -формулу \mathfrak{A}' , являющуюся ее спутником.

Применяя теперь к $\mathfrak{A}'(t)$ такие же преобразования, какие были ранее применены к формуле $\mathfrak{A}(t)$, мы получаем формулу $\mathfrak{A}''(t)$, в которой наивысший ранг кванторов типа $[\omega]$ не больше чем $R - 2$ и т. д. Очевидно, что после конечного числа шагов такого рода будет получена формула $\mathfrak{A}'''(t)$, для которой, кроме утверждений 1 — 3, справедливо еще следующее утверждение:

4. $\mathfrak{A}'''(t)$ не содержит кванторов типа $[\omega]$.

Без ограничения общности можно также считать выполненным условие:

5. В $\mathfrak{A}'''(t)$ не содержатся кванторы вида $[\tau]$ или $[\tau]$ (т. е. t может фигурировать только в верхних ограничениях кванторов).

Покажем как элиминируются кванторы данного вида.

а) Поскольку после спуска кванторов все связанные переменные являются строго t -контролируемыми, допустима замена выражения $(\tau) \mathfrak{B}$ на \mathcal{I} и выражения $(E\tau) \mathfrak{B}$ на \mathcal{L} . Это непосредственно вытекает из соотношений:

$$(\tau) \mathfrak{B} \text{ экв. } (\tau) \mathfrak{B} \cdot \mathcal{I} \text{ экв. } (*). \quad \text{экв. } (E\tau) \mathfrak{B} \ \& \ \omega < t \text{ экв. } \\ t < \tau < \omega \quad t < \tau < \omega \quad t < \tau < \omega$$

$$\text{экв. } (E\tau)(t < \tau \ \& \ \tau < \omega \ \& \ \mathfrak{B}) \ \& \ \omega < t \text{ экв. } \mathcal{L}.$$

Аналогично обстоит дело в случае квантора общности.

б) $[\tau] \mathfrak{B}$ очевидным образом эквивалентно \mathfrak{B}_t^t . В прочих случаях $[\tau] \mathfrak{B}$ заменяется на \mathcal{I} или \mathcal{L} в соответ-

*) По лемме, доказанной на стр. 255—256.

ствии с 1', 2' (стр. 83). Так, например, в (5.18') элиминируется $(E\varrho)x_2(\varrho)$, в результате чего получается

$$z(t+1) = (\tau)_{\tau < t} (x_1(\tau) \vee (E\varrho)_{\tau < \varrho < t} (x_2(\varrho))) \cdot x_1(t). \quad (5.18'')$$

Получением формулы $\tilde{\mathfrak{A}}(t)$ и завершается процесс спуска кванторов применительно к $\mathfrak{A}(t)$. В нашем конкретном случае процедура спуска кванторов завершается путем спуска квантора первого ранга $(E\varrho)$:

$$z(t+1) = (\tau)_{\tau < t} [x_1(\tau) \vee (E\varrho)_{\tau < \varrho < t} x_2(\varrho) \vee x_2(t)] \cdot x_1(t). \quad (5.18''')$$

2) Разложение по регулярным спутникам

Перейдем теперь к так называемому разложению по регулярным спутникам.

Сравнение формул $\mathfrak{A}(t)$ и $\tilde{\mathfrak{A}}(t)$ в нашем конкретном случае (то есть формул (5.18) и (5.18''')) обнаруживает следующую особенность первой из них, нарушенную во второй:

6. В предикатах формулы $\mathfrak{A}(t)$ ни одно аргументное место не замещено свободной переменной t .

Условимся называть *регулярной* t -формулу с ограниченными кванторами, обладающую свойствами 4, 5 и 6. Содержательно регулярность означает, что при интерпретации формулы ее истинность не зависит от того, как определены предикаты для значения t (и для значений больших t).

В формуле (5.18''') имеется две «иррегулярности»: $x_1(t)$ и $x_2(t)$, причем первая из них входит в область действия квантора (τ) . Легко видеть, что правило выноса

относительной константы за знаки кванторов позволяет преобразовать данную формулу (5.18''') в такую формулу (5.21), которая построена посредством операций логики суждений, примененных к ее регулярным спутникам и иррегулярностям $x_1(t)$, $x_2(t)$:

$$z(t+1) = \{ (\tau)_{\tau < t} [x_1(\tau) \vee (E\varrho)_{\tau < \varrho < t} x_2(\varrho)] \vee x_2(t) \} \cdot x_1(t). \quad (5.21)$$

Такое преобразование мы будем называть *разложением формулы по регулярным спутникам*.

В нашем конкретном случае единственный регулярный спутник, фигурирующий в этом разложении, задает как раз $z(t)$ (ср. с формулой (5.18)); поэтому справедливо:

$$z(t+1) = [z(t) \vee x_2(t)] \cdot x_1(t). \quad (5.22)$$

Кроме того, по формуле (5.18) непосредственно определяется начальное значение

$$z(1) = (\tau) [x_1(\tau) \vee (E\varrho) x_2(\varrho)],$$

$\tau < 1 \qquad \tau < \varrho < 1$

которое равно 1 (см. гл. I, § 5).

Переход от (5.22) и начальных условий к каноническим уравнениям осуществляется далее обычным способом.

Таким образом, установлено, что оператор, задаваемый t -формулой (5.18) без предикатных кванторов, является ограниченно-детерминированным оператором. Более того, примененная нами процедура позволяет получить канонические уравнения этого оператора. Мы покажем сначала, что так обстоит дело всегда, когда оператор задан произвольной t -формулой без предикатных кванторов *).

Заметим прежде всего, что процедура выноса иррегулярностей за знак кванторов применима ко всякой такой t -формуле \mathfrak{A} и приводит ее к виду

$$\Phi[x_1(t), \dots, x_m(t), \mathfrak{E}_1(t), \dots, \mathfrak{E}_v(t)], \quad (5.23)$$

где Φ — функция алгебры логики, x_i — входные предикаты, \mathfrak{E}_i — некоторые регулярные спутники формулы \mathfrak{A} . Чтобы убедиться в справедливости этого утверждения, достаточно вынести иррегулярности сначала за знак кванторов первого ранга, потом за знак кванторов второго ранга и т. д. Предварительно приходится преобразовать подкванторное выражение к виду, удобному для выноса относительной константы (см. гл. I, § 5); например, в случае квантора общности (τ) подкванторное выражение можно представить в виде конъюнкции дизъюнкций

*) Позднее это ограничение будет снято.

и придать формуле вид

$$(\tau) \{[\Gamma_1 \vee U_1(t)] \& [\Gamma_2 \vee U_2(t)] \& \dots\},$$

где Γ_i не содержит иррегулярностей, а $U(t)$ не содержит связанной переменной τ . Тогда, распределяя квантор (τ) между членами конъюнкции, составляющей его подкванторную область, мы получаем конъюнкцию выражений вида $(\tau) [\Gamma_i \vee U_i(t)]$, в каждом из которых относительная константа уже выносится. Аналогично обстоит дело в случае квантора существования.

Пример. Разложить по регулярным спутникам формулу:

$$\mathfrak{A}(t) = (\exists \tau) \{x_1(t) \cdot (\exists q) [x_2(q) \vee \bar{x}_1(q) \cdot x_2(t)]\}.$$

Поскольку предикат $x_1(t)$ сразу выносится за знак квантора $(\exists \tau)$, займемся выносом предиката $x_2(t)$ сначала за знак квантора $(\exists q)$, а потом и за знак квантора $(\exists \tau)$.

Преобразуем область действия квантора $(\exists q)$ в соответствии с эквивалентностью:

$$A \vee B \cdot C \text{ экв. } (A \vee B) \cdot (A \vee C).$$

Эквивалентности с кванторами позволяют вынести $x_2(t)$ из-под знака квантора $(\exists q)$. Отдельные этапы этого процесса выглядят так:

$$\begin{aligned} & (\exists q) \{[x_2(q) \vee \bar{x}_1(q)] \& [x_2(q) \vee x_2(t)]\}, \\ & (\exists q) [x_2(q) \vee \bar{x}_1(q)] \& (\exists q) [x_2(q) \vee x_2(t)], \\ & (\exists q) [x_2(q) \vee \bar{x}_1(q)] \& [(\exists q) x_2(q) \vee x_2(t)]. \end{aligned}$$

Для выноса предиката $x_2(t)$ из-под знака квантора $(\exists \tau)$ можно представить полученное выражение как дизъюнкцию конъюнкций; что возможно на основе эквивалентности

$$A \& (B \vee C) \text{ экв. } A \& B \vee A \& C.$$

Мы получим:

$$(\exists q) [x_2(q) \vee \bar{x}_1(q)] \cdot (\exists q) x_2(q) \vee (\exists q) [x_2(q) \vee \bar{x}_1(q)] \cdot x_2(t).$$

Легко видеть, что первый дизъюнктивный член эквивалентен выражению $(\varrho) x_2(\varrho)$; для этого достаточно конъюнкцию кванторов

$$(\varrho) [x_2(\varrho) \vee \bar{x}_1(\varrho)] \cdot (\varrho) x_2(\varrho)$$

заменить квантором (ϱ) по конъюнкции $[x_2(\varrho) \vee \bar{x}_1(\varrho)] \cdot x_2(\varrho)$ и воспользоваться эквивалентностью логики высказываний

$$A \& (A \vee B) \text{ экв. } A.$$

Поэтому мы имеем:

$$\mathfrak{A}(t) = x_1(t) \cdot (\mathbf{E}\tau) \{ (\varrho) x_2(\varrho) \vee (\varrho) [x_2(\varrho) \vee \bar{x}_1(\varrho)] \cdot x_2(t) \},$$

$$\mathfrak{A}(t) = x_1(t) \cdot (\mathbf{E}\tau) (\varrho) x_2(\varrho) \vee x_2(t) \cdot (\mathbf{E}\tau) (\varrho) (x_2(\varrho) \vee \bar{x}_1(\varrho)).$$

Здесь фигурируют два регулярных спутника исходной формулы, а именно:

$$(\mathbf{E}\tau) (\varrho) x_2(\varrho)$$

■

$$(\mathbf{E}\tau) (\varrho) (x_2(\varrho) \vee \bar{x}_1(\varrho)).$$

Другой прием «выноса» иррегулярностей основан на следующих соображениях. Пусть $\mathfrak{A}[\dots, x(t), \dots]$ — произвольная формула логики предикатов (быть может, и с кванторами по предикатам, в которой имеются свободные*) вхождения $x(t)$. Обозначим через $\mathfrak{A}[\dots, I, \dots]$ и $\mathfrak{A}[\dots, L, \dots]$ формулы, получаемые из \mathfrak{A} путем замены каждого вхождения $x(t)$ на I и, соответственно, на L . Нетрудно усмотреть, что из эквивалентностей с заменой (см. лемму на стр. 255) вытекает следствие:

С л е д с т в и е.

$$\mathfrak{A}[\dots, x(t), \dots] \text{ экв. } x(t) \cdot \mathfrak{A}[\dots, I, \dots] \vee \bar{x}(t) \cdot \mathfrak{A}[\dots, L, \dots].$$

*) То есть такие вхождения, в которых как предикатная переменная x , так и предметная переменная t являются свободными.

Это соотношение вполне аналогично разложению функции алгебры логики по одной из переменных (см. гл. I) и допускает обобщение, аналогичное разложению функции по нескольким переменным. Отсюда непосредственно и вытекает способ «выноса» иррегулярных вхождений свободных предикатов. Проиллюстрируем этот метод применительно к ранее уже рассмотренной формуле из примера, разбиравшегося нами на стр. 261—262.

Обозначив через $\mathfrak{A}[x_1(t), x_2(t)]$ эту формулу, мы будем иметь:

$$\mathfrak{A}[I, I] \text{ экв. } (\exists \tau) (q) (x_2(q) \vee \bar{x}_1(q)),$$

$$\tau < t \quad \tau < q < t$$

$$\mathfrak{A}(I, L) \text{ экв. } (\exists \tau) (q) x_2(q),$$

$$\tau < t \quad \tau < q < t$$

$$\mathfrak{A}(L, I) \text{ экв. } L,$$

$$\mathfrak{A}(L, L) \text{ экв. } L.$$

Следовательно,

$$\mathfrak{A} = x_1(t) \cdot x_2(t) \cdot (\exists \tau) (q) (x_2(q) \vee \bar{x}_1(q)) \cdot \vee$$

$$\tau < t \quad \tau < q < t$$

$$\vee x_1(t) \cdot \bar{x}_2(t) (\exists \tau) (q) x_2(q).$$

$$\tau < t \quad \tau < q < t$$

Легко видеть, что этот результат эквивалентен результату, полученному нами ранее.

3) Описание алгоритма

Рассмотрим теперь произвольный оператор

$$z(t) = \mathfrak{A}[x_1, \dots, x_m, t],$$

где \mathfrak{A} есть t -формула без предикатных кванторов.

Опишем работу алгоритма, дающего в результате своего применения канонические уравнения этого оператора.

Предварительно заметим следующее *). Какова бы ни была формула \mathfrak{A} , множество ее спутников бесконечно; это

*) Это замечание относится и к формулам с предикатными кванторами.

видно хотя бы из того, что оно содержит все формулы вида \mathcal{A} , $\mathcal{A} \& \mathcal{A}$, $\mathcal{A} \& \mathcal{A} \& \mathcal{A}$, ... Однако путем тождественных преобразований любой из спутников формулы \mathcal{A} приводится к некоторому стандартному виду, причем заведомо существует лишь конечное число «стандартных» — в принятом смысле — спутников. Возможна, например, такая стандартизация: пусть \mathfrak{B} — произвольная формула; каков бы ни был в ней квантор ранга ν , его область действия D является функцией алгебры логики $\Phi[\Gamma_1, \dots, \Gamma_s]$ от некоторых подформул $\Gamma_1, \dots, \Gamma_s$ ранга, меньшего чем ν . В частности, Φ могла оказаться в дизъюнктивной (или конъюнктивной) нормальной форме. Объявим стандартными те формулы, в которых область действия любого квантора приведена либо к дизъюнктивной, либо к конъюнктивной нормальной форме.

У т в е р ж д е н и е 1. Всякую формулу можно эффективно привести к стандартному виду. Это утверждение очевидно.

У т в е р ж д е н и е 2. При фиксированном конечном запасе предикатных и предметных символов существует лишь конечное число стандартных формул.

Из этих утверждений непосредственно вытекает конечность числа стандартных спутников для любой заданной формулы.

Вернемся теперь к нашему алгоритму. Алгоритм складывается из операций двух типов:

А. Спуск квантора.

Б. Разложение по регулярным спутникам.

П р и м е ч а н и е. Без ограничения общности можно считать в дальнейшем, что спутники имеют стандартную форму.

Алгоритм реализуется следующим образом:

1. Если \mathcal{A} не является регулярной формулой, то применение операции А (если имеются кванторы вида $\left[\omega \right]_{\substack{q < \omega < t}}$), а затем операции Б (если обнаружены иррегулярности вида $x(t)$) дает:

$$z(t) = \Phi[x_1(t), \dots, x_m(t), \mathfrak{B}_1(t), \dots, \mathfrak{B}_s(t)].$$

2. Замещая в каждом из регулярных спутников $\mathfrak{B}_i(t)$ всюду в кванторах $\left[\omega \right]_{\substack{q < \omega < t}}$ нестрогое неравенство $\omega < t$ на

Теорема 3. *Всякая t -формула $\mathcal{A}(x_1, x_2, \dots, x_m, t)$ без предикатных кванторов задает ограниченно-детерминированный оператор, преобразующий систему входных предикатов x_1, x_2, \dots, x_m в выходной предикат.*

4) Пример работы алгоритма

Проиллюстрируем теперь полную работу алгоритма на примере 3 из § 1 настоящей главы (стр. 220, см. также введение к книге). Приведенный там оператор может быть выражен следующей формулой:

$$z(t) = (\text{E}\sigma)_{\sigma < t} \{x_1(\sigma) \& (\tau)_{\sigma < \tau < t} [x_2(\tau) \vee (\text{E}\lambda)_{\sigma < \lambda < \tau} x_3(\lambda)]\}.$$

Исходная формула уже является регулярной; поэтому к формуле $\Gamma_1(t+1)$ можно применить операцию А, в результате чего мы получим:

$$\begin{aligned} \Gamma_1(t+1) &= (\text{E}\sigma)_{\sigma < t} \{x_1(\sigma) \& (\tau)_{\sigma < \tau < t} [x_2(\tau) \vee (\text{E}\lambda)_{\sigma < \lambda < \tau} x_3(\lambda)]\} = \\ &= (\text{E}\sigma)_{\sigma < t} \{x_1(\sigma) \& (\tau)_{\sigma < \tau < t} [x_2(\tau) \vee (\text{E}\lambda)_{\sigma < \lambda < \tau} x_3(\lambda)]\} \vee \\ &\vee \{x_1(t) \& (\tau)_{t < \tau < t} [x_2(\tau) \vee (\text{E}\lambda)_{\sigma < \lambda < \tau} x_3(\lambda)]\} = \\ &= (\text{E}\sigma)_{\sigma < t} \{x_1(\sigma) \& (\tau)_{\sigma < \tau < t} [x_2(\tau) \vee (\text{E}\lambda)_{\sigma < \lambda < \tau} x_3(\lambda)]\} \vee x_1(t) = \\ &= (\text{E}\sigma)_{\sigma < t} \{x_1(\sigma) \& (\tau)_{\sigma < \tau < t} [x_2(\tau) \vee (\text{E}\lambda)_{\sigma < \lambda < \tau} x_3(\lambda)] \& \\ &\& [x_2(t) \vee (\text{E}\lambda)_{\sigma < \lambda < t} x_3(\lambda)]\} \vee x_1(t). \end{aligned}$$

Далее следует применять операцию Б, которая приводит к представлению:

$$\Gamma_1(t+1) = x_1(t) \vee [z(t) x_2(t)] \vee \Gamma_2(t),$$

где $\Gamma_2(t)$ есть сокращенное обозначение регулярного спутника:

$$(\text{E}\sigma)_{\sigma < t} \{x_1(\sigma) \& (\tau)_{\sigma < \tau < t} [x_2(\tau) \vee (\text{E}\lambda)_{\sigma < \lambda < \tau} x_3(\lambda)] \& (\text{E}\lambda)_{\sigma < \lambda < t} x_3(\lambda)\}.$$

Дальнейшим преобразованиям подлежит только формула $\Gamma_2(t)$, ибо регулярный спутник $\Gamma_1(t)$ был подвергнут необходимым преобразованиям уже ранее. Применяя к $\Gamma_2(t)$ операцию А, а потом операцию Б, мы получаем:

$$\Gamma_3(t+1) = [z(t) \& x_2(t) \& x_3(t)] \vee \Gamma_2(t).$$

Поскольку новых регулярных спутников здесь не появилось, на этом процесс обрывается. Искомые канонические уравнения нашего оператора имеют вид:

$$z(t) = q_1(t),$$

$$q_1(t+1) = x_1(t) \vee [q_1(t) \cdot x_2(t)] \vee q_2(t),$$

$$q_2(t+1) = [q_1(t) \cdot x_2(t) \cdot x_3(t)] \vee q_2(t)$$

при начальных условиях $q_1(1)=0$, $q_2(1)=0$.

5) Расширение языка

Естественно, возникает вопрос о справедливости теоремы, обратной предшествующей. Иначе говоря, возникает вопрос, можно ли любой ограниченно-детерминированный оператор, преобразующий систему входных предикатов в выходной предикат, задавать посредством t -формулы без предикатных кванторов? Следующий пример показывает, что на этот вопрос следует дать отрицательный ответ.

Допустим, что оператор может быть задан посредством t -формулы $z(t) = \mathfrak{A}(x, t)$. Тогда, подставив вместо предикатной переменной x предикат, тождественно равный 1, мы должны были бы получить предикат $z(t)$, равный 1 при нечетных t и равный 0 при четных t . Однако можно показать, что в результате указанной подстановки в формулу \mathfrak{A} из нее может быть получен лишь такой предикат, который сохраняет постоянное значение, то есть 1, или 0, для всех достаточно больших t *).

В связи с этим интересно выяснить, каким образом можно расширить рассмотренный раньше формальный язык с тем, чтобы в нем возможно было описать все ограниченно-детерминированные операторы. Для решения этого вопроса введем в рассмотрение некоторые возможные расширения языка t -формул. Дело в том, что описанный ранее алгоритм применим фактически к более широкому формальному языку, чем тот, который привлекался нами

*) См. также C. Elgot, J. Wright, Quantifier elimination in a problem of logical design. Michigan Mathematical Journal, 1959, v. 6, № 1, p. 65—69.

до сих пор. Рассмотрим, например, оператор, определяемый условием

$$z(t) = (\mathbf{E}\sigma) [x_1(\sigma) \& (\tau) z(\sigma)]_{\substack{\sigma < t \\ \sigma < \tau < t}}.$$

В отличие от формулы (5.18) здесь в правой части фигурирует определяемый предикат, отнесенный ко времени, предшествующему времени t . Однако легко видеть, что алгоритм полностью применим и в подобных случаях.

Дальнейшее расширение языка, при котором наш алгоритм еще применим, может быть получено за счет допущения аргументов вида $\sigma - k$ (где σ — переменная, k — фиксированное натуральное число). Пусть, например, оператор выражен формулой

$$z(t) = (\mathbf{E}\sigma) (x_1(\sigma - 1) \& (\tau) x_2(\sigma - \tau))_{\substack{\sigma < t \\ \sigma < \tau < t}}.$$

Аргументы вида $\sigma - k$ можно было бы элиминировать, интерпретируя, например, $x_1(\sigma - 1)$ как $(\mathbf{E}\lambda) [\sigma = \lambda + 1 \& x_1(\lambda)]^*$. Отсюда уже вытекает, что такие расширения не обеспечивают еще обратимости теоремы 3.

Более существенным является расширение формального языка за счет допущения предикатных кванторов.

Именно, справедлива следующая теорема.

Т е о р е м а 4. *Каков бы ни был ограниченно-детерминированный оператор $z(t) = T[x_1, \dots, x_m, t]$, существует t -формула $\mathfrak{A}[x_1, \dots, x_m, t]$ расширенного исчисления предикатов с ограниченными предметными кванторами, которая задает этот оператор.*

Д о к а з а т е л ь с т в о. Пусть, для определенности, оператор T описывается следующими каноническими уравнениями:

$$\begin{aligned} z(t) &= \Phi[x_1(t), x_2(t), \Gamma_1(t), \Gamma_2(t)], \\ \Gamma_1(t+1) &= \Psi_1[x_1(t), x_2(t), \Gamma_1(t), \Gamma_2(t)], \\ \Gamma_2(t+1) &= \Psi_2[x_1(t), x_2(t), \Gamma_1(t), \Gamma_2(t)] \end{aligned}$$

при начальных условиях $\Gamma_1(1) = 1$ и $\Gamma_2(1) = 0$.

*) На самом деле алгоритм можно применять при некоторой незначительной его модификации без того, чтобы элиминировать такого рода выражения.

Тогда, как легко понять, это описание может быть выражено также и таким образом: *)

$$\begin{aligned} z(t) &= (E\Gamma_1)(E\Gamma_2) \{ \Gamma_1(1) \& \bar{\Gamma}_2(1) \& (\tau) [\Gamma_1(\tau) \cdot 1 < \tau \equiv \\ &\quad \tau \leq t \\ &\quad \equiv \Psi_1(x_1(\tau-1), x_2(\tau-1), \Gamma_1(\tau-1), \Gamma_2(\tau-1))] \& \\ &\quad \& (\tau) [\Gamma_2(\tau) \cdot 1 < \tau \equiv \\ &\quad \tau \leq t \\ &\quad \equiv \Psi_2(x_1(\tau-1), x_2(\tau-1), \Gamma_1(\tau-1), \Gamma_2(\tau-1))] \& \\ &\quad \& \Phi(x_1(t), x_2(t), \Gamma_1(t), \Gamma_2(t)) \} ; \end{aligned}$$

$\Gamma(1)$ может быть выражена как $(E\sigma) [\sigma = 1 \cdot \Gamma(\sigma)]$, а вхождения аргументов $\tau - 1$ можно элиминировать так, как это было указано выше. Теорема, таким образом, доказана.

Вместе с тем справедлива и обратная к ней теорема 3', являющаяся усилением теоремы 3.

Теорема 3'. *Всякая t -формула $\mathcal{A}[x_1, \dots, x_m, t]$ расширенного исчисления одноместных предикатов с ограниченными предметными кванторами задает ограниченно-детерминированный оператор, преобразующий систему входных предикатов x_1, \dots, x_m в выходной предикат.*

Доказательство. Спуск кванторов (предметных!) производится так же, как и прежде. Таким образом, остается показать, как осуществляется разложение по регулярным спутникам в общем случае, то есть в случае, когда в формуле могут встречаться кванторы по предикатам. Иррегулярные вхождения $x_i(t)$ свободных предикатов выносятся так же, как и раньше. Вместе с тем справедливо следующее утверждение, из которого вытекает способ элиминации иррегулярных вхождений связанных предикатов.

Пусть \mathcal{A} — произвольная t -формула с ограниченными кванторами, причем t входит как граница кванторов $[q]$, но не кванторов $[q]^*$). Тогда любую ее подформулу вида $(E\Gamma) D[\dots \Gamma(t) \dots]$ можно заменить формулой

*) См. стр. 82—83.

**) То есть предполагается, что спуск кванторов уже осуществлен.

$(E\Gamma) [D [\dots И \dots] \vee D [\dots Л \dots]]$, а подформулу $(\Gamma)D [\dots \Gamma(t) \dots]$ заменить формулой $(\Gamma)[D (\dots И \dots) \& D (\dots Л \dots)]$.

Для обоснования этого утверждения заметим сначала следующее. Пусть в некоторой формуле D вхождения предиката Γ встречаются только в подформулах вида $\Gamma(\tau) \cdot \tau < t$, $\Gamma(q) \cdot q < t$ и т. д. Тогда $(E\Gamma)D$ экв. $(E\Gamma)[\Gamma(t) \cdot D]$ экв. экв. $(E\Gamma)[\bar{\Gamma}(t) \cdot D]$, поскольку ложность или истинность формулы D не зависит от значения Γ для аргументов $\geq t$.

Рассмотрим теперь, для определенности, подформулу с квантором существования. Преобразуя область действия квантора к виду $\Gamma(t) \cdot D (\dots И \dots) \vee \bar{\Gamma}(t) D (\dots Л \dots)$, можно саму подформулу привести к виду $(E\Gamma)[\Gamma(t) \cdot D (\dots И \dots)] \vee \vee (E\Gamma)[\bar{\Gamma}(t) \cdot D (\dots Л \dots)]$. Дизъюнктивные же члены можно заменить соответственно на $(E\Gamma)D [\dots И \dots]$ и $E[\Gamma]D [\dots Л \dots]$, если учесть предыдущее замечание, а также следующее. В наших условиях вхождения Γ в подформулы $D [\dots И \dots]$ и $D [\dots Л \dots]$ возможны лишь с замещением аргументных мест переменными q, τ, \dots , которые являются связанными, а значит и строго t -контролируемыми. Поэтому допустима переинтерпретация $\Gamma(q), \Gamma(\tau), \dots$ как $\Gamma(q) \cdot q < t, \Gamma(\tau) \cdot \tau < t, \dots$

Итак, допустима замена подформулы $(E\Gamma)D [\dots \Gamma(t) \dots]$ на формулу $(E\Gamma)D [\dots И \dots] \vee (E\Gamma)D [\dots Л \dots]$, а следовательно, и на формулу $(E\Gamma)[D (\dots И \dots) \vee D (\dots Л \dots)]$. Аналогично элиминируется иррегулярность $\Gamma(t)$ из-под формулы вида $(\Gamma)D (\dots \Gamma(t) \dots)$. Теорема, таким образом, доказана.

Для иллюстрации применения этой теоремы ограничимся разложением по регулярным спутникам следующей формулы:

$$(E\tau) \{x(\tau) \cdot (\Gamma) [y(t) \cdot \Gamma(\tau) \vee_{\tau < q < t} (q) (\Gamma(t) \cdot x(q))]\}.$$

Процесс начинается с выноса $y(t)$ из формулы $\mathfrak{A} [\dots y(t) \dots]$. Мы имеем:

$$\mathfrak{A} [\dots И \dots] = (E\tau) \{x(\tau) \cdot (\Gamma) [\Gamma(\tau) \vee_{\tau < q < t} (q) (\Gamma(t) \cdot x(q))]\},$$

$$\mathfrak{A} [\dots Л \dots] = (E\tau) \{x(\tau) \cdot (\Gamma) (q) (\Gamma(t) \cdot x(q))\}$$

Обозначая области действия квантора (Γ) в первой из этих формул через $\mathfrak{B}_1[\dots \Gamma(t) \dots]$, а во второй — через $\mathfrak{B}_2[\dots \Gamma(t) \dots]$, мы получаем:

$$\mathfrak{B}_1[\dots И \dots] = \Gamma(\tau) \vee_{\tau < q < t} (q) x(q),$$

$$\mathfrak{B}_1[\dots Л \dots] = \Gamma(\tau) \vee_{\tau < q < t} (q) Л,$$

$$\mathfrak{B}_2[\dots И \dots] =_{\tau < q < t} (q) x(q),$$

$$\mathfrak{B}_2[\dots Л \dots] =_{\tau < q < t} (q) Л.$$

Разложение по регулярным спутникам получает вид

$$y(t) \cdot (\mathbf{E}\tau)_{\tau < t} \{x(\tau) \cdot (\Gamma)[(\Gamma(\tau) \vee_{\tau < q < t} (q) x(q)) \cdot (\Gamma(\tau) \vee_{\tau < q < t} (q) Л)]\} \vee \\ \vee \bar{y}(t) \cdot (\mathbf{E}\tau)_{\tau < t} \{x(\tau) \cdot (\Gamma)[(q) x(q) \cdot (q) Л]\}.$$

После надлежащих упрощений мы имеем:

$$y(t) \cdot (\mathbf{E}\tau)_{\tau < t} \{x(\tau) \cdot (\Gamma)[\Gamma(\tau) \vee_{\tau < q < t} (q) Л]\} \vee \bar{y}(t) \cdot (\mathbf{E}\tau)_{\tau < t} [x(\tau) \cdot (q) Л]_{\tau < q < t}.$$

З а м е ч а н и я.

1°. До сих пор мы рассматривали операторы с одним выходным предикатом. Ясно, что в случае нескольких выходных предикатов z_1, z_2, \dots, z_n система t -формул

$$z_1(t) = \mathfrak{A}_1[x_1, \dots, x_m, t],$$

$$z_2(t) = \mathfrak{A}_2[x_1, \dots, x_m, t],$$

$$\dots \dots \dots$$

$$z_n(t) = \mathfrak{A}_n[x_1, \dots, x_m, t]$$

будет также перерабатываться в систему капонических уравнений.

2°. Если в заданной t -формуле имеются подформулы, являющиеся сами t -формулами (или τ -формулами для некоторой переменной τ), то их целесообразно выделить и подвергнуть самостоятельной переработке. Следующий пример поясняет эту ситуацию.

Пусть дано:

$$z(t) = (\mathbf{E}\tau)_{\tau < t} \{x_1(\tau) \&_{q < \tau} (q) [x_2(q) \vee_{q < \sigma < \tau} (\mathbf{E}\sigma) x_3(\sigma)]\}.$$

Тогда

$$z(t) = (\mathbf{E}\tau) [x_1(\tau) \& \Gamma(\tau)],$$

где $\Gamma(\tau)$ есть τ -формула, которая может быть самостоятельной по преобразована. Мы имеем (см. переход от 5.18 к 5.22):

$$z(t+1) = z(t) \vee [x(t) \& \Gamma(t)],$$

$$\Gamma(t+1) = [\Gamma(t) \vee x_3(t)] \cdot x_2(t).$$

Укажем еще некоторые расширения рассмотренного ранее языка, для которых все еще применим (при соответствующих естественных модификациях) прежний алгоритм.

Допустим, что аргументные места предикатов разрешается замещать не только переменными, но и выражениями (так называемыми термами) вида k_i и $q + k_i$, где k_i — произвольные натуральные константы. Если объявить иррегулярностями вхождения вида $x(t+s)$, $\Gamma(t+s)$, ..., где s — наибольшая из констант, фигурирующих в формуле, то алгоритм, примененный, например, к формуле $\mathfrak{A}(x, t)$, работая в остальном как и прежде, приведет к уравнениям вида:

$$\left. \begin{aligned} z(t) &= \Phi[x(t), x(t+1), \dots \\ &\quad \dots, x(t+s), q_1(t), \dots, q_k(t)], \\ q_j(t+1) &= \Psi_j[x(t), x(t+1), \dots \\ &\quad \dots, x(t+s), q_1(t), \dots, q_k(t)], \\ q_j(1) &= X_j[x(1), \dots, x(s)]. \end{aligned} \right\} \quad (*)$$

Эти уравнения задают оператор с s -предвосхищением (условие детерминированности нарушено при $s > 0$), но с конечной памятью. Если натуральные константы не участвуют в замещениях аргументных мест для свободных предикатов (для связанных предикатов это допускается), то формула будет по-прежнему задавать ограниченно-детерминированный оператор, канонические уравнения которого будут получены в результате работы алгоритма.

Рассмотрим простой пример:

$$\begin{aligned} z(t) &= (\mathbf{E}\tau)_{\tau < t} (\mathbf{E}\Gamma) [x(\tau + 3) \cdot (\mathbf{q})_{\mathbf{q} < \tau} \Gamma(\mathbf{q} + 4)], \\ z(t+1) &= (\mathbf{E}\tau)_{\tau \leq t} (\mathbf{E}\Gamma) [x(\tau + 3) \cdot (\mathbf{q})_{\mathbf{q} < \tau} \Gamma(\mathbf{q} + 4)], \\ z(t+1) &= (\mathbf{E}\tau)_{\tau < t} (\mathbf{E}\Gamma) [x(\tau + 3) \cdot (\mathbf{q})_{\mathbf{q} < \tau} \Gamma(\mathbf{q} + 4)] \vee \\ &\quad \vee (\mathbf{E}\Gamma) [x(t+3) \cdot (\mathbf{q})_{\mathbf{q} < t} \Gamma(\mathbf{q} + 4)], \\ z(t+1) &= z(t) \vee x(t+3) \cdot q(t), \end{aligned}$$

где

$$q(t) = (\mathbf{E}\Gamma) (\mathbf{q})_{\mathbf{q} < t} \Gamma(\mathbf{q} + 4).$$

Ясно, что $q(t)$ тождественно равно 1. Однако если бы мы этого и не заметили, то дальнейшая работа алгоритма протекала бы так:

$$q(t+1) = (\mathbf{E}\Gamma) [(\mathbf{q})_{\mathbf{q} < t} \Gamma(\mathbf{q} + 4) \cdot \Gamma(t+4)];$$

после элиминаций иррегулярности $\Gamma(t+4)$ мы получили бы:

$$q(t+1) = q(t).$$

В данном случае получаются уравнения оператора с предвосхищением:

$$\begin{aligned} z(t) &= p(t), \\ p(t+1) &= p(t) \vee x(t+3), \\ p(1) &= 0. \end{aligned}$$

В заключение заметим, что алгоритм применим непосредственно к t -формулам, в которых фигурируют равенства и неравенства термов, без предварительной их замены определяющими их формулами с ограниченными кванторами (см. гл. I, § 5). Необходимые в этом случае модификации алгоритма таковы:

а) Наряду с иррегулярностями типа $x(t)$ при разложении на регулярные спутники выносятся также иррегулярности типа $t = k$, $t < k$.

б) Благодаря строгой t -контролируемости связанных переменных допустима замена на Π выражений вида $q + k_1 < t + k_2$ при $k_2 \geq k_1$.

в) Для отношений вида $t + k_1 = t + k_2$, $t + k_1 < t + k_2$ значение истинности определяется непосредственно. (Эти дополнительные операции предотвращают рост констант в регулярных формулах.)

* *
* *

Вернемся теперь к задаче о переездной железнодорожной сигнализации, рассмотренной в предыдущем параграфе (см. стр. 251—252). К полученной там формуле для $z(t)$ можно применить предложенный алгоритм. Для этого достаточно выделить t -формулу $(\exists \tau) \{A(\tau) \cdot B(\tau) \& (q) [\bar{A}(q) B(q)]\}$ $\tau < t$ $\tau < q < t$ и преобразовать ее. Тогда мы будем иметь:

$$\left. \begin{aligned} z(t) &= \bar{A}(t) \cdot \bar{B}(t) \vee \bar{A}(t) \cdot B(t) \cdot q_1(t) \vee \\ &\quad \vee A(t) \cdot \bar{B}(t) \cdot q_2(t), \\ q_1(t+1) &= q_1(t) \cdot \bar{A}(t) B(t) \vee A(t) \cdot B(t), \\ q_2(t+1) &= q_2(t) \cdot A(t) \cdot \bar{B}(t) \vee A(t) \cdot B(t), \\ q_1(1) &= 0, \\ q_2(1) &= 0. \end{aligned} \right\} \quad (5.25)$$

Выражения для $q_1(t+1)$ и $q_2(t+1)$ могут быть упрощены с помощью обычных алгебрологических преобразований, на чем мы здесь останавливаться не будем.

Физический смысл внутренних предикатов $q_1(t)$ и $q_2(t)$ очевиден. Они запоминают факт вступления поезда на участок AB в направлении от A к B и в направлении от B к A соответственно.

Примечание. Исходя из физического смысла задачи, можно добиться еще и дальнейших упрощений. Именно, подача в течение некоторого сплошного промежутка времени противоположных значений по входным каналам A и B возможна лишь при условии, что в течение всего этого времени 0 поступает от одного и того же элемента, а 1 от другого. Поэтому для $z(t)$ можно было бы

нений и точно воспроизводит содержащееся в этих уравнениях описание. Поэтому если первоначальное описание какого-либо оператора представляется естественным на языке канонических уравнений, то оно легко осуществляется и посредством логической формулы. Однако это явление необратимо; даже при наличии описания в логических терминах его формулировка посредством канонических уравнений может представлять значительные трудности. Подмеченную только что особенность можно высказать так: язык логики предикатов шире (в некотором смысле) языка канонических уравнений, и поэтому он удобнее для первоначального описания ограниченно-детерминированных операторов. Конечно, при этом мы отвлекаемся от таких субъективных факторов, как личные навыки или характер выбираемых примеров.

Можно показать, что наш логический язык шире (в том же смысле) и другого языка, применяемого в теории конечных автоматов, а именно языка *регулярных выражений*, созданного С. К. Клини и усовершенствованного В. М. Глушковым и другими авторами. Излагая ниже этот вопрос, мы намереваемся привести дополнительные доводы в пользу высказанного выше положения о широте логического языка.

Следуя терминологии Клини, назовем *событием над алфавитом* $\{x_1, x_2, \dots, x_m\}$ любое множество слов в данном алфавите (допускается и пустое слово, обозначаемое впредь буквой e). Вводятся три основные операции над событиями: *дизъюнкция*, *умножение* и *итерация*. Дизъюнкция (теоретико-множественное объединение) $S_1 \vee S_2$ двух событий S_1, S_2 состоит в точности из тех слов, каждое из которых принадлежит по крайней мере одному из событий S_1, S_2 .

Произведение $S_1 \cdot S_2$ событий S_1, S_2 состоит в точности из тех слов, которые могут быть получены путем приписывания справа какого-либо слова из S_2 какому-либо слову из S_1 .

В силу очевидной ассоциативности операций « \vee » и « \cdot » можно, опуская скобки, указывающие порядок действий, рассматривать многочленные дизъюнкции $S_1 \vee S_2 \vee \dots \vee S_n$ и многочленные произведения $S_1 \cdot S_2 \cdot \dots \cdot S_n$. Итерация $\{S\}$ события S есть событие, состоящее в точности из тех

слов, которые принадлежат по крайней мере одному из событий

$$e, S, S \vee S, S \vee S \vee S, \dots$$

Иными словами, в итерацию $\{S\}$, кроме пустого слова, входит всякое такое слово A , которое может быть получено путем приписывания друг за другом любого конечного числа слов из S . Назовем *элементарными событиями* $m+1$ одноэлементных множеств, составленных из однобуквенных входных слов x_1, \dots, x_m и из пустого слова e . Назовем *регулярными событиями* все элементарные события, а также все те события, которые могут быть получены из элементарных событий с помощью конечного числа дизъюнкций, умножений и итераций.

В соответствии с этим определением каждое регулярное событие может быть задано (причем многими разными способами) посредством так называемых *регулярных выражений*, которые индуктивно определяются следующим образом:

1. x_1, x_2, \dots, x_m, e являются регулярными выражениями.

2. Если $\mathfrak{M}, \mathfrak{N}$ — регулярные выражения, то таковыми являются и выражения

$$(\mathfrak{M} \vee \mathfrak{N}), (\mathfrak{M}, \mathfrak{N}), \{\mathfrak{M}\}.$$

3. Других регулярных выражений не существует.

П р и м е р. В алфавите $\{x, y, z\}$ регулярное выражение

$$((x \vee y)) \cdot (z \cdot z)$$

задает событие, состоящее из всех слов, которые оканчиваются двумя буквами z .

Скобки (круглые), применяемые для обозначения порядка действий, иногда опускаются; опускание скобок происходит в соответствии с правилом: в случае отсутствия скобок первыми выполняются итерации, затем умножения и, наконец, дизъюнкции. Можно также опускать и точку (знак умножения). Кроме того, могут опускаться внешние скобки. Например, предыдущее выражение можно записать так:

$$\{x \vee y\} z z.$$

Каждому ограниченно-детерминированному оператору T с двумя выходными буквами (для определенности: 1 и 0) соответствует событие S_T , состоящее из всех таких входных слов, которые перерабатываются в выходные слова, оканчивающиеся единицей. При этих условиях говорят, что оператор T *распознает* наступление события S_T .

Говорят, что событие *представимо в конечном автомате*, если существует ограниченно-детерминированный оператор, распознающий его наступление. Относительно представимости событий в конечных автоматах справедлива следующая теорема.

Теорема.*) *Для того чтобы событие было представимо в конечном автомате, необходимо и достаточно, чтобы оно было регулярным.*

Существуют различные доказательства этой теоремы. Мы ограничимся здесь установлением *достаточности*. Это может быть сделано путем указания для каждого регулярного выражения \mathfrak{M} такой t -формулы $\mathfrak{A}^{\mathfrak{M}}(t)$, которая эквивалентна ей в следующем смысле: оператор, описываемый формулой $\mathfrak{A}^{\mathfrak{M}}$, распознает наступление регулярного события, задаваемого выражением \mathfrak{M} . Процедура перехода от \mathfrak{M} и $\mathfrak{A}^{\mathfrak{M}}$ строится ниже индуктивно в соответствии с правилами образования регулярных выражений. В сущности она представляет собой естественное погружение языка регулярных выражений в наш логический язык. Это обстоятельство и должно мотивировать признание нашего логического языка более широким, чем язык регулярных выражений **).

Предварительно условимся о следующих обозначениях. Для произвольной t -формулы $\mathfrak{B}(t)$, не содержащей предметной переменной τ , формула $\mathfrak{B}_{\tau}(t)$ есть формула, получаемая из $\mathfrak{B}(t)$ путем замены всякого *о д н о с т о р о н н е г о* квантора $[\lambda]$ *д в у с т о р о н н и м* квантором $[\lambda]$.

$\lambda \leq \mu$
 $\tau < \lambda < \mu$

*) См. С. К. Клини, Представление событий в нервных сетях и конечных автоматах, Сб. «Автоматы», ИЛ, М., 1956.

**) При этом следует иметь в виду, что переход (который мы здесь рассматривать не будем) от логической формулы к регулярному выражению весьма сложен.

П р и м е р. Обозначим t -формулу

$$(\mathbf{E}q) [X(q) \cdot (\sigma)_{\sigma < q} Y(\sigma)]$$

через $\mathfrak{B}(t)$; тогда формула

$$(\mathbf{E}q) [X(q) \cdot (\sigma)_{\tau < \sigma < q} Y(\sigma)]$$

будет формулой $\mathfrak{B}_\tau(t)$.

Очевидно, что $\mathfrak{B}_\tau(t)$ уже не есть t -формула, ибо она содержит две свободные переменные τ и t . Ее смысл ясен: утверждение о входном слове, содержащееся в $\mathfrak{B}(t)$, теперь высказывается лишь о части входного слова, начинающейся с $(\tau + 1)$ -й буквы и простирающейся до конца входного слова.

Приведем теперь правила построения формул $\mathfrak{A}^{\mathfrak{M}}$, $\mathfrak{A}^{\mathfrak{N}}$, ... по заданным \mathfrak{M} , \mathfrak{N} , ... В них входным буквам x_1, \dots, x_m сопоставлены предикаты $X_1(t), \dots, X_m(t)$, имеющие смысл: t -я слева буква во входном слове есть x_1 (соответственно x_2, \dots, x_m). Пустому слову e сопоставлен тождественно ложный предикат \mathcal{L} :

1. Пусть \mathfrak{M} есть x_1, \dots, x_m или e . Тогда $\mathfrak{A}^{\mathfrak{M}}$ есть $X_1(t) \cdot t = 1$ (соответственно есть $X_2(t) \cdot t = 1, \dots, X_m(t) \cdot t = 1, \mathcal{L}$).

Пусть регулярным выражениям $\mathfrak{M}, \mathfrak{N}$ уже сопоставлены формулы $\mathfrak{A}^{\mathfrak{M}}, \mathfrak{A}^{\mathfrak{N}}$. Тогда:

2. Выражению $\mathfrak{M} \vee \mathfrak{N}$ соответствует формула

$$\mathfrak{A}^{\mathfrak{M}} \vee \mathfrak{A}^{\mathfrak{N}}.$$

3. Выражению $\mathfrak{M} \cdot \mathfrak{N}$ соответствует формула:

$$(\mathbf{E}\tau) [\mathfrak{A}^{\mathfrak{M}}(\tau) \cdot \mathfrak{A}^{\mathfrak{N}}_\tau(t)].$$

4. Выражению $\{\mathfrak{M}\}$ соответствует формула:

$$(\mathbf{E}\Gamma) (\Gamma(t) \& (\tau)_{\tau \leq t} (\Gamma(\tau) \cdot (q)_{q < \tau} \bar{\Gamma}(q) \rightarrow \mathfrak{A}^{\mathfrak{M}}(\tau)) \& (\tau)_{\tau \leq t} (\sigma)_{\sigma < \tau} (\Gamma(\sigma) \cdot \Gamma(\tau) \cdot (q)_{\sigma < q < \tau} \bar{\Gamma}(q) \rightarrow \mathfrak{A}^{\mathfrak{M}}_\sigma(\tau))).$$

П о я с н е н и е. Предикат Γ истинен при таких значениях аргумента $\tau_1 < \tau_2 < \dots < t$, при которых слово длины t из события $\{\mathfrak{M}\}$ разбивается на куски, имеющие вид: от начала до τ_1 -й буквы, от (τ_1+1) -й буквы до τ_2 -й буквы и т. д., каждый из которых есть слово из события \mathfrak{M} .

Вместо 4 можно применять:

4'. Выражению $\{\mathfrak{M}\}$ сопоставляется формула:

$$(\mathbf{E}\Gamma) \{ \Gamma(t) \cdot (\sigma) (\Gamma(\sigma) \equiv \mathfrak{M}(\sigma) \vee (\mathbf{F}\tau) (\Gamma(\tau) \cdot \mathfrak{M}_{\tau}(\sigma)) \}.$$

ПРАКТИЧЕСКИЕ ПРИЕМЫ СИНТЕЗА АВТОМАТОВ

§ 1. Общие замечания

При изложении практических приемов синтеза логической сети мы будем полагать, что оператор, подлежащий реализации, задан одним из способов, рассмотренных в предыдущей главе, и на его основе составлены канонические уравнения оператора, выраженные, например, в виде формул алгебры логики.

Мы будем также полагать, что указаны физические элементы, из которых должна строиться сеть, и эти элементы образуют функционально полные наборы. Таким образом обеспечивается возможность построения логической сети, моделирующей канонические уравнения заданного оператора. Необходимое для этого общее число элементов зависит от общего числа символов, входящих в моделируемые канонические уравнения, и от характера операторов того набора элементов, который нами используется.

Как указывалось в главе третьей, физические элементы, применяемые для построения логической сети, обеспечивают возможность выбора различных полных систем элементарных функций. Вместе с тем путем тождественных преобразований канонических уравнений могут быть получены эквивалентные формулы алгебры логики с различным числом символов. При синтезе логической сети стремятся, естественно, к ее оптимизации по некоторым параметрам. Чаще всего ставится задача построения сети, имеющей минимальное число физических элементов (или минимальное число входных каналов элементов сети).

Наиболее полно разработаны в настоящее время приемы *синтеза*, основанные на том предположении, что независимо от набора элементов, используемых для построения сети, наиболее простая схема получается при моделировании *минимальных форм* представления функций алгебры логики (*минимальных форм ул* *), представленных в виде суперпозиции *отрицания*, *дизъюнкции* и *конъюнкции*. Это обуславливает методику синтеза, получившую в практике наиболее широкое применение. Она сводится к выполнению *двух* основных этапов. Вначале канонические уравнения оператора подвергаются тождественным преобразованиям для получения *минимальных форм ул*, выраженных через дизъюнкцию, конъюнкцию и отрицания. При этом принимаются во внимание возможности упрощения формул за счет выбора двоичных кодов для алфавитов — внешних и внутреннего. В первой главе были рассмотрены некоторые общие способы минимизации формул алгебры логики, обеспечивающие во многих практически важных случаях эффективную реализацию этого процесса.

Второй этап сводится к преобразованию минимизированных канонических уравнений в суперпозицию операторов тех элементов, из которых строится логическая сеть. Это достигается либо путем представления формул канонических уравнений в виде суперпозиции операторов выбранных элементов, либо путем построения из этих элементов блоков, реализующих операторы отрицания, дизъюнкции и конъюнкции. По полученным тем или иным способом формулам строится логическая сеть, реализующая заданный оператор.

Необходимо отметить, что выполнение обоих указанных этапов синтеза логической сети сопряжено с перебором. Это обуславливается тем, что, как указывалось в первой главе, все известные в настоящее время способы минимизации формул алгебры логики в той или иной степени связаны с перебором. Необходимость перебора возникает также при преобразовании канонических уравнений в формулы, содержащие суперпозиции операторов

*) См. § 3 гл. I.

тех элементов, которые нами используются. Таким образом, выбор окончательной схемы логической сети в большой мере зависит от конкретных условий ее проектирования и эксплуатации, и определяющее значение здесь имеют опыт и квалификация проектировщика.

Во второй главе было показано, что реализация оператора по каноническим уравнениям приводит обычно к необходимости моделирования в логической сети системы функций алгебры логики. В этих случаях сеть должна иметь соответствующее число выходных полюсов (каналов), каждый из которых выдает по одной функции алгебры логики (из числа принадлежащих к заданной системе функций). При этом логическую сеть можно строить либо путем раздельной реализации каждой функции, то есть путем независимого построения для каждой функции своей логической сети, либо пользуясь методами совместной реализации всей системы в целом или отдельных ее частей. В последнем случае некоторые элементы сети используются для реализации нескольких функций.

В § 2 настоящей главы мы рассмотрим способы раздельной реализации, то есть способы построения логической сети, вырабатывающей в одном из полюсов требуемую функцию алгебры логики. Такую сеть, следуя Айкену *), мы будем называть *одновыходной* сетью. Синтез одновыходных сетей будет рассматриваться применительно к нормальной дизъюнктивной или конъюнктивной формам.

В § 3 мы укажем прием synthesis логических сетей посредством совместной реализации системы функций. Такие сети мы условно называем *многовыходными*. При синтезе многовыходных сетей рассматривается схемное выполнение не только нормальных формул алгебры логики, но и формул другого вида. Методы построения таких схем пригодны, в частности, и для синтеза одновыходных сетей.

В последнем параграфе содержится пример синтеза логической сети для решения лабиринтной задачи.

*) См. Синтез электронных вычислительных и управляющих схем, ИЛ, М., 1954.

§ 2. Синтез одновыходной логической сети

Пусть оператор, подлежащий реализации в логической сети, задан каноническими уравнениями, представленными в дизъюнктивной нормальной форме. Будем полагать, что необходимые упрощения канонических уравнений проведены теми или иными способами минимизации и они подготовлены для дальнейших преобразований применительно к операторам элементов, принятых для реализации сети.

Приведем некоторые общие соображения, связанные с выполнением этих преобразований путем суперпозиции истинностных операторов электронно-ламповых элементов, рассмотренных в главе третьей.

Нормальная дизъюнктивная форма m переменных

$$f(x_1, x_2, \dots, x_m) = \bigvee_{\sigma_1, \sigma_2, \dots, \sigma_m} f(\sigma_1, \sigma_2, \dots, \sigma_m) [x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m}] \quad (6.1)$$

непосредственно образуется суперпозицией операторов триода и операторов диодных схем. Примем во внимание, что $x_i^0 = \bar{x}_i = T(x_i^1)$, и в формуле (6.1) члены конъюнкций $x_i^{\sigma_i}$, для которых $\sigma_i = 0$, представим операторами триода $T(x_i)$. Тогда (6.1) может быть записано в следующем виде:

$$D_g \{D_h[x_1, x_2, \dots, T(x_k), \dots, T(x_j)], \dots, D_h[x_1, x_2, \dots, \dots, T(x_j), \dots, T(x_m)]\}. \quad (6.2)$$

Формула (6.2) однозначно определяет схему коммутации элементов и общее число входных каналов в сети, реализующей функцию (6.1). Это число соответствует общему числу символов D_h , T и x_i , заключенных в фигурные скобки формулы (6.2).

Пример 1. Дана функция $z(x_1, x_2, x_3)$, заданная формулой $x_1 x_2 \vee x_2 x_3 \vee x_1 x_3$. Согласно (6.2), преобразуем эту функцию в суперпозиции операторов триода и диодных схем и получим:

$$\begin{aligned} z &= x_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee x_1 \bar{x}_3 = \\ &= D_g \{D_h[x_1, T(x_2)], D_h[x_2, T(x_3)], D_h[T(x_1), x_3]\}. \end{aligned}$$

Внутри фигурной скобки полученного выражения заключено двенадцать символов D_k , T и x_i . Следовательно, схема, реализующая заданную функцию, имеет двенадцать входных каналов. Она представлена на рис. 6.1.

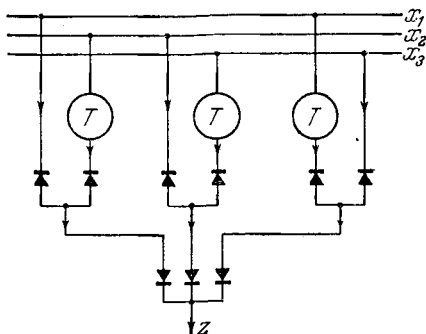


Рис. 6.1.

Дизъюнкция членов в формуле (6.1) может быть получена с помощью операторов двойного катодного повторителя. В этом случае дизъюнкция

$$z = \bigvee_i p_i \quad (6.3)$$

должна быть преобразована в суперпозицию операторов $K_2(p_j, p_k)$ на основе следующего соотношения:

$$z = K_2\{\dots K_2\{K_2[K_2(p_1, p_2), p_3], p_4\}, \dots, p_i\}, \quad (6.4)$$

и общее число входных каналов, необходимых для ее реализации, как и раньше, определяется общим число символов K_2 и p_i , заключенных во внешние фигурные скобки формулы (6.4).

Как указывалось в главе третьей, в электронно-ламповых схемах конъюнкция k переменных, образуемых выходами элементов, может быть реализована «фиктивной ячейкой» — обобществленным анодным сопротивлением этих элементов. При построении логической сети на указанных элементах этот способ находит широкое применение и во многих случаях позволяет сэкономить необходимое число входных каналов.

Таким путем могут быть реализованы, например, конъюнктивные члены дизъюнктивной нормальной формы с помощью операторов $T_k(x_1, x_2, \dots, x_k)$. Следует, однако, иметь в виду, что подключение входных каналов этой «фиктивной» ячейки непосредственно к источнику внешних сигналов не допускается. Для входных каналов, воспринимающих члены конъюнкции, для которых $\sigma_i = 1$,

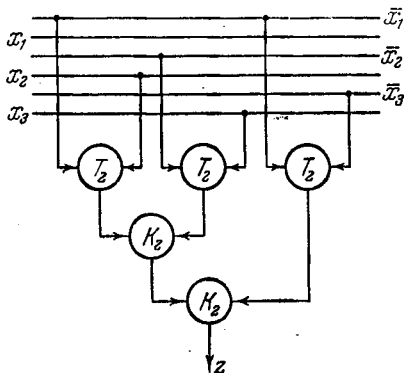


Рис. 6.2.

выполнение этого требования достигается путем двукратного применения оператора $T(x)$, то есть

$$x_i = T(T(x_i)). \quad (6.5)$$

При этом, например, конъюнкция $x_1 \bar{x}_2 \bar{x}_3 x_4$ реализуется оператором $T_4[T(x_1), T(x_2), x_3, x_4]$.

В дальнейшем в целях упрощения изложения мы будем полагать, что при построении схемы сети на ее входах вместе с переменными x_j всегда имеются их отрицания \bar{x}_j .

На рис. 6.2 дана схема реализации функции, рассмотренной в примере 1, путем суперпозиции операторов триода и двойного катодного повторителя на основе формулы

$$z = K_2\{K_2[T_2(\bar{x}_1, x_2), T_2(\bar{x}_2, x_3)], T_2(x_1, \bar{x}_3)\}$$

с общим числом входных каналов — управляющих сеток ламп (не считая каналов, необходимых для образования \bar{x}_i), равным 10.

При построении схемы логической сети с применением пентодов каноническое уравнение должно быть непосредственно представлено в конъюнктивной нормальной форме или к этой форме должна быть приведена формула (6.1)*).

Отметим, что в рассматриваемом случае часто бывает целесообразно строить функцию, заданную формулой (6.1), путем предварительной реализации функции z , представленной в конъюнктивной форме, и последующего применения к ней оператора отрицания

$$\bar{z} = T(\bar{z}). \quad (6.6)$$

При преобразовании функции z , заданной непосредственно в конъюнктивной форме или в виде формулы (6.6), в суперпозицию операторов пентода $\Pi(x_i, x_j)$ необходимо иметь в виду следующие соображения:

1) члены дизъюнкции, для которых $\sigma_i = 1$, должны поступать на входные каналы пентодов своими отрицаниями;

2) при образовании посредством суперпозиции операторов пентодов дизъюнкции, содержащей более двух членов, между выходными полюсами пентодов и подчиненными им входными полюсами последующих пентодов должны быть включены операторы отрицания.

Принимая во внимание эти замечания и полагая, как было указано выше, что отрицания аргументов имеются, представим конъюнктивные члены в виде следующей суперпозиции операторов пентода и триода:

$$\begin{aligned} x_1^{\sigma_1} \vee x_2^{\sigma_2} \vee \dots \vee x_m^{\sigma_m} = \\ = \Pi \{ \dots \Pi \{ T \{ \Pi \{ \bar{x}_1^{\sigma_1}, \bar{x}_2^{\sigma_2} \}, \bar{x}_3^{\sigma_3} \} \bar{x}_4^{\sigma_4} \}, \dots \bar{x}_m^{\sigma_m} \}. \end{aligned} \quad (6.7)$$

Функция \bar{z} , представленная в форме конъюнкций указанных членов, может быть реализована путем объединения выходных каналов, доставляющих дизъюнкции (6.7).

Пример 2. Построить схему с использованием операторов пентода для получения дизъюнкции

$$z = x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4$$

DI

*) Способы такого приведения рассмотрены в главе первой.

в соответствии с формулой (6.7). Представим функцию z в следующем виде:

$$z = \Pi \{T \{ \Pi [T [\Pi (\bar{x}_1, x_2)], \bar{x}_3] \}, x_4 \}.$$

Схема, реализующая эту функцию, изображена на рис. 6.3, а.

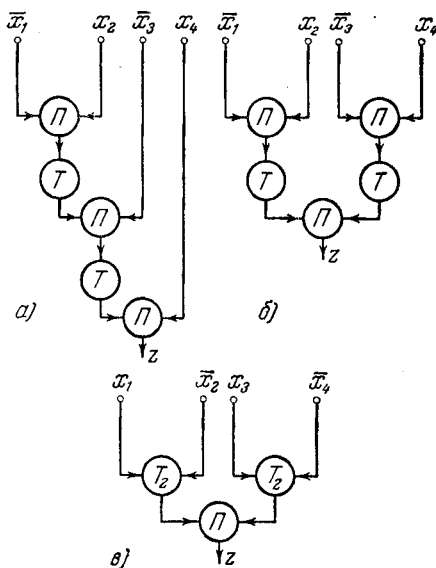


Рис. 6.3.

Отметим, что в рассматриваемом случае (при четном числе членов дизъюнкции) целесообразнее строить схему по соотношению такого вида:

$$z = \Pi \{T [\Pi (\bar{x}_1, x_2)], T [\Pi (\bar{x}_3, x_4)]\},$$

так как схема, моделирующая эту формулу (рис. 6.3, б), имеет меньше каскадов, чем предыдущая.

Однако наиболее экономная схема может быть получена на основе следующей формулы:

$$z = \Pi [T_2 (x_1, x_2), T_2 (x_3, \bar{x}_4)].$$

Она содержит 6 управляющих сеток (рис. 6.3, в) вместо 8 сеток, необходимых для реализации предыдущих схем.

На рис. 6.4 дана схема реализации функции, рассмотренной в примере 1, посредством суперпозиции операторов

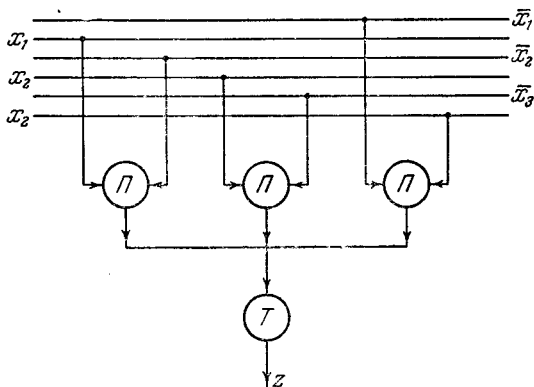


Рис. 6.4.

пентодов. Она построена на основе соотношения (6.6), имеющего в рассматриваемом случае следующий вид:

$$\begin{aligned} z = \bar{z} &= (\bar{x}_1 \vee x_2) (\bar{x}_2 \vee x_3) (x_1 \vee \bar{x}_3) = \\ &= T \{ \Pi(x_1, \bar{x}_2) \& \Pi(x_2, \bar{x}_3) \& \Pi(\bar{x}_1, x_3) \}, \end{aligned}$$

и содержит 7 управляющих сеток в соответствии с общим числом символов x_i , заключенных в фигурной скобке, и одним входным каналом элементарной ячейки $T\{\}$.

Заметим, наконец, что для получения конъюнктивных членов, представляющих собой дизъюнкции вида $(x_1^{s_1} \vee x_2^{s_2} \vee \dots \vee x_m^{s_m})$, можно использовать операторы $\Pi(x_1, x_2, \dots, x_m)$ ячейки, образуемой последовательным подключением полупроводниковых триодов к общему сопротивлению.

Приведенные выше соотношения не охватывают, конечно, всех возможных случаев построения схемы истинностной логической сети на электронно-ламповых элементах, поскольку на практике широко пользуются самыми разнообразными их наборами. Тем не менее эти соотношения доста-

точно хорошо иллюстрируют методику преобразования канонических уравнений в суперпозицию истинностных операторов электронно-ламповых элементов и связанную с этим необходимость перебора различных вариантов.

Заметим, что в общем случае число входных каналов истинностной логической сети, реализующей совершенную нормальную форму от m переменных, не превосходит

$$N \leq 2^{m-1}m + 2^{m-1} = 2^{m-1}(m + 1). \quad (6.8)$$

Здесь слагаемое $2^{m-1}m$ соответствует возможному общему числу вхождений переменных, а слагаемое 2^{m-1} соответствует возможному числу конъюнктивных или дизъюнктивных членов совершенной нормальной формы. При этом принято во внимание то, что совершенная дизъюнктивная нормальная форма, включающая более 2^{m-1} конъюнктивных членов, может быть заменена функцией дополнения, или конъюнктивной формой, имеющей не более 2^{m-1} членов, и наоборот. Возникающая обычно возможность минимизации совершенной нормальной формы уменьшает число входных каналов, необходимых для ее схемной реализации. Для минимальной нормальной формы оно определяется соотношением:

$$N_{\min} = w + \sum_i r_i, \quad (6.9)$$

где w — общее число дизъюнктивных или конъюнктивных членов, а r_i — ранг i -го члена.

Дальнейшая экономия числа входных каналов часто достигается преобразованиями, выводящими синтезируемую функцию из нормального представления. В частности, к таким преобразованиям относится вынос за скобки общих переменных.

Пример 3. Построить электронно-ламповую схему для реализации функции, заданной формулой:

$$z = \overline{x_1}x_2\overline{x_3}x_4 \vee \overline{x_1}\overline{x_2}\overline{x_3}\overline{x_4}.$$

Эта функция не поддается минимизации путем тождественных преобразований, не выводящих ее из нормальной дизъюнктивной (или конъюнктивной) формы. Непосредственная реализация данной функции по задающей ее фор-

муде требует 10 входных каналов (без учета элементов для получения отрицания входных переменных). Вынесем общий множитель за скобку и получим:

$$z = \bar{x}_1 \bar{x}_3 (x_2 x_4 \vee \bar{x}_2 \bar{x}_4).$$

Преобразуем это выражение в суперпозицию операторов триода и пентода. Мы получим:

$$z = \bar{x}_1 \bar{x}_3 (x_2 \vee \bar{x}_4) (\bar{x}_2 \vee x_4) = T_2(x_1, x_3) \& \Pi(\bar{x}_2, x_4) \& \Pi(x_2, \bar{x}_4).$$

Для построения схемы, соответствующей полученной формуле, требуется всего 6 управляющих сеток (рис. 6.5).

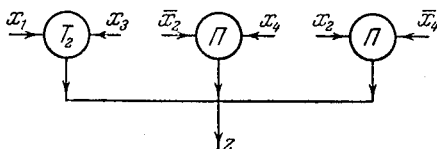


Рис. 6.5.

Во многих случаях значительная экономия числа входных каналов достигается посредством осуществления так называемой пирамидальной схемы построения логической сети, которая основана на разложении функции по независимым переменным. Способ построения таких схем излагается в следующем параграфе применительно к синтезу многовыходных логических сетей, но может быть использован и для синтеза одновыходной сети.

В практике находят широкое применение истинностные логические сети, называемые *дешифраторами* и предназначенные для выработки конъюнкций различных рангов от m переменных. Раздельная реализация каждой из конъюнкций приводит к так называемым *матричным схемам* дешифраторов. Построим в качестве примера схему дешифратора, имеющего 4 входа и 16 независимых выходов, каждый из которых вырабатывает одну конъюнкцию, то есть дешифратора, вырабатывающего все возможные конъюнкции вида $x_1^{\sigma_1} x_2^{\sigma_2} x_3^{\sigma_3} x_4^{\sigma_4}$, где $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ принимают значения 0 или 1 независимо друг от друга.

В соответствии с условиями работы дешифратора выходные функции определяются следующими соотношениями:

$$z_0 = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4},$$

$$z_1 = \overline{x_1} \overline{x_2} \overline{x_3} x_4,$$

$$\dots$$

$$z_7 = \overline{x_1} x_2 \overline{x_3} \overline{x_4},$$

$$\dots$$

$$z_{15} = x_1 x_2 x_3 x_4,$$

и вообще

$$z_j = x_1^{\sigma_1} x_2^{\sigma_2} x_3^{\sigma_3} x_4^{\sigma_4}, \quad (A)$$

где набор $(\sigma_1 \sigma_2 \sigma_3 \sigma_4)$ является двоичным эквивалентом номера выходной функции.

Полагая, что на входах дешифратора вместе с переменными x_i имеются и их отрицания $\overline{x_i}$, представим формулу (A) в виде оператора диодной схемы. Мы будем иметь:

$$z_j = D_k [x_1^{\sigma_1}, x_2^{\sigma_2}, x_3^{\sigma_3}, x_4^{\sigma_4}],$$

где $j = 0, 1, \dots, 15$.

Схема дешифратора, построенная по формуле (A), представлена на рис. 6.6. Общее число диодов, необходимых для реализации матричной схемы дешифратора при m входных переменных, определяется соотношением:

$$N = m \cdot 2^m. \quad (6.10)$$

В нашем примере число диодов $N = 64$.

Рассмотренные приемы синтеза истинностных логических сетей с помощью операторов электронно-ламповых элементов без каких-либо изменений могут быть применены для синтеза схем (на этих элементах), реализующих ограниченно-детерминированные операторы. В последнем случае необходимый вес синтезируемого ограниченно-детерминированного оператора обеспечивается включением в схему соответствующего числа элементов задержки.

Пример 4. Построить электронно-ламповую схему, реализующую ограниченно-детерминированный опера-

тор, заданный деревом, изображенным на рис. 6.7 *). Здесь буквам 0 и 1 входного алфавита сопоставлены

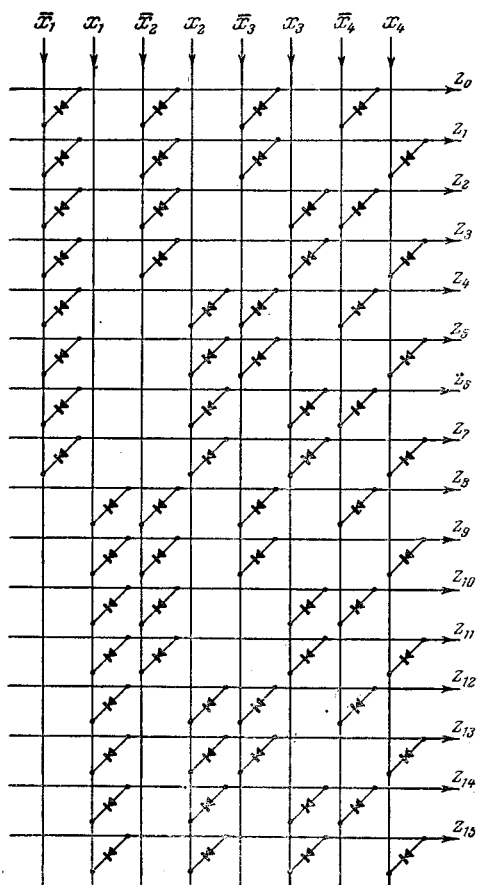


Рис. 6.6.

соответственно левое и правое ребра в дереве. Ребрам приписаны вырабатываемые оператором буквы двузначного алфавита.

*) Пунктир, соединяющий на схеме две какие-либо вершины дерева, указывает на то, что эти вершины неразличимы, т. е. являются вершинами совпадающих ветвей.

Нетрудно заметить, что это дерево задает ограниченно-детерминированный оператор, перерабатывающий последовательность единиц в выходную последовательность: 100100100... Подача на вход нуля не изменяет состояния схемы. Из рассмотрения дерева видно также, что вес этого оператора равен трем.

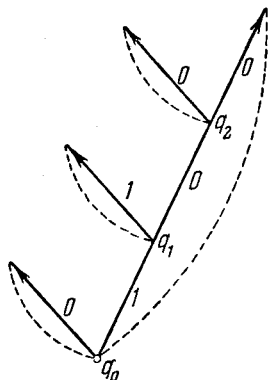


Рис. 6.7.

Припишем вершинам попарно-различимых ветвей различные буквы алфавита состояния q_0 , q_1 и q_2 , которые представим двузначными кодовыми группами $(\varphi_1 \varphi_2)$:

$$q_0 = 00, \quad q_1 = 01, \quad q_2 = 10.$$

По дереву можно составить таблицу VI. 1. Строки таблицы VI. 1, для которых одновременно $\varphi_1(t) = 1$ и $\varphi_2(t) = 1$, дополнены с таким расчетом, чтобы упростить функцию $\varphi_1(t+1)$. Из этой таблицы мы получаем канонические уравнения ограниченно-детерминированного оператора, заданного деревом, изображенным на рис. 6.7, в следующем виде:

$$\left. \begin{aligned} z(t) &= \bar{\varphi}_1(t) \bar{\varphi}_2(t) x(t) \vee \bar{\varphi}_1(t) \varphi_2(t) \bar{x}(t), \\ \varphi_1(t+1) &= \varphi_1(t) \bar{x}(t) \vee \varphi_2(t) x(t), \\ \varphi_2(t+1) &= z(t). \end{aligned} \right\} \quad (B)$$

Схему логической сети, реализующую этот оператор, мы будем строить для каждого из выходов независимо. Для этого предварительно преобразуем формулы (B) в суперпозицию операторов элементов, которыми мы будем пользоваться. Мы получим:

$$\left. \begin{aligned} z(t) &= \bar{\varphi}_1(t) [\bar{x}(t) \vee \bar{\varphi}_2(t)] [x(t) \vee \varphi_2(t)] = \\ &= T[\varphi_1(t)] \& \Pi[x(t), \varphi_2(t)] \& \Pi\{\bar{x}(t), T[\varphi_2(t)]\}, \\ \varphi_1(t+1) &= [\varphi_1(t) \vee x(t)] [\varphi_2(t) \vee \bar{x}(t)] = \\ &= \Pi\{\bar{x}(t), T[\varphi_1(t)]\} \& \Pi\{x(t), T[\varphi_2(t)]\}, \\ \varphi_2(t+1) &= z(t). \end{aligned} \right\} \quad (C)$$

Таблица VI.1

$\Phi_1(t)$	$\Phi_2(t)$	$x(t)$	$\Phi_1(t+1)$	$\Phi_2(t+1)$	$z(t)$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	1	0	0
1	1	1	1	0	0

На рис. 6.8 представлена схема, построенная по формулам (С). Она включает четыре пентода, три триода и два одноктактных элемента задержки \mathcal{Z} .

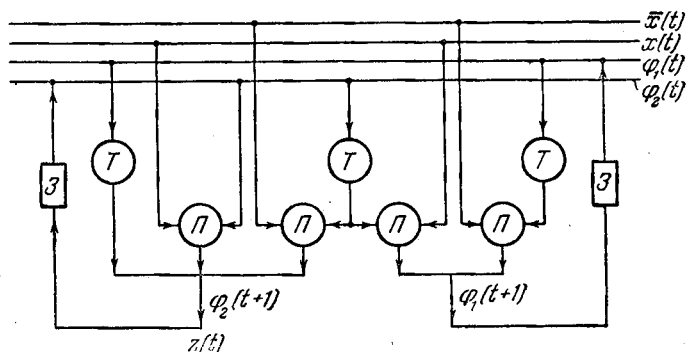


Рис. 6.8.

Перейдем теперь к рассмотрению практических приемов синтеза схем, реализующих ограниченно-детерминированные операторы с помощью триггерных элементов.

Пусть синтезируемый ограниченно-детерминированный оператор задан своими каноническими уравнениями:

$$z(t) = \Phi[x_1(t), x_2(t), \dots, x_j(t), \dots, x_m(t), \phi_1(t), \dots, \phi_i(t), \dots, \phi_s(t)], \quad (6.11)$$

Аналогично, путем сравнения друг с другом формул (6.13) и (3.23), легко определяется функция переключения j -го одновходного триггера:

$$\begin{aligned} v_j(t) = & \bar{\varphi}_j(t) \Psi_j[x_1(t), \dots, x_i(t), \dots, x_m(t), \varphi_1(t), \dots \\ & \dots, \varphi_{j-1}(t), 0, \varphi_{j+1}(t), \dots, \varphi_s(t)] \vee \varphi_j(t) \bar{\Psi}_j[x_1(t), \dots \\ & \dots, x_i(t), \dots, x_m(t), \varphi_{j-1}(t), 1, \varphi_{j+1}(t), \dots, \varphi_s(t)]. \end{aligned} \quad (6.15)$$

Полученные соотношения определяют порядок построения и схему коммутации логической сети ограниченно-детерминированного оператора веса K посредством суперпозиции операторов двухвходной или одновходной триггерной ячейки. Схема включает $s = \lg_2 K$ триггеров и истинностные элементы, необходимые для образования выходной функции $z(t)$ и функций (6.14), (6.15), обеспечивающих связи между триггерами.

Пример 5. Построить триггерный одноразрядный двоичный счетчик с управлением от синхронизирующего импульса. Выходная переменная $z(t)$ равна 1 при выполнении следующих условий:

1) если в момент t подан синхронизирующий импульс x_c ;

2) если ему предшествовало нечетное число насыщенных тактов, то есть таких тактов, в которых на вход схемы был подан хотя бы один рабочий импульс x_p .

Схема управляется импульсами положительной полярности; одновременная подача рабочего и синхронизирующего импульсов не допускается.

Схема, удовлетворяющая указанным условиям, должна обладать не более чем четырьмя состояниями. На рис. 6.9 дано дерево синтезируемого ограниченно-детерминированного оператора. Входными

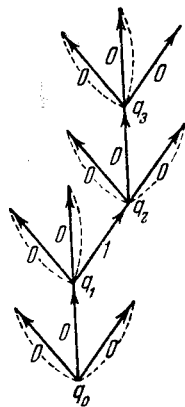


Рис. 6.9.

буквами являются наборы: $\bar{x}_p \bar{x}_c$, $\bar{x}_p x_c$, $x_p \bar{x}_c$, $x_p x_c$, которым сопоставлены соответственно левое, среднее и правое ребра.

Для кодирования букв состояния будем пользоваться двузначными наборами $(\varphi_{A1}, \varphi_{A2})$, которые для упрощения

будем записывать в виде $(\varphi_1 \varphi_2)$. Для букв q_i введем следующие коды:

$$q_0 = 00, \quad q_1 = 10, \quad q_2 = 11, \quad q_3 = 01.$$

Ниже приведена функциональная таблица (табл. VI.2), построенная по дереву рассматриваемого ограниченно-детерминированного оператора. В строчках, для которых

Таблица VI.2

$\varphi_1(t)$	$\varphi_2(t)$	$x_p(t)$	$x_c(t)$	$\varphi_1(t+1)$	$\varphi_2(t+1)$	$z(t)$
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	0	0	0
0	1	1	0	0	1	0
0	1	1	1	0	0	0
1	0	0	0	1	0	0
1	0	0	1	1	1	1
1	0	1	0	1	0	0
1	0	1	1	1	1	1
1	1	0	0	1	1	0
1	1	0	1	1	1	0
1	1	1	0	0	1	0
1	1	1	1	1	1	0

одновременно $x_c=1$, $x_p=1$, таблица дополнена с таким расчетом, чтобы получилось более простое аналитическое представление оператора.

Из таблицы после некоторой процедуры минимизации мы получаем следующие канонические уравнения нашего оператора:

$$\varphi_1(t+1) = \varphi_1(t) \bar{\varphi}_2(t) \vee \varphi_1(t) \bar{x}_p(t) \vee \bar{\varphi}_1(t) \bar{\varphi}_2(t) x_p(t), \quad (D)$$

$$\varphi_2(t+1) = \varphi_2(t) \varphi_1(t) \vee \varphi_2(t) \bar{x}_c(t) \vee \bar{\varphi}_2(t) \varphi_1(t) x_c(t), \quad (E)$$

$$z(t) = \varphi_1(t) \bar{\varphi}_2(t) x_c(t). \quad (F)$$

Схему логической сети, которая реализует уравнения (Г) и (Д) и работает от положительных импульсов $x_p(t)$ и

$x_c(t)$, мы построим на триггерах $Tr\ 1$ и $Tr\ 2$ с «тянущими» лампами, независимо для каждого из ее выходов. Имея в виду уравнения (6.14) и (3.27) и принимая

$$\begin{aligned}v'_1(t) &= \bar{\varphi}_2(t) x_p(t), \\v''_1(t) &= \bar{\varphi}_2(t) \vee \bar{x}_p(t),\end{aligned}$$

получаем следующее соотношение, определяющее схему включения триггера 1 с пентодами на входе:

$$\begin{aligned}\varphi_1(t+1) &= [\bar{\varphi}_2(t) \vee \bar{x}_p(t)] \varphi_1(t) \vee [\varphi_2(t) \vee \bar{x}_p(t)] \bar{\varphi}_1(t) = \\&= \Pi[\varphi_2(t), x_p(t)] \varphi_1(t) \vee \Pi[\bar{\varphi}_2(t), \bar{x}_p(t)] \bar{\varphi}_1(t).\end{aligned}$$

Аналогичным образом, полагая

$$\begin{aligned}v'_2(t) &= \varphi_1(t) x_c(t), \\v''_2(t) &= \varphi_1(t) \vee \bar{x}_c(t),\end{aligned}$$

мы получаем, что

$$\begin{aligned}\varphi_2(t+1) &= [\varphi_1(t) \vee \bar{x}_c(t)] \varphi_2(t) \vee [\bar{\varphi}_1(t) \vee \bar{x}_c(t)] \bar{\varphi}_2(t) = \\&= \Pi[\varphi_1(t), x_c(t)] \varphi_2(t) \vee \Pi[\bar{\varphi}_1(t), \bar{x}_c(t)] \bar{\varphi}_2(t).\end{aligned}$$

Последнее соотношение определяет схему включения триггера 2. На рис. 6.10 дана схема двоичного одnorазрядного счетчика с двумя триггерами, построенная на основе полученных соотношений.

На рис. 6.11 приведена схема S -разрядного двоичного счетчика, составленного из однотипных триггерных ячеек рассмотренного типа. При построении этой схемы было принято во внимание, что поступление рабочих импульсов на вход триггерной ячейки второго разряда должно происходить с частотой, вдвое меньшей частоты поступления сигналов $x_p(t)$. Из предыдущего изложения видно, что $\varphi_2(t)$ обращается в единицу при каждом нечетном насыщенном такте (после поступления соответствующего импульса x_c). Следовательно, указанное выше условие будет выполнено, если

$$x_p^{(2)}(t) = x_p(t) \varphi_2^{(1)}(t),$$

где верхние индексы 1, 2 указывают номера разрядов (начиная с низшего). Аналогичные зависимости имеют

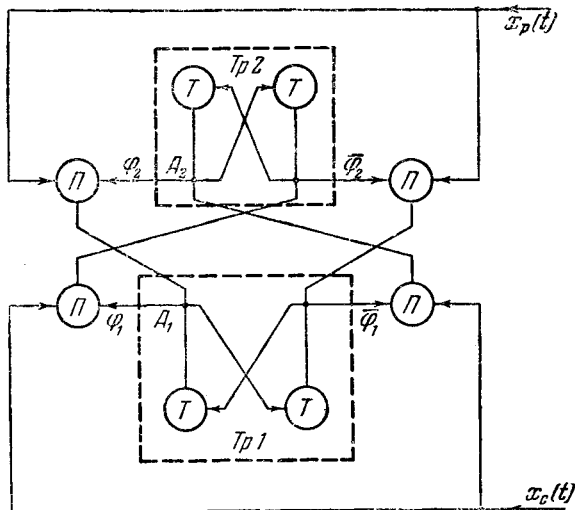


Рис. 6.10.

место и для последующих разрядов, то есть:

$$x_p^{(n)}(t) = x_p(t) \varphi_2^{(n-1)}(t) = T_2[\bar{x}_p(t), \bar{\varphi}_2^{(n-1)}(t)].$$

Полученное соотношение определяет схему связей между триггерными ячейками соседних разрядов.

Пример 6. Построить триггерную схему последовательного действия для сравнения двух n -разрядных двоичных чисел:

$$u = u_1 u_2 u_3 \dots u_n \quad \text{и} \quad w = w_1 w_2 w_3 \dots w_n.$$

Условия работы схемы:

1) в каждом такте t на ее вход поступают одноименные разряды обоих чисел:

$$u(t), w(t) \rightarrow (0, 0), (0, 1), (1, 0), (1, 1);$$

2) схема сравнивает последовательности:

$$u(1) \ u(2) \ \dots \ u(t-2) \ u(t-1) \ u(t), \quad (1)$$

$$w(1) \ w(2) \ \dots \ w(t-2) \ w(t-1) \ w(t) \quad (2)$$

и при их равенстве вырабатывает на выходе сигнал γ . Если последовательность (1) больше последовательности (2), на выходе вырабатывается сигнал α ; если же имеет место обратное соотношение, то вырабатывается сигнал β .

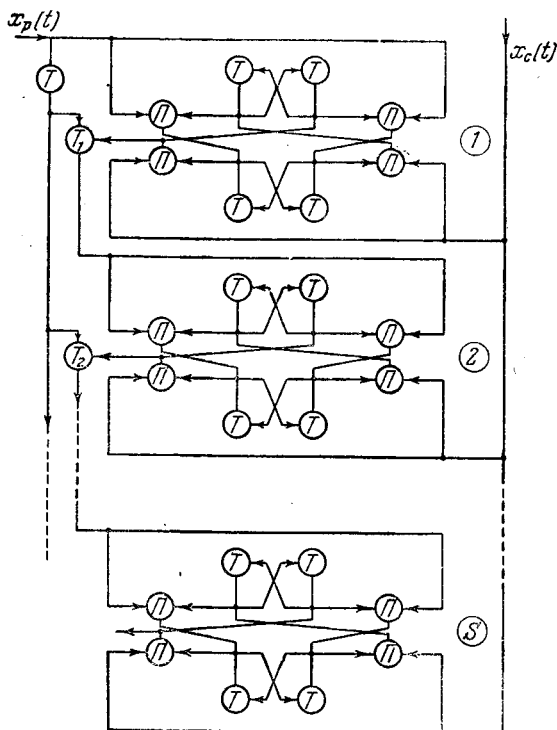


Рис. 6.11.

Введем входной алфавит из четырех букв a, b, c, d , каждая из которых закодирована двузначной кодовой группой $(x_1 x_2)$ из нулей и единиц; иначе говоря:

$$a = 00, \quad b = 01, \quad c = 10, \quad d = 11.$$

Для кодирования выходного алфавита применим также двузначные кодовые группы $(z_1 z_2)$, причем пусть будет

$$\alpha = 10, \quad \beta = 01, \quad \gamma = 00.$$

Построим дерево соответствующего ограниченно-детерминированного оператора, сопоставляя ребрам буквы

входного алфавита в порядке, указанном на рис. 6.12, а. Дерево представлено на рис. 6.12, б и имеет три попарно-различимые ветви. Следовательно, вес оператора равен трем, и алфавит состояний включает три буквы с кодовыми группами $(\varphi_1 \varphi_2)$:

$$q_0 = 00,$$

$$q_1 = 01,$$

$$q_2 = 10.$$

Используя для образования выходных сигналов статические выходы триггеров, то есть принимая, что

$$z_1 = \varphi_1$$

и

$$z_2 = \varphi_2,$$

построим по данному дереву функциональную таблицу (табл. VI.3) для ограниченно-детерминированного оператора, подлежащего реализации (строки, для которых одновременно $\varphi_1(t) = 1$ и $\varphi_2(t) = 1$, в

таблицу не включены, так как в рассматриваемом случае они доопределены нулями).

Из таблицы VI.3 получаем канонические уравнения рассматриваемого нами ограниченно-детерминированного оператора в следующей дизъюнктивной нормальной

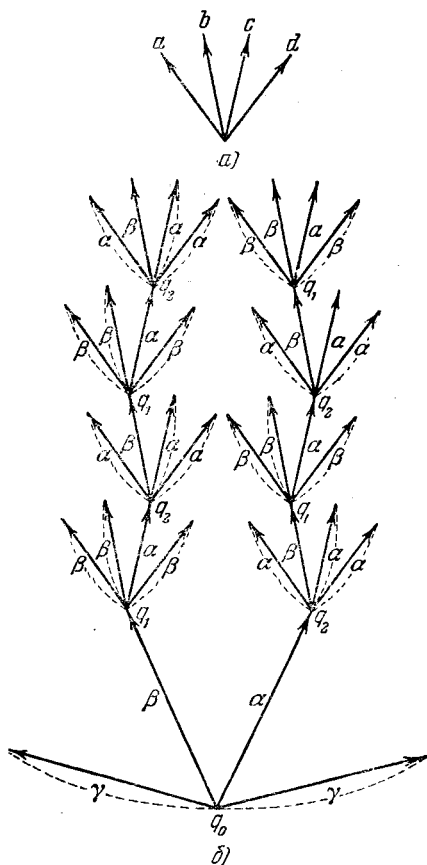


Рис. 6.12.

Таблица VI.3

$\varphi_1(t)$	$\varphi_2(t)$	$x_1(t)$	$x_2(t)$	$\varphi_1(t+1)$	$\varphi_2(t+1)$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	0

форме:

$$\left. \begin{aligned} \varphi_1(t+1) &= \bar{\varphi}_1(t) x_1(t) \bar{x}_2(t) \vee \varphi_1(t) \bar{\varphi}_2(t) \bar{x}_2(t) \vee \\ &\quad \vee \varphi_1(t) \bar{\varphi}_2(t) x_1(t), \\ \varphi_2(t+1) &= \bar{\varphi}_2(t) \bar{x}_1(t) x_2(t) \vee \bar{\varphi}_1(t) \varphi_2(t) \bar{x}_1(t) \vee \\ &\quad \vee \bar{\varphi}_1(t) \varphi_2(t) x_2(t). \end{aligned} \right\} \quad (A^*)$$

Схему для их реализации мы построим на двухвходных триггерах; входные сигналы будем задавать импульсами положительной полярности. Разложим каждую из формул (A*) по переменным φ_j и примем:

$$\left. \begin{aligned} v'_1(t) &= \bar{\varphi}_2(t) [x_1(t) \vee \bar{x}_2(t)] = \\ &= T_2 \{ \varphi_2(t), T_2 [x_1(t), x_2(t)] \}, \\ v''_1(t) &= T \{ T_2 [\bar{x}_1(t), x_2(t)] \}. \end{aligned} \right\} \quad (B^*)$$

Аналогично определяются

$$\left. \begin{aligned} v'_2 &= T_2 \{ \varphi_1(t), T_2 [\bar{x}_1(t), x_2(t)] \}, \\ v''_2 &= T \{ T_2 [x_1(t), \bar{x}_2(t)] \}. \end{aligned} \right\} \quad (C^*)$$

Схема, моделирующая полученные формулы (A*), (B*) и (C*), представлена на рис. 6.13.

При синтезе триггерных схем, составленных из однотипных ячеек, удобно пользоваться для задания соответствующего ограниченно-детерминированного оператора с фиксированным весом матрицами переходов для состояний и выходных букв, предложенными

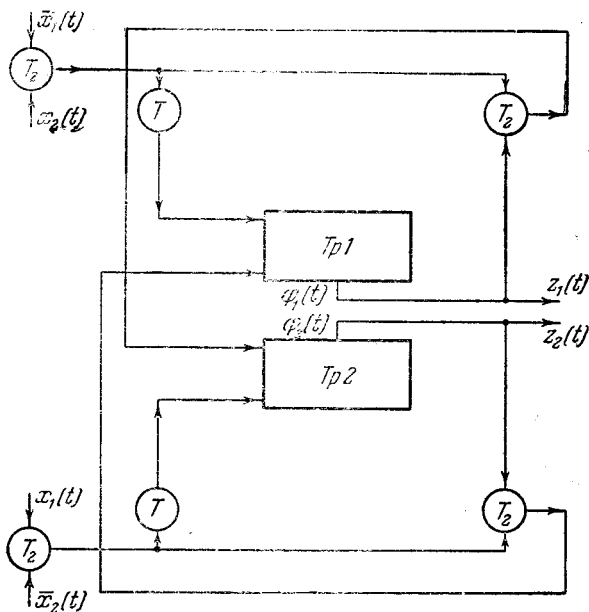


Рис. 6.13.

М. Л. Цетлиным*). При этом канонические уравнения оператора могут быть получены непосредственно из матриц (см. § 3 гл. V).

Пример 7**). Построить схему S -разрядного дифференциального бинарного счетчика с 2^S состояниями $(0, 1, 2, 3, \dots, 2^S - 1)$, имеющего два двоичных входных канала (x_1, x_2) и два двоичных выходных канала (u, w) .

*) М. Л. Цетлин, О непримитивных схемах, Сб. «Проблемы кибернетики», № 1, Физматгиз, М., 1958.

**) Пример заимствован из указанной выше статьи М. Л. Цетлина.

Условия работы схемы таковы:

1) при $x_1(t) = x_2(t) = 1$ или $x_1(t) = x_2(t) = 0$ состояние схемы не меняется;

2) при $x_1(t) = 1$ и $x_2(t) = 0$ схема переходит из состояния номер j в состояние номер $j+1$ (по mod 2^S); для $j = 2^S - 1$ состояние $j+1 = 0$;

3) при $x_1(t) = 0$ и $x_2(t) = 1$ схема переходит из состояния номер j в состояние номер $j-1$ (по mod 2^S); для $j = 0$ состояние $j-1 = 2^S - 1$;

4) при переходе схемы из состояния $2^S - 1$ в состояние 0 вырабатывается сигнал $u = 1$; при обратном переходе возникает сигнал $w = 1$. Во всех остальных случаях $u = w = 0$.

Построим вначале схему счетчика для $S=2$. Примем для кодирования десятичных чисел, определяющих номера состояний (q), кодовые группы ($\varphi_1\varphi_2$), выражающие десятичные числа в их двоичном эквиваленте, и составим матрицы перехода состояний и выходных букв.

Матрица перехода состояний представлена в таблице VI.4.

Таблица VI.4

$q(t+1) \backslash q(t)$	00	01	10	11
00	00, 11	10	—	01
01	01	00, 11	10	—
10	—	01	00, 11	10
11	10	—	01	00, 11

Матрица перехода для выходных букв представлена в таблице VI.5.

Таблица VI.5

$u, w \backslash x_1x_2$	00	01	10	11
00	00, 01, 10, 11	—	—	—
01	01, 10, 11	00	—	—
10	00, 01, 10	—	11	—
11	00, 01, 10, 11	—	—	—

Из матрицы перехода состояний мы после упрощений получаем следующие формулы для $\varphi_1(t+1)$ и $\varphi_2(t+1)$:

$$\left. \begin{aligned} \varphi_1(t+1) &= \overline{\varphi_1(t)} \overline{\varphi_2(t)} \overline{x_1(t)} x_2(t) \vee \\ &\vee \overline{\varphi_1(t)} \varphi_2(t) x_1(t) \overline{x_2(t)} \vee \\ &\vee \varphi_1(t) \overline{\varphi_2(t)} \overline{x_1(t)} x_2(t) \vee \\ &\vee \varphi_1(t) \varphi_2(t) \overline{x_1(t)} \overline{x_2(t)}, \\ \varphi_2(t+1) &= \overline{\varphi_2(t)} [\overline{x_1(t)} x_2(t) \vee x_1(t) \overline{x_2(t)}] \vee \\ &\vee \varphi_2(t) [\overline{x_1(t)} \overline{x_2(t)} \vee x_1(t) x_2(t)]. \end{aligned} \right\} (A^{**})$$

При реализации бинарного счетчика на одноходовых триггерных ячейках, управляемых импульсами отрицательной полярности, функции переключения триггеров, на основании (6.14), выражаются так:

$$\left. \begin{aligned} v_1(t) &= x_1(t) \overline{x_2(t)} \varphi_2(t) \vee \overline{x_1(t)} x_2(t) \varphi_2(t), \\ v_2(t) &= \overline{x_1(t)} x_2(t) \vee x_1(t) \overline{x_2(t)}. \end{aligned} \right\} (B^{**})$$

Найдем теперь соотношения, определяющие динамические выходы триггерных ячеек. Для первой ячейки они определяются так:

$$\left. \begin{aligned} z_{1A}(t) &= v_1(t) \varphi_1(t) = x_1(t) \overline{x_2(t)} \varphi_1(t) \varphi_2(t) \vee \\ &\vee \overline{x_1(t)} x_2(t) \varphi_1(t) \overline{\varphi_2(t)}, \\ z_{1B}(t) &= v_1(t) \overline{\varphi_1(t)} = x_1(t) \overline{x_2(t)} \overline{\varphi_1(t)} \overline{\varphi_2(t)} \vee \\ &\vee \overline{x_1(t)} x_2(t) \overline{\varphi_1(t)} \overline{\varphi_2(t)}. \end{aligned} \right\} (C^{**})$$

Аналогичным путем мы находим выходные функции второй ячейки:

$$\left. \begin{aligned} z_{2A}(t) &= v_2(t) \varphi_2(t) = \\ &= \overline{x_1(t)} x_2(t) \varphi_2(t) \vee x_1(t) \overline{x_2(t)} \varphi_2(t), \\ z_{2B}(t) &= v_2(t) \overline{\varphi_2(t)} = \\ &= \overline{x_1(t)} x_2(t) \overline{\varphi_2(t)} \vee x_1(t) \overline{x_2(t)} \overline{\varphi_2(t)}. \end{aligned} \right\} (D^{**})$$

Из (В**) и (С**) легко установить, что функция переключения первой ячейки определяется следующей ее зависимостью от выходов второй ячейки:

$$v_1(t) = z_{2A}(t) x_1(t) \vee z_{2B}(t) x_2(t). \quad (\text{E}^{**})$$

Из матрицы выходных букв мы находим:

$$\left. \begin{aligned} u(t) &= x_1(t) \bar{x}_2(t) \varphi_1(t) \varphi_2(t) = z_{1A}(t) x_1(t), \\ \omega(t) &= \bar{x}_1(t) x_2(t) \bar{\varphi}_1(t) \bar{\varphi}_2(t) = z_{1B}(t) x_2(t). \end{aligned} \right\} \quad (\text{F}^{**})$$

Формулы, моделирующие схему S -разрядного бинарного счетчика, могут быть получены путем обобщения зависимостей (А**—F**). Изменяя, для упрощения записи, порядок нумерации триггерных ячеек, выпишем эти формулы в следующем виде:

$$\begin{aligned} v_1(t) &= x_1(t) \bar{x}_2(t) \vee \bar{x}_1(t) x_2(t), \\ v_j(t) &= x_1(t) \bar{x}_2(t) \varphi_1(t) \varphi_2(t) \dots \varphi_{j-1}(t) \vee \\ &\vee \bar{x}_1(t) x_2(t) \varphi_1(t) \varphi_2(t) \dots \bar{\varphi}_{j-1}(t), \quad (j = 2, 3, \dots, S) \\ z_{Aj}(t) &= v_j(t) \varphi_j(t), \\ z_{Bj}(t) &= v_j(t) \bar{\varphi}_j(t), \\ v_{j+1}(t) &= z_{Aj}(t) x_1(t) \vee z_{Bj}(t) x_2(t), \\ u(t) &= z_{AS}(t) x_1(t), \\ \omega(t) &= z_{BS}(t) x_2(t). \end{aligned}$$

На рис. 6.14 изображена схема S -разрядного бинарного счетчика, построенная в соответствии с написанными формулами. В этой схеме низший разряд двоичного числа фиксируется в крайней левой ячейке (триггер 1).

Рассмотрим теперь приемы синтеза логических сетей с применением ферромагнитных трансформаторных элементов и двухпериодной организацией считывания.

Будем, как и ранее, полагать, что канонические уравнения синтезируемого оператора заданы в дизъюнктивной нормальной форме. При построении схемы, соответствующей синтезируемой логической сети, с помощью ферромагнитных элементов возникает необходимость преобразования канонических уравнений в сочетание операторов этих элементов. Примем во внимание замечание, сделанное

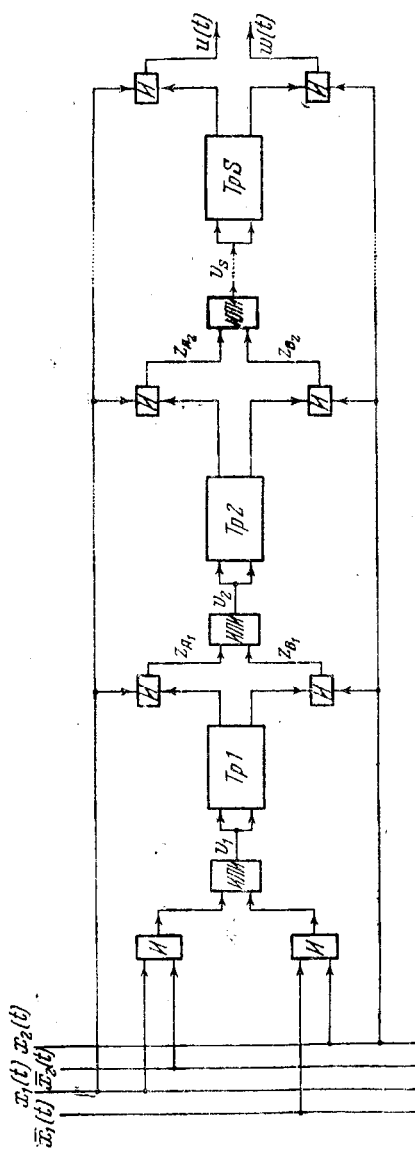


Рис. 6.14.

в третьей главе, о возможности реализации совершенной дизъюнктивной нормальной формы в масштабе времени $\sigma = 2$, и выразим ее, следуя М. Л. Цетлину и Л. М. Шехтману*), через операторы ферромагнитных элементов в следующем виде:

$$z(t+2) = f[x_1(t), x_2(t), \dots, x_m(t)] = \\ = \bigvee_{\sigma_1, \sigma_2, \dots, \sigma_m} f(\sigma_1, \sigma_2, \dots, \sigma_m) G \left\{ 1, \bigvee_{i=1}^m \bar{\sigma}_i H(x_i(t), 0) \vee \right. \\ \left. \vee \sigma_i H(1, x_i(t)) \right\}. \quad (6.16)$$

Формулы (6.16) мы будем называть нормальной G -формой представления функции f . Эта форма определяет схему коммутации элементов при построении логической сети по функции f . Соотношения, аналогичные (6.16), могут быть получены без особых затруднений для упрощенных дизъюнктивных нормальных формул канонических уравнений.

Пример 8. Построить на ферромагнитных элементах схему последовательного действия для сравнения n -разрядных двоичных чисел, работающую в режиме «равно-неравно». В отличие от схемы, рассмотренной в примере 5, здесь выходной канал должен возбуждаться только в случае неравенства сравниваемых двоичных чисел.

Нетрудно установить, что вес оператора, реализующего указанное преобразование, равен двум. Примем для кодирования входных букв двузначные кодовые группы

Таблица VI.6

$q(t+2) \backslash q(t)$	0	1
0	00, 11	01, 10
1	—	00, 01, 10 11

*) М. Л. Цетлин и Л. М. Шехтман, Двухтактные ферротранзисторные схемы и алгебраический метод их синтеза, Сб. «Проблемы кибернетики», № 2, Физматгиз, М., 1959.

примера 5 и составим для синтезируемого оператора матрицу переходов состояний в виде табл. VI. 6 (стр. 309).

По матрице таблицы VI.6 получаем каноническое уравнение синтезируемого ограниченно-детерминированного оператора в следующем виде:

$$z(t+2) = q(t+2) = q(t) \vee [x_1(t) \bar{x}_2(t) \vee \bar{x}_1(t) x_2(t)] \bar{q}(t),$$

или, после упрощения:

$$z(t+2) = q(t+2) = q(t) \vee x_1(t) \bar{x}_2(t) \vee \bar{x}_1(t) x_2(t). \quad (A')$$

Преобразуем формулу (A') применительно к сочетанию операторов ферромагнитных элементов. При этом примем во внимание, что сигналы, поступающие на обмотки возбуждения, с одной стороны, и на размагничивающие обмотки, с другой стороны, должны иметь различную длительность. В силу этого условия при реализации сети не допускается подключение каналов, питающих оба типа обмоток, к одним и тем же полюсам.

Учитывая сделанные замечания, представим (A') в следующем виде (для упрощения записи здесь и в дальнейшем мы часто будем опускать символ (t) в формулах и на рисунках):

$$z(t+2) = G\{0, [G(q, 0) \vee G(x_1, x_2)]\} \vee \\ \vee G\{1, [H(x_1, 0) \vee H(1, x_2)]\}.$$

На рис. 6.15 представлена схема, моделирующая написанную выше формулу. Здесь переменная $x_1(t)$ подается только на возбуждающие обмотки, а переменная $x_2(t)$ — на размагничивающие. Для реализации схемы требуется шесть сердечников — в соответствии с общим числом символов G и H в формуле.

В рассмотренном примере представлению канонического уравнения в виде сочетания элементарных операторов предшествовало упрощение его нормальной формы. В некоторых случаях более экономное преобразование реализуемого оператора, ориентированное на представление в виде сочетания операторов ферромагнитных элементов, достигается путем тождественных преобразований

непосредственно нормальной G -формы. Ниже приводятся полезные для этой цели формулы преобразований, взятые из упоминавшейся выше работы М. Л. Цетлина и Л. М. Шехтмана.

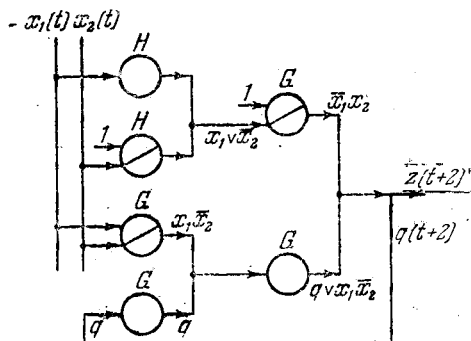


Рис. 6.15.

Из соотношений (3.39) непосредственно следует, что

$$\left. \begin{aligned} G(x, x) &= 0, \\ \bigvee_i G(x_i, 0) &= G\left(\bigvee_i x_i, 0\right). \end{aligned} \right\} \quad (6.17)$$

Нетрудно показать также справедливость следующих тождеств:

$$G[1, (H(x, y) \vee F)] = G[(G(1, x) \vee G(y, 0)), F], \quad (6.18)$$

$$G[G(x, y), F] = G[1, (H(1, x) \vee H(y, 0) \vee F)], \quad (6.18')$$

где F — произвольное выражение.

В тех случаях, когда функция $f(x_1, x_2, \dots, x_m)$ не зависит существенно от некоторых аргументов, упрощение схемы может быть достигнуто на основании следующего тождества:

$$\bigvee_{\sigma_1, \sigma_2, \dots, \sigma_n} G\left\{F, P \vee \bigvee_{i=1}^n [\sigma_i H(1, x_i) \vee \bar{\sigma}_i H(x_i, 0)]\right\} = G(F, P), \quad (6.19)$$

где F и P не зависят от x_1, x_2, \dots, x_n . Так, например, для функции $f(x_1, x_2, x_3) = \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3$ G -нормальная форма выражается так:

$$f(x_1, x_2, x_3) = G[1, (H(x_1, 0) \vee H(1, x_2) \vee H(1, x_3))] \vee \\ \vee G[1, (H(1, x_1) \vee H(1, x_2) \vee H(1, x_3))].$$

В соответствии с (6.19) она может быть преобразована в вид:

$$f(x_1, x_2, x_3) = G[1, (H(1, x_2) \vee H(1, x_3))].$$

Часто встречаются также функции, которые зависят от дизъюнкции некоторых своих аргументов. Здесь упрощение достигается на основании соотношения:

$$\bigvee_{i=1}^n G[F, (P \vee H(1, x_i))] = G[F, (P \vee H(1, \bigvee_{i=1}^n x_i))]. \quad (6.20)$$

В частном случае, при $n=2$, это соотношение принимает следующий вид:

$$G[F, (P \vee H(1, x_1))] \vee G[F, (P \vee H(1, x_2))] = \\ = G[F, (P \vee H(1, x_1 \vee x_2))]. \quad (6.21)$$

Например, функция $f = x_1 x_2 x_3 \vee x_2 x_3 x_4$, для которой G -нормальная форма имеет вид:

$$f = G[1, (H(1, x_1) \vee H(1, x_2) \vee H(1, x_3))] \vee \\ \vee G[1, (H(1, x_2) \vee H(1, x_3) \vee H(1, x_4))],$$

на основании (6.21) приводится к форме:

$$f = G[1, (H(1, x_2) \vee H(1, x_3) \vee H(1, x_1 \vee x_4))].$$

Пример 9. Построить логическую сеть на ферритах, эквивалентную одноходовому триггеру.

Воспользуемся уравнениями (3.25) и (3.26) и логическими операторами элементарных схем на ферритовых сердечниках; примем, далее, во внимание сделанные выше замечания о допустимом порядке объединения внешних каналов сети. Тогда мы будем иметь:

$$v(t) \bar{\varphi}(t) = G\{1, [H(1, v) \vee H(\varphi, 0)]\},$$

$$\bar{v}(t) \varphi(t) = G\{1, [H(v, 0) \vee H(1, \varphi)]\}.$$

Функция, определяющая состояние схемы, имеет вид

$$\varphi(t+2) = G\{1, [H(1, v) \vee H(\varphi, 0)]\} \vee G\{1, [H(v, 0) \vee H(1, \varphi)]\}.$$

Для ее реализации в этом виде необходимо 6 сердечников.

Преобразуя ее с помощью формулы (6.18'), мы получим более простую формулу:

$$\varphi(t+2) = G\{1, [H(1, v) \vee H(\varphi, 0)]\} \vee G[G(\varphi, v), 0]. \quad (A'')$$

Схема, моделирующая эту формулу, требует 5 сердечников.

Выходная функция, аналогичная динамическому выходу триггера, выразится так:

$$\begin{aligned} z(t+2) &= v(t) \varphi(t) = \\ &= G\{1, [H(1, \varphi) \vee H(1, v)]\}. \end{aligned} \quad (B'')$$

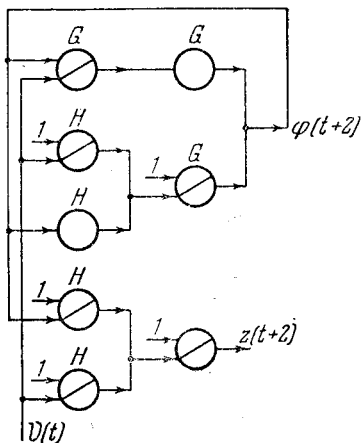


Рис. 6.16.

Схема, построенная в соответствии с формулами (A'') и (B''), дана на рис. 6.16. Она реализована на 8 сердечниках.

Пример 10. Построить ферромагнитную схему двоичного сумматора последовательного действия, работающего в масштабе времени $\sigma=2$. Условия работы сумматора заданы деревом, изображенным на рис. 6.17. На нем ребрам каждого яруса соответствуют буквы входного алфавита a, b, c, d ; выходной алфавит — двоичный.

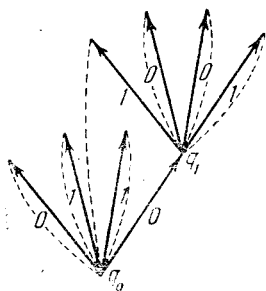


Рис. 6.17.

Закодируем буквы входного алфавита двузначными двоичными группами ($x_1 x_2$):

$$a = 00, \quad b = 01, \quad c = 10, \quad d = 11$$

и сопоставим буквам алфавита состояния двоичные символы $q_0=0$, $q_1=1$. Обозначим выходную функцию через z и построим по дереву, изображенному на рисунке, таблицу оператора, реализуемого сумматором (см. табл. VI.7).

Таблица VI.7

$q(t)$	$x_1(t)$	$x_2(t)$	$q(t+2)$	$z(t+2)$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Из таблицы VI.7 мы получаем канонические уравнения оператора в следующем виде:

$$q(t+2) = \bar{q}(t) x_1(t) x_2(t) \vee q(t) \bar{x}_1(t) x_2(t) \vee \\ \vee q(t) x_1(t) \bar{x}_2(t) \vee q(t) x_1(t) x_2(t) = \\ = x_1(t) x_2(t) \vee q(t) x_1(t) \vee q(t) x_2(t), (A'')$$

$$z(t+2) = \bar{q}(t) \bar{x}_1(t) x_2(t) \vee \bar{q}(t) x_1(t) \bar{x}_2(t) \vee \\ \vee q(t) \bar{x}_1(t) \bar{x}_2(t) \vee q(t) x_1(t) x_2(t). (B'')$$

Формулы (A'') и (B'') не поддаются дальнейшим упрощениям, не выводящим канонических уравнений из дизъюнктивной нормальной формы.

Преобразуем их в сочетание элементарных операторов ферромагнитных элементов. Мы получим:

$$q(t+2) = H[1, (H(1, x_1) \vee H(1, x_2))] \vee \\ \vee H[1, (H(1, x_1) \vee H(1, q))] \vee \\ \vee H[1, (H(1, x_2) \vee H(1, q))], (C'')$$

$$z(t+2) = G[1, (H(q, 0) \vee H(x_1, 0) \vee H(1, x_2))] \vee \\ \vee G[1, (H(q, 0) \vee H(1, x_1) \vee H(x_2, 0))] \vee \\ \vee G[1, (H(1, q) \vee H(x_1, 0) \vee H(x_2, 0))] \vee \\ \vee G[1, (H(1, q) \vee H(1, x_1) \vee H(1, x_2))]. (D'')$$

Схема, моделирующая формулы (C'') и (D''), изображена на рис. 6.18 и содержит 13 сердечников. В этой схеме

выход $z(t+2)$ выдает сумму двоичных слагаемых, а $q(t+2)$ дает перенос в последующий разряд единицы, накопленной в предыдущем разряде.

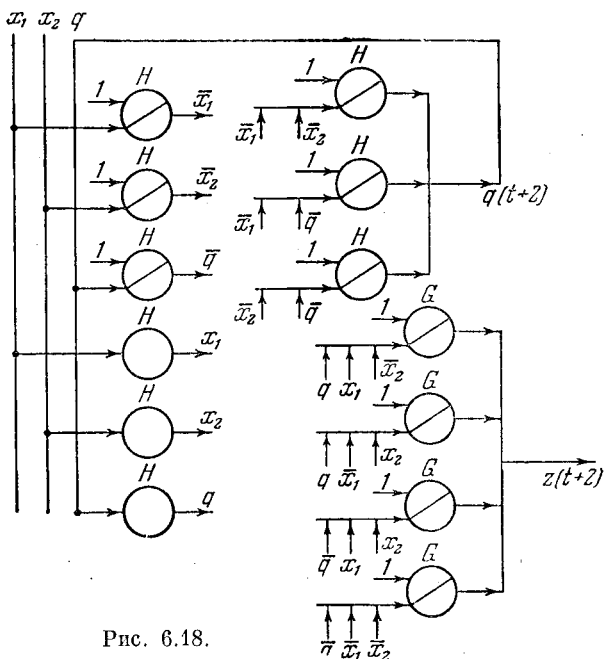


Рис. 6.18.

В заключение параграфа приведем некоторые общие соображения, относящиеся к синтезу сетей на ферродиодных дроссельных элементах. На основании формул, указанных в гл. III, представим совершенную дизъюнктивную нормальную форму с помощью операторов ферродиодных элементов в следующем виде:

$$\begin{aligned}
 z(t+2) &= f[x_1(t), x_2(t), \dots, x_m(t)] = \\
 &= \bigvee_{\sigma_1, \sigma_2, \dots, \sigma_m} f(\sigma_1, \sigma_2, \dots, \sigma_m) R \left\{ \bigvee_{i=1}^m [\bar{\sigma}_i R[1, x_i(t)] \vee \right. \\
 &\quad \left. \vee \sigma_i R[x_i(t), 0], 0] \right\}. \quad (6.22)
 \end{aligned}$$

Формула (6.22) (в которой R есть нормальная форма) определяет схему над множеством ферродиодных элементов,

моделирующую функцию f . Эта формула строится с применением операторов конъюнкции, дизъюнкции и отрицания, и для ее упрощения можно пользоваться уже известными преобразованиями.

Следует, однако, иметь в виду, что с помощью ферродиодного элемента может быть непосредственно реализован оператор импликации. Это позволяет, как показал Л. М. Шехтман*), строить более простые (в смысле числа элементов или в смысле улучшения условий их работы) схемы путем упрощения формы (6.22) на основе следующих преобразований**):

$$R(x, x) = 1, \quad (6.23)$$

$$R[R(x, 0), 0] = R[1, R(1, x)], \quad (6.24)$$

$$\bigvee_i R(x, y_i) = R[x, \bigvee_i y_i], \quad (6.25)$$

$$\begin{aligned} R[(R(x, 0) \vee R(1, y) \vee R(1, S)), 0] = \\ = R[(R(x, y \vee S)), 0], \end{aligned} \quad (6.26)$$

$$R[(S \vee R(x, 0)), 0] \vee R[(S \vee R(1, x)), 0] = R(S, 0), \quad (6.27)$$

где S — произвольное выражение.

Для иллюстрации методики синтеза логической сети с использованием ферродиодных элементов построим схему двоичного сумматора по данным предыдущего примера. Представим формулы (A''') и (B''') в следующей R -нормальной форме:

$$\begin{aligned} q(t+2) &= R[(R(x_1, 0) \vee R(x_2, 0)), 0] \vee \\ &\vee R[(R(q, 0) \vee R(x_1, 0)), 0] \vee R[(R(q, 0) \vee R(x_2, 0)), 0], \quad (E''') \\ z(t+2) &= R[(R(1, q) \vee R(1, x_1) \vee R(x_2, 0)), 0] \vee \\ &\vee R[(R(1, q) \vee R(1, x_2) \vee R(x_1, 0)), 0] \vee \\ &\vee R[(R(q, 0) \vee R(1, x_1) \vee R(1, x_2)), 0] \vee \\ &\vee R[(R(q, 0) \vee R(x_1, 0) \vee R(x_2, 0)), 0]. \quad (F''') \end{aligned}$$

*) Л. М. Шехтман, Алгебраический метод синтеза ферродиодных дроссельных схем, Сб. «Вопросы электроники», серия 12, вып. 18, 1959.

**) Здесь приводятся только некоторые из формул преобразований, указанные в работе Л. М. Шехтмана.

Реализация этих формул требует тринадцати элементов типа R . В рассматриваемом случае преобразование этих формул на основе написанных выше соотношений не уменьшает числа элементов, но улучшает условия работы схемы за счет уменьшения нагрузки на один элемент и уменьшения числа элементов, возбуждаемых «единицей».

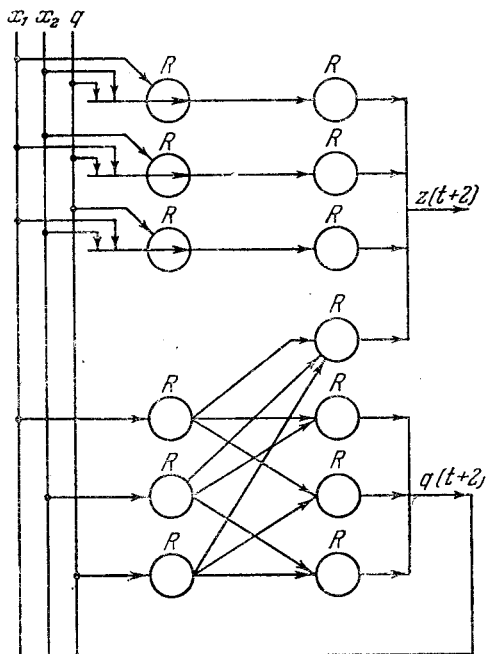


Рис. 6.19.

Используя соотношение (6.26), формулу (F''') можно преобразовать к следующему виду:

$$z(t+2) = R[(R(x_1, x_2 \vee q)), 0] \vee R[(R(q, x_1 \vee x_2)), 0] \vee \\ \vee R[(R(x_2, x_1 \vee q)), 0] \vee \\ \vee R[(R(q, 0) \vee R(x_1, 0) \vee R(x_2, 0)), 0]. \quad (H''')$$

На рис. 6.19 показана схема сумматора, построенная на ферродинамических дроссельных элементах по формулам (E''') и (H''').

§ 3. Синтез многовыходной логической сети

Как мы указывали в § 1, синтез логической сети, реализующей систему n функций алгебры логики, всегда может осуществляться путем построения n «одновыходных» логических сетей, каждая из которых реализует одну функцию. Очень часто, однако, такое решение приводит к неэкономичной схеме сети. Поэтому целесообразно рассмотреть возможности упрощения схемы, возникающие при синтезе сети, непосредственно реализующей систему n функций.

Отметим прежде всего, что упрощение схемы «многовыходной» логической сети может быть иногда достигнуто на основе тщательного рассмотрения набора функций и установления зависимостей между отдельными функциями в наборе. Пусть, например, заданные функции трех переменных имеют следующие выражения:

$$\begin{aligned} z_1 &= \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3, \\ z_2 &= \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3, \\ z_3 &= \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3, \\ z_4 &= \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3, \\ z_5 &= \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3. \end{aligned}$$

Не составляет труда построить пять независимых сетей для образования каждой из указанных функций. При реализации этих сетей с помощью электронно-ламповых элементов потребуется 41 управляющая сетка.

Обратим теперь внимание на то, что для заданного набора справедливы следующие равенства:

$$\begin{aligned} z_3 &= z_2 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3, \\ z_4 &= \bar{z}_5, \\ z_5 &= z_1 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3; \end{aligned}$$

кроме того, в связи с этими соотношениями не требуется образования конъюнкций $\bar{x}_1 \bar{x}_2 x_3$ и $\bar{x}_1 x_2 x_3$. Сеть, построенная с учетом этих зависимостей, требует

всего 22 сетки. Таким образом, достигнута экономия в 19 входных каналов.

Укажем два достаточно общих приема синтеза многовыходной сети, которые во многих случаях обеспечивают ее экономную реализацию. Первый способ практически применим при синтезе сети, имеющей не более чем два или три выхода. Он основан на систематическом использовании построенной выходной функции для получения последующего выхода *).

Предположим, что требуется построить схему для реализации двух функций z_1 и z_2 от m переменных. Построим вначале сеть, реализующую функцию z_1 (рис. 6.20), и используем последнюю в качестве дополнительного входа сети, реализующей функцию z_2 . При этом окажется, что

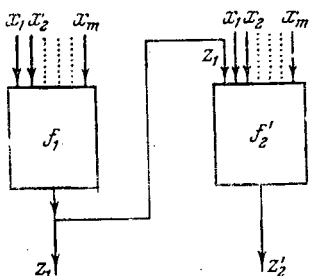


Рис. 6.20.

$$z_2' = f_2'(x_1, x_2, \dots, x_m, z_1) \quad (6.28)$$

становится функцией $m+1$ переменных. Поскольку, однако, z_1 и z_2 зависят от одних и тех же аргументов, совершенная дизъюнктивная нормальная форма для z_2' содержит не более 2^m конъюнкций.

Подчиним выбор соотношения (6.28) следующему условию:

$$f_2'(x_1, x_2, \dots, x_m, z_1) = f_2(x_1, x_2, \dots, x_m). \quad (6.29)$$

Пусть обе функции — z_1 и z_2 — заданы таблицами или формулами, приведенными к совершенной дизъюнктивной нормальной форме. Тогда для того, чтобы добиться выполнения условия (6.28), функцию z_2' следует представить в следующем виде:

$$z_2' = \bigvee_{\sigma_1, \sigma_2, \dots, \sigma_m} f_2'(\sigma_1, \sigma_2, \dots, \sigma_m) (z_1^{\sigma_1} \cdot x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_m^{\sigma_m}), \quad (6.30)$$

*) Синтез электронных вычислительных и управляющих схем, ИИЛ, М., 1954.

причем $\sigma=1$ для тех дизъюнктивных членов, которые входят в z_1 , и $\sigma=0$ для тех дизъюнктивных членов, которые не входят в z_1 . Упрощение функций z_1 и z'_2 и реализация соответствующей логической сети проводятся ранее изложенными способами.

Представляется очевидным, что в рассмотренном случае возможен другой путь построения логической сети. Вначале можно построить функцию z_2 и принять ее в качестве дополнительного входа сети, реализующей функцию z_1 . Таким образом, здесь, как и в других аналогичных случаях, построение наиболее экономной многовыходной логической сети сопряжено с перебором и оценкой различных вариантов, возникающих в связи с возможностью различного «порядка» реализации заданных функций.

Укажем, что в общем случае целесообразно выбирать в качестве исходной такую функцию, которая требует для своей реализации *наименьшее* число элементов или входных каналов, а по ней строить следующую в порядке сложности функцию и т. п. При этом синтезу многовыходной сети должно предшествовать независимое рассмотрение каждой из одновыходных сетей, реализующих заданные функции.

Пример 1. Построить электронно-ламповую схему двухвыходной логической сети, заданной следующими уравнениями:

$$z_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3,$$

$$z_2 = \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3.$$

При синтезе мы будем использовать выход z_1 в качестве дополнительного входа сети, реализующей функцию z_2 . Тогда, в соответствии с (6.30), функция z'_2 будет выражена следующим образом:

$$z'_2 = \bar{z}_1 \bar{x}_1 x_2 x_3 \vee \bar{z}_1 x_2 \bar{x}_2 \bar{x}_3 \vee \bar{z}_1 x_1 \bar{x}_2 x_3.$$

После минимизации обеих функций z_1 и z'_2 и их преобразования в форму, представляющую собой сочетание операторов электронно-ламповых элементов, мы получим:

$$\left. \begin{aligned} z_1 &= \bar{x}_1 (\bar{x}_2 \vee \bar{x}_3) = T(x_1) \& \Pi(x_2, x_3), \\ z_2 &= \bar{z}_1 (\bar{x}_1 \vee \bar{x}_2) = T(z_1) \& \Pi(x_1, x_2). \end{aligned} \right\} \quad (A)$$

На рис. 6.21 изображена схема, моделирующая формулы (А). Реализация этой схемы требует 6 управляющих сеток. Отметим попутно, что при раздельной (для обоих

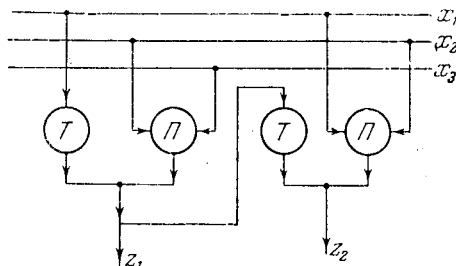


Рис. 6.21.

выходов z_1 и z_2) реализации этой логической сети требуется 9 управляющих сеток.

Рассмотрим теперь еще один общий метод синтеза многовыходной сети, при котором часть элементов схемы используется одновременно для образования нескольких функций. Этот метод, называемый *методом каскадов*, предложен Г. Н. Поваровым*). Он основан на систематическом разложении заданных функций по независимым переменным и на последующем построении *многовыходной пирамидальной сети*. При этом объединяются элементы, реализующие равные друг другу коэффициенты.

Применение метода каскадов для синтеза логической сети мы изложим вначале применительно к реализации одной функции

$$z = f(x_1, x_2, \dots, x_m).$$

Раскладывая ее по переменной x_1 , мы будем иметь:

$$z = \bar{x}_1 f(0, x_2, x_3, \dots, x_m) \vee x_1 f(1, x_2, x_3, \dots, x_m). \quad (6.31)$$

В формуле (6.31) функции $f(0, x_2, x_3, \dots, x_m)$ и $f(1, x_2, x_3, \dots, x_m)$ не содержат переменной x_1 и ее отри-

*) Г. Н. Поваров, Метод синтеза вычислительных и управляющих контактных схем, «Автоматика и телемеханика», 1957, т. 18, № 2.

пания. Раскладывая каждую из них по переменной x_2 , мы получим функции, не содержащие переменной x_2 и ее отрицания. Продолжая этот процесс, придем к функциям, допускающим реализацию элементарной ячейкой.

Преобразуем далее коэффициенты разложения и функции вида (6.31) в суперпозицию операторов элементарных ячеек. При этом коэффициенты разложения, равные нулю или единице, исключаются. Схема, построенная по полученным формулам (и реализующая заданную функцию), называется *пирамидальной*.

Заметим, что функция m переменных может быть представлена формулой вида (6.31) по-разному, в зависимости от выделяемой переменной. На каждом последующем этапе построения пирамидальной схемы число возможных способов приведения функции к виду (6.31) уменьшается на единицу. Следовательно, всего существует

$$m(m-1)(m-2)\dots 4\cdot 3 = \frac{m!}{2}^*)$$

различных пирамидальных схем, реализующих функцию m переменных и зависящих от порядка выделения переменных. Эти пирамидальные схемы будут иметь различное число входных каналов и в общем случае получение минимальной схемы требует пересмотра всех вариантов.

Пример 2. Построить на электронно-ламповых элементах пирамидальную схему для реализации функции

$$z = \overline{x_1}\overline{x_2}\overline{x_3}\overline{x_4} \vee \overline{x_1}x_2\overline{x_3}x_4 \vee x_1x_2\overline{x_3}x_4 \vee x_1x_2x_3x_4,$$

представленной в совершенной дизъюнктивной нормальной форме (в данном случае — минимальной).

Разложим заданную функцию сначала по переменной x_1 . Мы получим:

$$z = \overline{x_1}(\overline{x_2}\overline{x_3}\overline{x_4} \vee x_2\overline{x_3}x_4) \vee x_1(x_2\overline{x_3}\overline{x_4} \vee x_2x_3x_4) = \\ = \overline{x_1}f^{(1)}(x_2, x_3, x_4) \vee x_1f^{(2)}(x_2, x_3, x_4),$$

где под $f^{(1)}(x_2, x_3, x_4)$ и $f^{(2)}(x_2, x_3, x_4)$ имеются в виду соответственно функции $\overline{x_2}\overline{x_3}\overline{x_4} \vee x_2\overline{x_3}x_4$ и $x_2\overline{x_3}\overline{x_4} \vee x_2x_3x_4$.

*) См. ссылку на стр. 319.

Функции $f^{(1)}$ и $f^{(2)}$ мы представим теперь в следующем виде:

$$f^{(1)}(x_2, x_3, x_4) = \bar{x}_3 (\bar{x}_2 \bar{x}_4 \vee x_2 x_4) = \bar{x}_3 f^{(3)}(x_2, x_4),$$

где $f^{(3)}(x_2, x_4)$ есть $\bar{x}_2 \bar{x}_4 \vee x_2 x_4$,

$$f^{(2)}(x_2, x_3, x_4) = x_2 (\bar{x}_3 \bar{x}_4 \vee x_3 x_4) = x_2 f^{(4)}(x_3, x_4),$$

где $f^{(4)}(x_3, x_4)$ есть $\bar{x}_3 \bar{x}_4 \vee x_3 x_4$.

Дальнейшее разложение функций $f^{(3)}$ и $f^{(4)}$ нецелесообразно, так как не дает никакого упрощения.

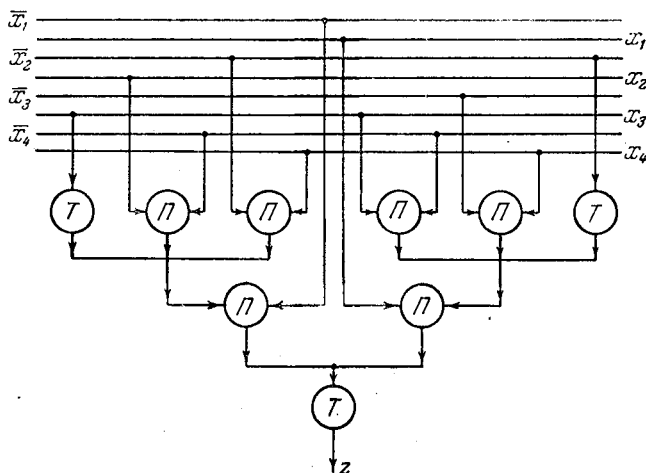


Рис. 6.22.

Преобразуем полученные формулы применительно к операторам пентода и триода. Мы получаем:

$$\left. \begin{aligned} f^{(3)}(x_2, x_4) &= (\bar{x}_2 \vee x_4) \cdot (x_2 \vee \bar{x}_4) = \\ &= \Pi(x_2, \bar{x}_4) \& \Pi(\bar{x}_2, x_4), \\ f^{(4)}(x_3, x_4) &= \Pi(x_3, \bar{x}_4) \& \Pi(\bar{x}_3, x_4), \\ f^{(1)}(x_2, x_3, x_4) &= T(x_3) f^{(3)}, \\ f^{(2)}(x_2, x_3, x_4) &= T(\bar{x}_2) f^{(4)}. \end{aligned} \right\} \quad (A')$$

Итак, функция z выражается следующим образом:

$$z = T \{ \Pi [\bar{x}_1, f^{(1)}] \& \Pi [x_1, f^{(2)}] \}. \quad (B')$$

Схема, построенная по формулам (A') и (B'), изображена на рис. 6.22. Она имеет 15 входных каналов (управляющих сеток) (отметим, что схема реализации функции z непосредственно по совершенной нормальной форме ее представления требует 20 входных каналов).

Пример 3. Построить электронно-ламповую схему реализации логической сети, заданной таблицей VI.8.

Таблица VI.8

x_1	x_2	x_3	z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Из таблицы VI.8 мы получаем следующую совершенную дизъюнктивную нормальную форму представления функции z :

$$z = \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3.$$

Эта форма представления не поддается дальнейшему упрощению в классе нормального представления. Ее реализация в этом виде требует 16 входных каналов.

Построим теперь для функции z пирамидальную схему, преобразуя ее посредством разложения по переменной x_1 . Мы получаем:

$$\begin{aligned} z &= \bar{x}_1 (\bar{x}_2 x_3 \vee x_2 \bar{x}_3) \vee x_1 (\bar{x}_2 \bar{x}_3 \vee x_2 x_3) = \\ &= \bar{x}_1 f^{(1)}(x_2, x_3) \vee x_1 f^{(2)}(x_2, x_3), \quad (A'') \end{aligned}$$

где $f^{(1)}(x_2, x_3)$ есть $\bar{x}_2 x_3 \vee x_2 \bar{x}_3$, а $f^{(2)}(x_2, x_3)$ есть $\bar{x}_2 \bar{x}_3 \vee x_2 x_3$.

Но в рассматриваемом случае $f^{(2)} = \bar{f}^{(1)}$, и соотношение (A'') принимает следующий вид:

$$z = \bar{x}_1 f^{(1)} \vee x_1 \bar{f}^{(1)} = \Pi(x_1, f^{(1)}) \& \Pi[\bar{x}_1, T(\bar{f}^{(1)})], \quad (B'')$$

где функция $f^{(1)}$ в свою очередь может быть представлена в следующем виде:

$$f^{(1)} = (\bar{x}_2 \vee \bar{x}_3)(x_2 \vee x_3) = \Pi(x_2, x_3) \& \Pi(\bar{x}_2, \bar{x}_3). \quad (C'')$$

Схема, моделирующая формулы (B'') и (C''), включает четыре пентода и один триод, и, следовательно, всего 9 управляющих сеток (рис. 6.23).

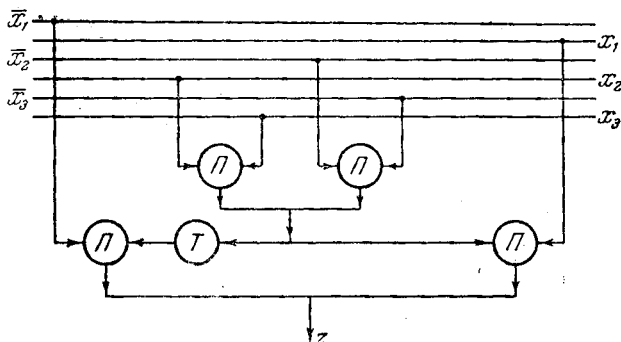


Рис. 6.23.

Порядок синтеза многовыходной сети с использованием метода каскадов лучше всего пояснить на примере.

Пример 4. Пусть синтезируемые функции $z_1 = f_1(x_1, x_2, x_3)$, $z_2 = f_2(x_1, x_2, x_3)$ и $z_3 = f_3(x_1, x_2, x_3)$ заданы в дизъюнктивной нормальной форме, отличны друг от друга и каждая из них не равна нулю или единице:

$$z_1 = \bar{x}_1 x_2 \vee \bar{x}_1 x_3 \vee x_2 x_3,$$

$$z_2 = x_1 \bar{x}_2 \vee x_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3,$$

$$z_3 = \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3.$$

Следуя Г. Н. П о в а р о в у, разложим эти функции сначала по переменной x_1 , затем по x_2 и x_3 ; построим таблицу разложения в следующем виде (см. табл. VI.9).

Таблица VI.9

		z_1	z_2	z_3
1	$z_1 = \bar{x}_1 f_1^{(1)} \vee x_1 f_1^{(2)}$	$f_1^{(1)} = x_2 \vee x_3$ $f_1^{(2)} = x_2 x_3$		
2	$z_2 = \bar{x}_1 f_2^{(1)} \vee x_1 f_2^{(2)}$	$f_2^{(1)} = \bar{x}_2 \bar{x}_3$ $f_2^{(2)} = \bar{x}_2 \vee \bar{x}_3$		
3	$z_3 = \bar{x}_1 f_3^{(1)} \vee x_1 f_3^{(2)}$	$f_3^{(1)} = x_2 x_3 \vee \bar{x}_2 \bar{x}_3$ $f_3^{(2)} = x_2 \bar{x}_3 \vee \bar{x}_2 x_3$		
4	$f_1^{(1)} = \bar{x}_2 f_1^{(3)} \vee x_2 f_1^{(4)}$	$f_1^{(3)} = x_3$ $f_1^{(4)} = 1$		
5	$f_1^{(2)} = \bar{x}_2 f_1^{(5)} \vee x_2 f_1^{(6)}$	$f_1^{(5)} = 0$ $f_1^{(6)} = x_3$		
6	$f_2^{(1)} = \bar{x}_2 f_2^{(3)} \vee x_2 f_2^{(4)}$	$f_2^{(3)} = \bar{x}_3$ $f_2^{(4)} = 0$		
7	$f_2^{(2)} = \bar{x}_2 f_2^{(5)} \vee x_2 f_2^{(6)}$	$f_2^{(5)} = 1$ $f_2^{(6)} = x_3$		
8	$f_3^{(1)} = \bar{x}_2 f_3^{(3)} \vee x_2 f_3^{(4)}$	$f_3^{(3)} = \bar{x}_3$ $f_3^{(4)} = x_3$		
9	$f_3^{(2)} = \bar{x}_2 f_3^{(5)} \vee x_2 f_3^{(6)}$	$f_3^{(5)} = x_3$ $f_3^{(6)} = \bar{x}_3$		

В самой верхней строке (справа) таблицы вписаны обозначения z_1, z_2, z_3 заданных функций; в левом столбце в строках 1—3 приведены разложения этих функций по переменной x_1 . В правом столбце тех же строк выписаны коэффициенты разложения: полный вид функций $f_1^{(1)}, f_1^{(2)}, f_2^{(1)}, f_2^{(2)}, f_3^{(1)}$ и $f_3^{(2)}$. В строках 4—9 таблицы представлено разложение этих коэффициентов по переменной x_2 и выписаны

соответствующие коэффициенты. Дальнейшее разложение (по переменной x_3) в рассматриваемом примере не требуется.

Вычеркнем теперь в списке коэффициентов (правая колонка табл. VI.9) разложения те из них, которые повторяются, равны нулю или единице. Тогда наша таблица примет следующий вид (см. табл. VI.10).

Таблица VI.10

	z_1	z_2	z_3
$z_1 = \bar{x}_1 f_1^{(1)} \vee x_1 f_1^{(2)}$	$f_1^{(1)} = x_2 \vee x_3$ $f_1^{(2)} = x_2 x_3$		
$z_2 = \bar{x}_1 f_2^{(1)} \vee x_1 f_2^{(2)}$	$f_2^{(1)} = \bar{x}_2 \bar{x}_3$ $f_2^{(2)} = \bar{x}_2 \vee \bar{x}_3$		
$z_3 = \bar{x}_1 f_3^{(1)} \vee x_1 f_3^{(2)}$	$f_3^{(1)} = x_2 x_3 \vee \bar{x}_2 \bar{x}_3$ $f_3^{(2)} = \bar{f}_3^{(1)}$		
$f_1^{(1)} = \bar{x}_2 \alpha \vee x_2$	$\alpha = x_3$		
$f_1^{(2)} = x_2 \alpha$	$\beta = \bar{x}_3$		
$f_2^{(1)} = \bar{x}_2 \beta$			
$f_2^{(2)} = \bar{x}_2 \vee x_2 \beta$			
$f_3^{(1)} = \bar{x}_2 \beta \vee x_2 \alpha$			
$f_3^{(2)} = \bar{f}_3^{(1)}$			

Построение многовыходной пирамидальной схемы по формулам, выписанным в таблице VI.10, не вызывает особых затруднений. Для этого коэффициенты разложения и заданные функции преобразуются в формы, пред-

ставляющие собой сочетания элементарных операторов. Таким образом, мы получаем формулы, моделирующие требуемую схему.

Предположим, например, что синтезируемая схема строится на электронно-ламповых элементах. Тогда заданные функции могут быть представлены следующими формулами:

$$z_1 = (\bar{x}_1 \vee f_1^{(2)}) (x_1 \vee f_1^{(1)}) = \Pi(x_1, \bar{f}_1^{(2)}) \& \Pi(\bar{x}_1, \bar{f}_1^{(1)}),$$

$$z_2 = (\bar{x}_1 \vee f_2^{(2)}) (x_1 \vee f_2^{(1)}) = \Pi(x_1, \bar{f}_2^{(2)}) \& \Pi(\bar{x}_1, \bar{f}_2^{(1)}),$$

$$z_3 = (\bar{x}_1 \vee f_3^{(2)}) (x_1 \vee f_3^{(1)}) = \Pi(x_1, \bar{f}_3^{(2)}) \& \Pi(\bar{x}_1, \bar{f}_3^{(1)}).$$

Коэффициенты разложения получают вид:

$$\bar{f}_1^{(1)} = \bar{\alpha} \bar{x}_2 = T_2(x_2, x_3),$$

$$\bar{f}_1^{(2)} = \bar{\alpha} \vee \bar{x}_2 = \Pi(x_2, x_3),$$

$$\bar{f}_2^{(1)} = x_2 \vee \bar{\beta} = \Pi(\bar{x}_2, \bar{x}_3),$$

$$\bar{f}_2^{(2)} = x_2 \bar{\beta} = T_2(\bar{x}_2, \bar{x}_3),$$

$$\bar{f}_3^{(1)} = (x_2 \vee \bar{\beta}) (\bar{x}_2 \vee \bar{\alpha}) = \Pi(\bar{x}_2, \bar{x}_3) \& \Pi(x_2, x_3),$$

$$\bar{f}_3^{(2)} = T(\bar{f}_3^{(1)}).$$

Схема, моделирующая эти формулы, представлена на рис. 6.24. Она содержит 25 управляющих сеток.

Пример 5. Построить электронно-ламповую схему двоичного сумматора последовательного действия по данным, приведенным в примере 10 § 2.

При построении схемы воспользуемся выходной функцией

$$q(t+1) = x_1(t) x_2(t) \vee q(t) x_1(t) \vee q(t) x_2(t) \quad (A^*)$$

в качестве входной переменной сети, вырабатывающей функцию $z(t)$. В соответствии с этим перестроим таблицу VI.7 (стр. 312), представив ее в виде табл. VI.11 (стр. 330).

Из таблицы VI.11 (после минимизации) мы получаем функцию $z(t)$ в следующем виде:

$$z(t) = \bar{q}(t+1) [q(t) \vee x_1(t) \vee x_2(t)] \vee q(t) x_1(t) x_2(t). \quad (B^*)$$

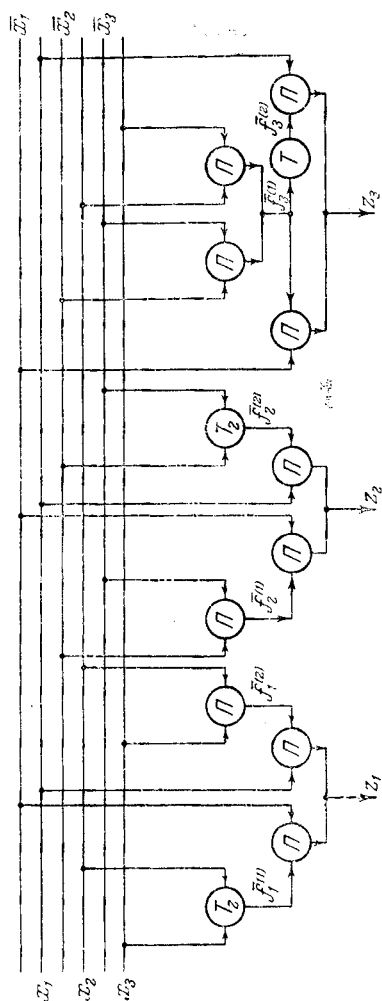


Рис. 6.24.

Таблица VI.11

$q(t+1)$	$q(t)$	$x_1(t)$	$x_2(t)$	$z(t)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	—
0	1	0	0	1
0	1	0	1	—
0	1	1	0	—
0	1	1	1	—
1	0	0	0	—
1	0	0	1	—
1	0	1	0	—
1	0	1	1	0
1	1	0	0	—
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Преобразуем формулы (A*) и (B*) в сочетания элементарных операторов электронно-ламповых схем. Мы получим:

$$\left. \begin{aligned} \bar{q}(t+1) &= \Pi [x_1(t), x_2(t)] \& \Pi [q(t), x_1(t)] \& \\ &\qquad \& \Pi [q(t), x_2(t)], \\ q(t+1) &= T [\bar{q}(t+1)], \\ z(t) &= \Pi \{Q, \Pi [q(t), x_1(t)] \& \\ &\qquad \& \Pi \{Q, \Pi [x_1(t), x_2(t)]\}, \end{aligned} \right\} \quad (C^*)$$

где

$$Q = \Pi [\bar{q}(t+1), q(t)] \& \Pi [\bar{q}(t+1), x_1(t)] \& \Pi [\bar{q}(t+1), x_2(t)].$$

Схема сумматора, моделирующего формулы (C*), изображена на рис. 6.25. Она требует для своей реализации 21 управляющую сетку.

Пользуясь методом каскадов, мы можем построить более экономичную схему двоичного сумматора. Для этого представим функцию $z(t)$ (ср. формулу B''' из

примера 10 § 2) в следующем виде:

$$z(t) = \bar{q}(t) [\bar{x}_1(t) x_2(t) \vee x_1(t) \bar{x}_2(t)] \vee \\ \vee q(t) [\bar{x}_1(t) \bar{x}_2(t) \vee x_1(t) x_2(t)] = \bar{q}(t) \bar{f}(t) \vee q(t) f(t), \quad (E^*)$$

где

$$f(t) = \bar{x}_1(t) \bar{x}_2(t) \vee x_1(t) x_2(t). \quad (F^*)$$

Теперь, воспользовавшись таблицей VI.7 (стр. 312), представим функцию $\bar{q}(t+1)$ в следующем виде:

$$\bar{q}(t+1) = \bar{q}(t) \bar{f}(t) \vee \bar{x}_1(t) \bar{x}_2(t). \quad (G^*)$$

Электронно-ламповая схема сумматора, построенная по формулам (E*), (F*) и (G*), представлена на рис. 6.26.

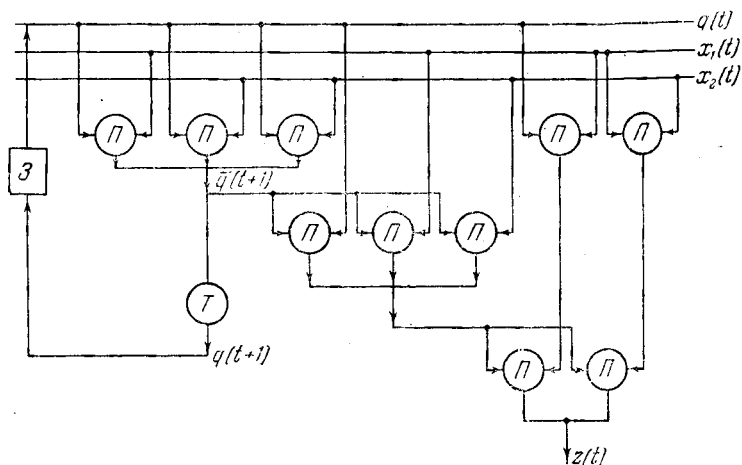


Рис. 6.25.

Она содержит всего 14 управляющих сеток (если не принимать во внимание сетки, необходимые для образования отрицаний переменных).

Способы синтеза сумматора, которыми мы пользовались в этом примере, а также полученные при этом формулы можно применить для построения схемы сумматора на ферритовых сердечниках. Это приведет к экономии числа сердечников. Вместе с тем следует заметить, что соответствующая логическая сеть будет в этом случае работать

во вдвое большем масштабе времени ($\sigma=4$), чем схема, полученная нами в § 2 (см. пример 10, стр. 311—313).

В заключение параграфа отметим, что пирамидальными схемами обычно широко пользуются при синтезе дешиф-

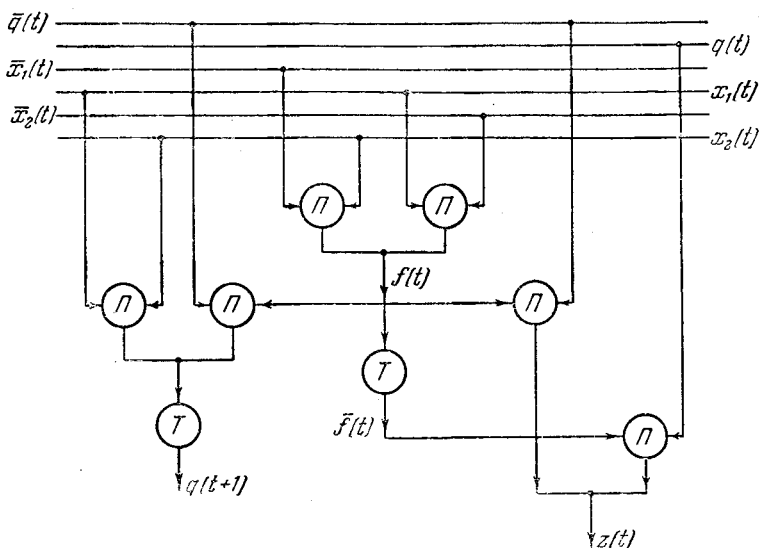


Рис. 6.26.

раторов — многовыходных сетей, каждый из выходов в которых вырабатывает один из конъюнктивных членов. Пирамидальная схема дешифраторов более экономична, чем матричная схема, представленная на рис. 6.6 (см. стр. 291).

Пример 6. Построить пирамидальную схему дешифратора для выработки членов совершенной дизъюнктивной нормальной формы, представляющих собой конъюнкции, содержащие каждая по четыре переменных.

Представим эти конъюнкции из четырех переменных в следующем виде:

$$\begin{aligned} z_0 &= \bar{x}_4 p'_0, & z_1 &= x_4 p'_0, & z_2 &= \bar{x}_4 p''_0, \\ z_3 &= x_4 p''_0, & z_4 &= \bar{x}_4 p'_1, & z_5 &= x_4 p'_1, \\ z_6 &= \bar{x}_4 p''_1, & z_7 &= x_4 p''_1, & \dots \end{aligned}$$

где

$$p'_0 = \bar{x}_3 p_0, \quad p''_0 = x_3 p_0,$$

$$p'_1 = \bar{x}_3 p_1, \quad p''_1 = x_3 p_1,$$

$$\dots \dots \dots$$

и, наконец,

$$p_0 = \bar{x}_1 x_2, \quad p_1 = \bar{x}_1 x_2, \dots$$

Построенная на основе этих соотношений пирамидальная схема изображена на рис. 6.27. Она включает 56 дио-

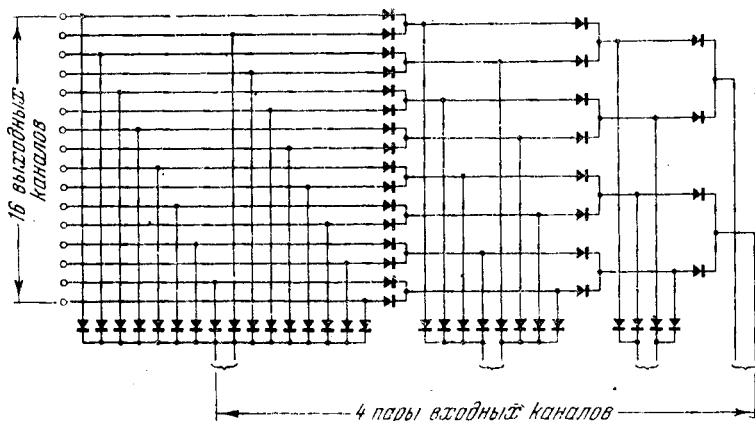


Рис. 6.27.

дов. В общем случае для построения диодного пирамидального дешифратора для m переменных требуется $N = 2^{m+2} - 8$ диодов.

Еще более экономную схему дешифратора можно получить путем разбиения каждой из конъюнкций на подгруппы и использования одинаковых подгрупп для выработки нескольких конъюнкций. В частности, в рассмотренном примере конъюнкции целесообразно представить в следующем виде:

$$z_0 = p_0 p'_0, \quad z_1 = p_0 p'_1,$$

$$z_2 = p_0 p'_2, \quad z_3 = p_0 p'_3,$$

$$z_4 = p_1 p'_0, \quad z_5 = p_1 p'_1,$$

$$\dots \dots \dots$$

где

$$p_0 = \overline{x_1} \overline{x_2}, \quad p'_0 = \overline{x_3} \overline{x_4},$$

$$p_1 = \overline{x_1} x_2, \quad p'_1 = \overline{x_3} x_4,$$

$$p_2 = x_1 \overline{x_2}, \quad p'_2 = x_3 \overline{x_4},$$

$$p_3 = x_1 x_2, \quad p'_3 = x_3 x_4.$$

Соответствующая схема дешифратора, называемая прямоугольной схемой, представлена на

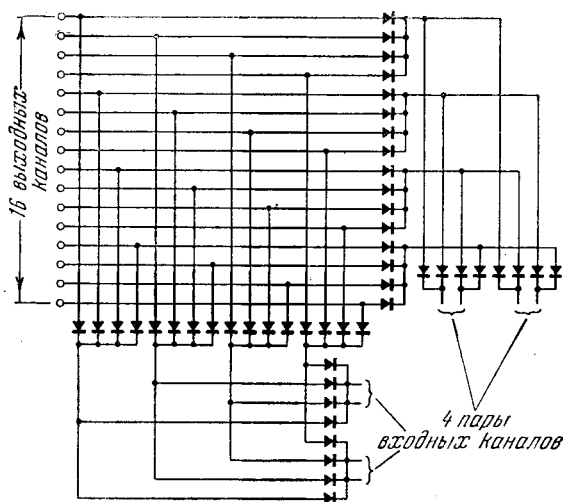


Рис. 6.28.

рис. 6.28. Она строится на 48 диодах. В общем случае реализация диодного прямоугольного дешифратора для m входных переменных требует $N = m2^{\frac{m}{2}} + 2^{m+1}$ диодов.

§ 4. Синтез логической сети «мышь в лабиринте»

Кибернетическая игрушка «мышь в лабиринте», разработанная К. Шенноном, выполнена в виде квадратного лабиринта, образованного 25 площадками, отгороженными друг от друга системой перегородок. На одной

из площадок лабиринта помещается «сало», на другой — «мышь». При запуске игрушки «мышь» начинает блуждать по лабиринту, пока не находит «сало». При этом ее путь образует ряд петель. Если «мышь» запустить повторно с той же площадки, то она движется к «салу», уже не делая петель.

Мы рассмотрим здесь аналогичную задачу поиска пути в лабиринте и построим логическую сеть, осуществляющую переработку информации, воспринимаемой «мышью», в соответствующие сигналы, управляющие ее движением по лабиринту. Для упрощения задачи мы ограничимся лабиринтом, образованным системой треугольных площадок.

Этот лабиринт устроен так. В некоторых стенках, разделяющих пары смежных площадок, имеются двери, остальные стенки являются глухими. «Мышь» должна отыскать путь (если таковой существует), ведущий от той площадки M , на которой она расположена в начале процесса, к площадке C , где расположено «сало». После того как «сало» обнаружено, «мышь» должна остановиться. При этом условия задачи требуют, чтобы был зафиксирован путь без петель, ведущий от M к C (то есть путь, не проходящий дважды через одну и ту же площадку). Если пути от M к C не существует, то «мышь» должна вернуться на исходное место, остановиться и дать сигнал об окончании поиска (в данном случае, безуспешного).

Мы не налагаем никакого ограничения на число площадок в лабиринте или на способы размещения дверей между ними. Вместе с тем мы предполагаем, что каждая дверь снабжена устройством, различающим и фиксирующим факт однократного или двукратного ее прохождения «мышью». На каждом этапе поиска, пока «сало» не обнаружено, «мышь» обзревает лишь одну из трех стенок площадки, на которой она находится. Устройство, которым снабжена дверь в каждой стенке, выдает один из трех сигналов, именуемых условно: 1) желтый ($Ж$), 2) красный ($К$) и 3) зеленый свет ($З$), различаемых в зависимости от того, была ли пройдена дверь в данной стене один раз, два раза, или же она еще ни разу не была пройдена; эти сигналы воспринимаются «мышью». «Красный» сигнал поступает к «мышь» также от глухой стены, снаб-

женной для этого соответствующим устройством; при этом предполагаемый нами механизм поиска таков, что «мышь» более двух раз никогда не проходит через какую-нибудь дверь. Поэтому стена с дважды пройденной дверью приравнивается глухой стене. Кроме того, на площадке *С* «мышь» воспринимает сигнал о наличии «сала».

Отметим тут же следующее соглашение, которого мы в дальнейшем будем придерживаться в наших рассуждениях: когда «мышь» входит в треугольную площадку *ABE* через дверь в стене *A*, то из двух стен *B* и *E* она обзревает ту, которая правее (образно: «мышь» имеет лишь один правый глаз). Следовательно, войдя на площадку, «мышь» сразу же поворачивается на 60° вправо (если треугольники считать равносторонними).

В соответствии со сказанным выше входной алфавит оператора, реализуемого логической сетью «мыши», будет состоять из четырех букв: *З* (зеленый), *К* (красный), *Ж* (желтый), *С* (сало).

Выходной алфавит будет также состоять из четырех букв, а именно из букв *Л*, *П*, \uparrow и $!$, обозначающих: *Л* — поворот на 120° налево, *П* — поворот на 120° направо, \uparrow — прохождение вперед через обзриваемую дверь, $!$ — остановка.

Б. А. Трахтенбротом*) обоснован и описан алгоритм поиска пути в лабиринте, который в несколько измененном виде задается следующей схемой.

Признак, воспринимаемый «мышью»

1. На площадке находится «сало».
2. На площадке имеются две двери, отмеченные желтым сигналом (площадка однажды уже пройдена мышью).

Предписание для ее поведения (хода)

1. Остановиться. Выдать сигнал об успешном окончании поиска.
2. Вернуться на соседнюю площадку через дверь, пройденную последней. При этом желтый сигнал заменяется красным.

*) Б. А. Трахтенброт, Алгоритмы и машинное решение задач, Физматгиз, М., 1960.

3. На площадке имеется хотя бы одна дверь, отмеченная зеленым сигналом (через которую «мышь» еще не проходила).
4. На площадке имеется хотя бы одна дверь, отмеченная желтым сигналом (пройденная «мышью» в точности один раз).
5. Отсутствует любой из перечисленных признаков: на всех стенках данной площадки красные сигналы.
3. Пройти на соседнюю площадку через любую из таких дверей. При этом зеленый сигнал заменяется желтым.
4. Пройти на соседнюю площадку через дверь, пройденную последней. При этом желтый сигнал заменяется красным.
5. Остановиться. Выдать сигнал о безуспешном окончании поиска.

В случае успешного окончания поиска в лабиринте у «мыши» остается зафиксированным путь без петель, ведущий от площадки *М* к площадке *С*. Этот путь однозначно определяется всеми дверьми, которые были пройдены в точности по одному разу.

Используем описанный алгоритм для синтеза логической сети, управляющей поиском в рассматриваемом лабиринте. Перечисленные выше признаки определяют ту или иную букву входного алфавита, воспринимаемую сетью. Выработанная буква является сигналом для выполнения соответствующего хода. Исполнительные схемы, обеспечивающие реализацию хода, то есть передвижение «мыши», изменение цвета сигналов, выдачу сигналов об окончании поиска — успешном или безуспешном, — здесь не рассматриваются.

В соответствии с принятым алгоритмом построим дерево ограниченно-детерминированного оператора, реализующего требуемый процесс переработки информации. Предварительно заметим, что этот алгоритм непосредственно определяет два функционально различных состояния синтезируемой сети, которые мы назовем q_0 и q_1 . Состояние q_0 — это состояние, в которое переходит сеть после прохождения «мышью» какой-либо двери в первый раз;

состояние q_1 — это состояние, в которое переходит сеть после вторичного прохождения «мышью» какой-либо двери

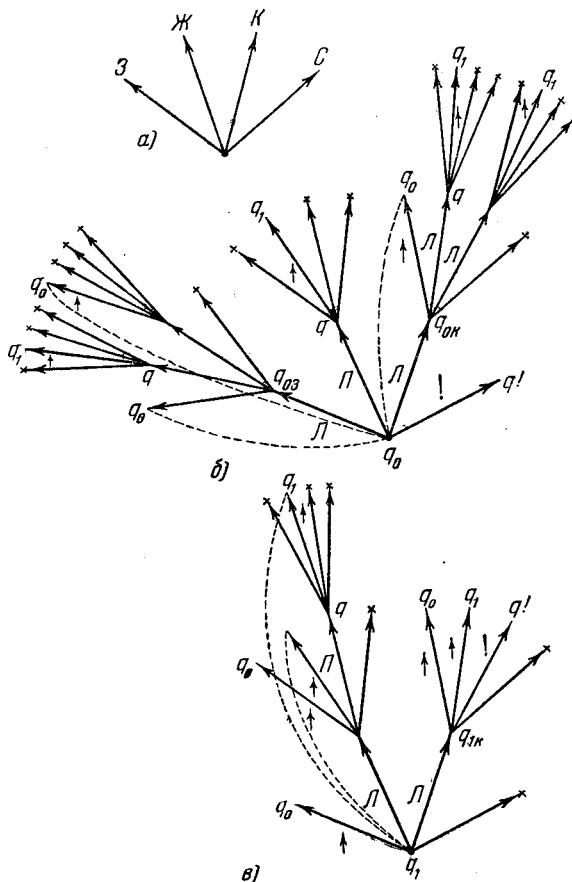


Рис. 6.29.

Различение этих состояний позволяет упростить построение дерева путем его расчленения на две отдельные ветви: 1) ветвь, выходящую из вершины q_0 , и 2) ветвь, выходящую из вершины q_1 .

Сопоставим буквам входного алфавита ребра так, как это показано на рис. 6.29, а, и построим сначала дерево с вер-

пиной q_0 (рис. 6.29,б), имея в виду при этом следующие соображения.

После прохождения какой-либо двери в первый раз и получения от правой стенки площадки, на которую попала «мышь» из этой двери, сигнала «З» нужно, в соответствии с алгоритмом, проверить, не является ли данная площадка уже однажды пройденной, так что на этой площадке замкнулась некоторая петля. В последнем случае от двух дверей площадки, в том числе и от только что пройденной, будут поступать сигналы «Ж». Поэтому «мышь», находясь в состоянии q_0 и получив сигнал «З», должна повернуться на 120° влево для обозрения левой стенки. И если от нее также поступает сигнал «З», то очередным ходом «мыши» является передвижение ее на соседнюю площадку с переходом сети в начальное состояние q_0 . Если же от левой стенки поступает желтый сигнал, то это означает, что на площадке замкнулась петля. При этом должен последовать поворот «мыши» еще на 120° влево (здесь от стенки может поступить только сигнал «Ж»). Последующим ходом «мышь» должна перейти на соседнюю площадку через только что пройденную дверь, а сеть переходит в состояние q_1 .

На схеме дерева, изображенной на рис. 6.29, ребра, соответствующие входным сигналам, поступление которых в данном состоянии сети исключено, снабжены знаком (\times). Этим ребрам и вершинам, в которые они втекают, можно приписать любые из букв принятых алфавитов — выходного алфавита и алфавита состояния, соответственно. Аналогичные соображения приняты во внимание при построении остальных путей в дереве с вершиной q_0 .

На рис. 6.29,в показано дерево с вершиной q_1 , соответствующей состоянию сети после двукратного прохождения «мышью» какой-либо из дверей.

Ниже приведена объединенная каноническая таблица оператора, реализуемого сетью, которая построена по его дереву (табл. VI.12). Для понимания этой таблицы следует иметь в виду, что алфавит состояний рассматриваемой сети включает буквы q_0 , q_1 , q , q^1 , $q_{0к}$, $q_{0з}$, $q_{1к}$, $q_{1ж}$, имеющие следующие значения:

q_0 — состояние, в которое переходит сеть после однократного прохождения «мышью» какой-либо двери;

Таблица VI.12

Входные состояния	Красный	Зеленый	Желтый	«Сало»
q_0	Π/q_{0K}	$\Pi/q_{0З}$	Π/q	$!/q!$
q_{0K}	Π/q	\uparrow/q_0	Π/q	—
$q_{0З}$	Π/q	\uparrow/q_0	Π/q	—
q_1	Π/q_{1K}	\uparrow/q_0	$\Pi/q_{1Ж}$	—
q_{1K}	$!/q!$	\uparrow/q_0	\uparrow/q_1	—
$q_{1Ж}$	Π/q	\uparrow/q_0	—	—
q	—	\uparrow/q_0	\uparrow/q_1	—
$q!$	$!/q!$	—	—	$!/q!$

q_1 —состояние, в которое переходит сеть после прохождения «мышью» какой-либо двери второй раз;

q —состояние, которое можно описать как готовность к прохождению через обозреваемую дверь;

$q!$ —состояние покоя (при остановке);

q_{0K} —состояние, в которое переходит сеть в случае, когда после первого прохождения какой-либо двери воспринимается красный сигнал;

$q_{0З}$ —состояние, в которое переходит сеть в случае, когда после первого прохождения какой-либо двери воспринимается зеленый сигнал;

q_{1K} —состояние, в которое переходит сеть в случае, когда после повторного прохождения данной двери воспринимается красный сигнал;

$q_{1Ж}$ —состояние, в которое переходит сеть в случае, когда после повторного прохождения данной двери воспринимается желтый сигнал.

Двоичные коды для алфавитов — внешних и внутреннего — выберем в соответствии с таблицами, которые помещены ниже.

Двоичные коды (x_1x_2) входного алфавита представлены в таблице VI.13.

Двоичные коды (z_1z_2) выходного алфавита представлены в таблице VI.14.

Таблица VI.13

З	Ж	К	С
00	01	10	11

Таблица VI.14

Л	П	↑	!
00	01	10	11

Двоичные коды ($\varphi_1\varphi_2\varphi_3$) алфавита состояний представлены в таблице VI.15.

Таблица VI.15

q_0	q_{0K}	$q_{0З}$	q_1	q_{1K}	$q_{1Ж}$	q	q^1
000	001	010	011	100	101	110	111

При этих условиях таблица алфавита «мыши» будет иметь следующий вид (см. табл. VI.16).

Примечание. В столбцах 10—14 в скобках указаны доопределенные значения функций $z_1(t)$, $z_2(t)$, $\varphi_1(t+1)$, $\varphi_2(t+1)$, $\varphi_3(t+1)$.

Из таблицы VI.16 мы после упрощения получаем следующую систему канонических уравнений:

$$z_1(t) = \bar{x}_1(t) \{ \bar{x}_2(t) [\varphi_2(t) \vee \varphi_3(t)] \vee \varphi_1(t) \} \vee \\ \vee x_1(t) \{ \varphi_1(t) [\varphi_2(t) \vee \bar{\varphi}_3(t)] \vee x_2(t) \}, \quad (\mathcal{A})$$

$$z_2(t) = x_2(t) \{ [\bar{\varphi}_1(t) \bar{\varphi}_2(t) \bar{\varphi}_3(t) \vee x_1(t)] \vee \\ \vee x_1(t) [\varphi_2(t) \bar{\varphi}_3(t) \vee \varphi_1(t)], \quad (\mathcal{B})$$

$$\varphi_1(t+1) = x_2(t) [x_1(t) \vee \bar{\varphi}_1(t)] \vee \\ \vee x_1(t) [\varphi_1(t) \vee \varphi_2(t) \vee \varphi_3(t)], \quad (\mathcal{C})$$

$$\varphi_2(t+1) = \bar{x}_1(t) \{ x_2(t) [\varphi_1(t) \vee \bar{\varphi}_2(t) \vee \bar{\varphi}_3(t)] \vee \\ \vee \bar{\varphi}_1(t) \bar{\varphi}_2(t) \bar{\varphi}_3(t) \} \vee x_1(t) [x_2(t) \vee \varphi_1(t) \vee \\ \vee \bar{\varphi}_2(t) \varphi_3(t) \vee \varphi_2(t) \bar{\varphi}_3(t)], \quad (\mathcal{D})$$

$$\varphi_3(t+1) = \bar{x}_1(t) \{ x_2(t) [\varphi_2(t) \varphi_3(t) \vee \varphi_1(t)] \vee \\ \vee x_1(t) \{ \bar{\varphi}_2(t) \varphi_3(t) \vee \varphi_1(t) [\varphi_2(t) \vee \varphi_3(t)] \vee x_2(t) \}. \quad (\mathcal{E})$$

Таблица VI.16

Вход- ной сиг- нал	Вход- ное состо- яние	Код входа		Код входного состояния			Вы- ход- ной сиг- нал	Выход- ное со- стояние	Код выхода		Код выходного состояния			
		$x_1(t)$	$x_2(t)$	$\varphi_1(t)$	$\varphi_2(t)$	$\varphi_3(t)$			$z_1(t)$	$z_2(t)$	$\varphi_1(t+1)$	$\varphi_2(t+1)$	$\varphi_3(t+1)$	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
3	q_0	0	0	0	0	0	L	q_{03}	0	0	0	1	0	
3	q_{0K}	0	0	0	0	1	\uparrow	q_0	1	0	0	0	0	
3	q_{03}	0	0	0	1	0	\uparrow	q_0	1	0	0	0	0	
3	q_1	0	0	0	1	1	\uparrow	q_0	1	0	0	0	0	
3	q_{1K}	0	0	1	0	0	\uparrow	q_0	1	0	0	0	0	
3	q_{1K}	0	0	1	0	1	\uparrow	q_0	1	0	0	0	0	
3	q	0	0	1	1	0	\uparrow	q_0	1	0	0	0	0	
3	$q^!$	0	0	1	1	1	—	—	(1)	(0)	(0)	(0)	(0)	
K	q_0	0	1	0	0	0	L	q	0	1	1	1	0	
K	q_{0K}	0	1	0	0	1	L	q	0	0	1	1	0	
K	q_{03}	0	1	0	1	0	L	q	0	0	1	1	0	
K	q_1	0	1	0	1	1	L	q_{1K}	0	0	1	1	1	
K	q_{1K}	0	1	1	0	0	\uparrow	q_1	1	0	0	1	1	
K	q_{1K}	0	1	1	0	1	—	—	(1)	(0)	(0)	(4)	(1)	
K	q	0	1	1	1	0	\uparrow	q_1	1	0	0	1	1	
K	$q^!$	0	1	1	1	1	—	—	(1)	(0)	(0)	(4)	(1)	

Продолжение табл. VI.16

Вход- ной сиг- нал	Вход- ное состо- яние	Код входа			Код входного состояния			Выход- ной сиг- нал	Выход- ное со- сто- яние	Код выхода			Код выходного состояния		
		$x_1(t)$	$x_2(t)$		$\Phi_1(t)$	$\Phi_2(t)$	$\Phi_3(t)$			$z_1(t)$	$z_2(t)$		$\Phi_1(t+1)$	$\Phi_2(t+1)$	$\Phi_3(t+1)$
1	2	3	4	5	6	7	8	9		10	11		12	13	14
K	q_0	1	0	0	0	0	Π	q_{0K}		0	0		0	0	1
K	q_{0K}	1	0	0	0	1	Π	q		0	0		1	1	0
K	q_{03}	1	0	0	1	0	Π	q		0	1		1	1	0
K	q_1	1	0	0	1	1	Π	q_{1K}		0	0		1	0	0
K	q_{1K}	1	0	1	0	0	!	q_1		1	1		1	1	1
K	q_{1K}	1	0	1	0	1	Π	q		0	1		1	1	0
K	q	1	0	1	1	0	—	—		(1)	(1)		(1)	(1)	(1)
K	q_1	1	0	1	1	1	!	q_1		1	1		1	1	1
C	q_0	1	1	0	0	0	!	q_1		1	1		1	1	1
C	q_{0K}	1	1	0	0	1	—	—		(1)	(1)		(1)	(1)	(1)
C	q_{03}	1	1	0	1	0	—	—		(1)	(1)		(1)	(1)	(1)
C	q_1	1	1	0	1	1	—	—		(1)	(1)		(1)	(1)	(1)
C	q_{1K}	1	1	1	1	0	—	—		(1)	(1)		(1)	(1)	(1)
C	q_{1K}	1	1	1	0	0	—	—		(1)	(1)		(1)	(1)	(1)
C	q	1	1	1	1	1	!	q_1		(1)	(1)		(1)	(1)	(1)
C	q_1	1	1	1	1	1	!	q_1		1	1		1	1	1

Перейдем теперь к построению логической сети по формулам (У) — (Ф). В этом построении мы будем пользоваться методом каскадов. Будем, далее, предполагать, что сеть реализуется с помощью электронно-ламповых элементов. В соответствии со сказанным преобразуем уравнения (У) — (Ф) по следующей схеме (индексы t для упрощения опускаем):

$$\begin{aligned}
 z_1 &= \bar{x}_1 f_1^{(1)} \vee x_1 f_1^{(2)}, \\
 f_1^{(1)} &= \bar{x}_2 f_1^{(3)} \vee x_2 \varphi_1, \\
 f_1^{(3)} &= \varphi_1 \vee \varphi_2 \vee \varphi_3, \\
 f_1^{(2)} &= \bar{x}_2 f_1^{(4)} \vee x_2, \\
 f_1^{(4)} &= \varphi_1 (\varphi_2 \vee \bar{\varphi}_3). \\
 z_2 &= \bar{x}_1 f_2^{(1)} \vee x_1 f_2^{(2)}, \\
 f_2^{(1)} &= x_2 f_2^{(3)}, \\
 f_2^{(3)} &= \bar{f}_1^{(3)}, \\
 f_2^{(2)} &= \bar{x}_2 f_2^{(4)} \vee x_2, \\
 f_2^{(4)} &= \varphi_1 \vee \varphi_2 \bar{\varphi}_3. \\
 \varphi_1(t+1) &= \bar{x}_1 f_3^{(1)} \vee x_1 f_3^{(2)}, \\
 f_3^{(1)} &= x_2 \bar{x}_3, \\
 f_3^{(2)} &= \bar{x}_2 f_3^{(3)} \vee x_2, \\
 f_3^{(3)} &= f_1^{(3)}. \\
 \varphi_2(t+1) &= \bar{x}_1 f_4^{(1)} \vee x_1 f_4^{(2)}, \\
 f_4^{(1)} &= \bar{x}_2 f_4^{(3)} \vee x_2 f_4^{(4)}, \\
 f_4^{(3)} &= \bar{f}_1^{(3)}, \\
 f_4^{(4)} &= \varphi_1 \vee \bar{\varphi}_2 \vee \bar{\varphi}_3, \\
 f_4^{(2)} &= \bar{x}_2 f_4^{(5)} \vee x_2, \\
 f_4^{(5)} &= \varphi_1 \vee \varphi_2 \bar{\varphi}_3 \vee \bar{\varphi}_2 \bar{\varphi}_3. \\
 \varphi_3(t+1) &= \bar{x}_1 f_5^{(1)} \vee x_1 f_5^{(2)}, \\
 f_5^{(1)} &= x_2 f_5^{(3)}, \\
 f_5^{(3)} &= \varphi_1 \vee \varphi_2 \varphi_3, \\
 f_5^{(2)} &= \bar{x}_2 f_5^{(4)} \vee x_2, \\
 f_5^{(4)} &= \varphi_1 \varphi_2 \vee \varphi_1 \varphi_3 \vee \bar{\varphi}_2 \bar{\varphi}_3.
 \end{aligned}$$

Представим, далее, функции $f_2^{(4)}$, $f_5^{(3)}$, $f_4^{(5)}$ и $f_5^{(4)}$ в следующем виде:

$$\begin{aligned} f_2^{(4)} &= \alpha\gamma, \\ f_5^{(3)} &= \beta\gamma, \\ f_4^{(5)} &= \bar{\varphi}_2\beta \vee \varphi_2\alpha, \\ f_5^{(4)} &= \bar{\varphi}_2\alpha \vee \varphi_2\varphi_1, \end{aligned}$$

где:

$$\begin{aligned} \alpha &= \varphi_1 \vee \bar{\varphi}_3, \\ \beta &= \varphi_1 \vee \varphi_3, \\ \gamma &= \varphi_1 \vee \varphi_2. \end{aligned}$$

Преобразование полученных формул в сочетания элементарных операторов электронно-ламповых схем можно выполнить на основе следующих соотношений:

$$\begin{aligned} z_1(t) &= K_2 \{T_2[x_1, \bar{f}_1^{(1)}], T_2[\bar{x}_1, \bar{f}_1^{(2)}]\}, \\ z_2(t) &= K_2 \{T_2[x_1, \bar{f}_2^{(1)}], T_2[\bar{x}_1, \bar{f}_2^{(2)}]\}, \\ \varphi_1(t+1) &= K_2 \{T_2[x_1, \bar{f}_3^{(1)}], T_2[\bar{x}_1, \bar{f}_3^{(2)}]\}, \\ \varphi_2(t+1) &= K_2 \{T_2[x_1, \bar{f}_4^{(1)}], T_2[\bar{x}_1, \bar{f}_4^{(2)}]\}, \\ \varphi_3(t+1) &= K_2 \{T_2[x_1, \bar{f}_5^{(1)}], T_2[\bar{x}_1, \bar{f}_5^{(2)}]\}. \end{aligned}$$

Далее:

$$\bar{f}_1^{(1)} = (x_2 \vee \bar{f}_1^{(3)}) (\bar{x}_2 \vee \bar{\varphi}_1) = \Pi(\bar{x}_2, f_1^{(3)}) \& \Pi(x_2, \varphi_1);$$

поскольку

$$f_1^{(3)} \doteq T[T_3(\varphi_1, \varphi_2, \varphi_3)],$$

мы получаем:

$$\bar{f}_1^{(1)} = \Pi\{\bar{x}_2, T[T_3(\varphi_1, \varphi_2, \varphi_3)] \& \Pi(x_2, \varphi_1)\}.$$

Вывод формул, представляющих остальные функции в терминах операторов электронно-ламповых схем, мы предоставляем читателю. В окончательной форме они имеют

следующий вид:

$$\begin{aligned}\bar{\Phi}_1 &= T(\Phi_1), & \alpha &= \Pi(\bar{\Phi}_1, \Phi_3), \\ \bar{\Phi}_2 &= T(\Phi_2), & \beta &= \Pi(\bar{\Phi}_1, \bar{\Phi}_3), \\ \bar{\Phi}_3 &= T(\Phi_3), & \gamma &= \Pi(\bar{\Phi}_1, \bar{\Phi}_2),\end{aligned}$$

$$\begin{aligned}\bar{f}_1^{(2)} &= \bar{x}_2 \bar{f}_1^{(4)} = T_2[x_2, (T(\bar{\Phi}_1) \cdot \Pi(\bar{\Phi}_2 \vee \Phi_3))], \\ \bar{f}_2^{(1)} &= \bar{x}_2 \vee \bar{f}_2^{(3)} = \Pi(x_2, \bar{f}_2^{(3)}) = \Pi(x_2, \bar{f}_1^{(3)}), \\ \bar{f}_2^{(2)} &= \bar{x}_2 \bar{f}_2^{(4)} = T_2(x_2, \bar{f}_2^{(4)}), \\ f_2^{(4)} &= \alpha \gamma = D_K(\alpha, \gamma), \\ \bar{f}_3^{(1)} &= \bar{x}_2 \vee x_3 = \Pi(x_2, x_3), \\ \bar{f}_3^{(2)} &= \bar{x}_2 \bar{f}_3^{(3)} = T_2(x_2, \bar{f}_1^{(3)}), \\ \bar{f}_4^{(1)} &= (x_2 \vee \bar{f}_4^{(3)}) (\bar{x}_2 \vee \bar{f}_4^{(4)}) = \Pi(\bar{x}_2, \bar{f}_1^{(3)}) \cdot \Pi[x_2, K_2(\alpha, \bar{\Phi}_2)], \\ f_4^{(2)} &= \bar{x}_2 \bar{f}_4^{(5)} = T_2(x_2, \bar{f}_4^{(5)}), \\ f_4^{(5)} &= T[\Pi(\bar{\Phi}_2, \beta) \cdot \Pi(\Phi_2, \alpha)], \\ \bar{f}_5^{(1)} &= \bar{x}_2 \vee \bar{f}_5^{(3)} = \Pi(x_2, \bar{f}_5^{(3)}), \\ f_5^{(3)} &= \beta \gamma = D_K(\beta, \gamma), \\ f_5^{(2)} &= \bar{x}_2 \bar{f}_5^{(4)} = T_2(x_2, \bar{f}_5^{(4)}), \\ f_5^{(4)} &= T[\Pi(\bar{\Phi}_2, \alpha) \cdot \Pi(\Phi_2, \Phi_1)].\end{aligned}$$

Электронно-ламповая схема, моделирующая написанные формулы, включает 84 входных канала, не считая сеток, необходимых для реализации функции отрицания переменных x_1, x_2 .

ГЛАВА VII

АСИМПТОТИЧЕСКАЯ ОЦЕНКА СЛОЖНОСТИ ЛОГИЧЕСКИХ СЕТЕЙ

§ 1. Основные понятия. Формулировка предельной теоремы

В этой главе исследуется в самом общем виде задача построения оптимальной логической сети, реализующей заданный ограниченно-детерминированный оператор.

Специфическая особенность этого исследования заключается в том, что оно направлено на выяснение асимптотических закономерностей, проявляющихся при больших значениях параметров ограниченно-детерминированных операторов (веса, числа входных или выходных букв). Такой подход к делу был впервые развит К. Шенноном*) применительно к реализации истинностных операторов в контактных схемах; им же были получены первые существенные результаты в этом направлении. Окончательное решение задачи, сформулированной Шенноном, было получено О. Б. Лупановым, который обобщил эту проблематику также на схемы других типов, в том числе и на логические сети (истинностные)**). Этому кругу вопросов посвящен § 2. Последующие параграфы содержат материал, необходимый для распространения

*) C. E. Shannon, The synthesis of two-terminal switching circuits. BSTJ, 1949, v. 28, № 1, p. 59—98.

**) О. Б. Лупанов, Об одном методе синтеза схем. Изв. высш. учебн. завед. Радиофизика, № 1, 1958.

предельной теоремы Шеннона—Лупанова на логические сети с памятью.

В этом параграфе мы введем ряд понятий и обозначений, необходимых для дальнейшего изложения, и сформулируем основные теоремы. Доказательство этих теорем и соответствующие конструкции будут осуществлены в следующих параграфах; в настоящем параграфе мы ограничимся лишь некоторыми предварительными пояснениями.

Будем рассматривать всевозможные логические сети*) над некоторой произвольной, — но зафиксированной — системой элементов $\{\mathfrak{M}_i\}$, от которой требуется лишь, чтобы она была п о л н о й; такую систему элементов будем называть *базисом*. Для постановки задачи о нахождении оптимальной сети, реализующей данный оператор, необходимо располагать объективным критерием для сравнения сетей (по сложности, стоимости, надежности или другим признакам). Пусть в соответствии с таким критерием к каждой сети Ω отнесено положительное действительное число $L(\Omega)$, именуемое *индексом простоты* или *стоимостью* сети Ω . Каждый ограниченно-детерминированный оператор θ может быть реализован сетями над $\{\mathfrak{M}_i\}$ (в силу полноты этой системы), причем, вообще говоря, различными сетями, отличающимися, быть может, своими индексами простоты.

Обозначим через $L(\theta)$ наименьший из таких индексов простоты. Задача оптимальной реализации оператора θ заключается в построении любой из тех реализующих его сетей, индекс простоты которых как раз и равняется $L(\theta)$. Так, например, в сети над ламповыми элементами за индекс простоты можно было бы принять число всех управляющих сеток, встречающихся в сети; построение оптимальной сети означало бы в таком случае построение сети с минимальным числом управляющих сеток.

Очевидно, оптимальная реализация зависит от выбранной системы оценки сложности сетей. В излагаемой далее теории этот выбор может быть любым, лишь бы только выполнялись следующие два требования:

*) В дальнейшем слово «логические» мы будем иногда опускать.

I. Каждый элемент \mathfrak{M}_i из рассматриваемого базиса имеет положительный индекс простоты

$$L(\mathfrak{M}_i) > 0 \quad (\text{где } i = 1, 2, \dots, \mu^*).$$

II. Индекс простоты сети равняется сумме индексов простоты элементов, из которых она составлена.

Следовательно, индексы простоты сетей являются числами вида:

$$n_1 L(\mathfrak{M}_1) + n_2 L(\mathfrak{M}_2) + \dots + n_\mu L(\mathfrak{M}_\mu).$$

Оценка сложности, удовлетворяющая этим постулатам, имеет, может быть, тот недостаток, что она не отражает характера соединений между элементами, а только их состав в рассматриваемой сети. В остальном же естественность и оправданность сформулированных постулатов представляются очевидными.

Из постулатов I, II и определения логической сети легко вытекает тривиальный алгоритм для решения задачи оптимальной реализации данного оператора θ .

Этот алгоритм основан на следующих соображениях:

1. Числа, выражающие индексы простоты всевозможных логических сетей над $\{\mathfrak{M}_i\}$, можно эффективно расположить в виде возрастающей последовательности:

$$L_1, L_2, \dots, L_j, \dots$$

Очевидно, L_1 является индексом простоты сети, состоящей из одного элемента, а именно из «самого простого» элемента.

2. Для каждого из чисел L_j существует лишь конечное множество S_{L_j} различных сетей с индексом простоты, не большим, чем L_j , причем эти сети могут быть все построенные одна за другой.

Пусть теперь задан какой-либо оператор θ (мы можем предполагать, например, что он задан своими каноническими таблицами), для которого требуется найти оптимальную реализацию. Тогда для фиксированного L_j мы начинаем одну за другой строить сети из S_{L_j} ; каждую построенную сеть можно проанализировать и путем сравнения с каноническими таблицами оператора θ установить, реали-

*) μ — число элементов в базисе.

зует ли эта сеть оператор θ или нет. Начиная этот процесс с сетей из множества S_{L_1} и переходя потом к S_{L_2}, S_{L_3}, \dots , мы продолжаем этот процесс до тех пор, пока впервые не обнаружим сеть, реализующую оператор θ ; найденная сеть и будет искомой минимальной сетью.

Итак, предложенный алгоритм действительно решает поставленную задачу, причем идея, лежащая в его основе, весьма проста и сводится, по существу, к просмотру большого множества сетей. Нетрудно показать, что число сетей в S_L с ростом L растет как $(cL)^{cL}$, где c есть некоторая константа; поэтому применение алгоритма перебора практически невозможно уже при сравнительно небольших значениях параметров.

Но помимо громоздкости, этот тривиальный алгоритм обладает еще одним существенным недостатком: из него нельзя извлечь никакого предварительного представления о сложности той сети, которая будет получена в результате его применения.

Основное содержание настоящей главы посвящено исследованию эффективных методов решения задачи оптимизации, не требующих пересмотра всего множества сетей. Применение этих методов приводит не только к получению почти оптимальных (в определенном смысле) сетей, но и позволяет заранее оценить их индексы простоты.

Для разъяснения этого круга вопросов необходимо предварительно ввести некоторые важные понятия и обозначения.

Рассмотрим класс всех ограниченно-детерминированных операторов с логарифмическим весом, равным k (то есть с весом $\lambda = 2^k = 1, 2, 3, \dots$), для которых входными буквами служат всевозможные наборы длины m из единиц и нулей ($m=1, 2, 3, \dots$), а выходными — аналогичные наборы длины n ($n=1, 2, 3, \dots$). Этот класс, который при любых фиксированных m, n, k конечен, мы обозначим через $\theta_{m,n,k}$. Число операторов, входящих в класс $\theta_{m,n,k}$, мы обозначим через $N(m, n, k)$. Очевидно, что $\theta_{m,n,k}$ состоит из таких операторов, которые могут быть заданы посредством систем канонических уравнений вида:

$$\left. \begin{aligned} z_i(t) &= \varphi_i[x_1(t), \dots, x_m(t), q_1(t), \dots, q_k(t)], \\ q_j(t) &= \psi_j[x_1(t), \dots, x_m(t), q_1(t), \dots, q_k(t)], \end{aligned} \right\} \quad (7.1)$$

где $2^{\tilde{k}-1} < \lambda = 2^{\tilde{k}} \leq 2^{\tilde{k}+1}$. Следуя К. Шеннону, введем в рассмотрение функцию $L(m, n, k)$, равную наименьшему числу L такому, что любой оператор класса $\theta_{m,n,k}$ может быть реализован в логической сети, стоимость которой не превосходит L . Очевидно, что

$$\left. \begin{aligned} L(m, n, k) &= \max_{\theta \in \theta_{m,n,k}} L(\theta), \\ \theta &\in \theta_{m,n,k}. \end{aligned} \right\} \quad (7.2)$$

Выше мы указали тривиальный алгоритм решения задачи оптимизации. Ясно, что этот алгоритм решает также задачу точного вычисления значений функции $L(m, n, k)$. Для этого достаточно построить (в соответствии с тривиальным алгоритмом) оптимальные реализации всех операторов из $\theta_{m,n,k}$ (а число этих операторов конечно) и из всех найденных для них чисел $L(\theta)$ выбрать наибольшее. Вместе с тем изложенные выше соображения, относящиеся к громоздкости и практической неприемлемости описанного алгоритма, автоматически переносятся и на задачу вычисления функции $L(m, n, k)$.

Легко понять, что такая процедура вычисления функции $L(m, n, k)$ совершенно неприемлема. В связи с этим представляет интерес разработка таких методов, которые позволяли бы оценить функцию $L(m, n, k)$ хотя бы и приближенно, но достаточно эффективно, например в виде адекватной и легко обозримой формулы.

С этой целью мы рассмотрим поведение функции $L(m, n, k)$ в случае, когда $m+k$ стремится к бесконечности, а на рост n наложены некоторые ограничения. Смысл этих ограничений заключается в следующем. Как известно, для рассматриваемых нами операторов выходная буква $z(t)$ оператора однозначным образом определяется парой входных букв $x(t), q(t)$. Поэтому число букв выходного алфавита, которые фактически употребляются при задании оператора, не больше произведения числа букв входного алфавита на число букв алфавита состояний. Если все (или «почти» все) выходные кодовые группы фактически использованы, то

$$2^n \leq 2^m \cdot 2^k, \text{ то есть } n \leq m + k. \quad (7.3)$$

Но обычно не все 2^n кодовые группы фактически используются в качестве выходных букв (часто, например, из

соображений надежности специально выбирают избыточный код). Поэтому n может быть и больше, чем $m+k$. В дальнейшем такая избыточность не исключается, но требуется, чтобы она не была слишком большой. Точнее, мы принимаем условие:

$$\lim_{m+k \rightarrow \infty} \frac{\lg n}{m+k} = 0. \quad (7.4)$$

При этом условии удастся выяснить асимптотическое поведение функции $L(m, n, k)$ *).

Представляется вполне естественным, что в асимптотической формуле для $L(m, n, k)$, кроме параметров m, n, k , характеризующих реализуемые операторы класса $\theta_{m, n, k}$, должны участвовать и параметры, характеризующие систему элементов, из которых строятся логические сети, реализующие эти операторы, а также принятый способ оценки их сложности. Мы ограничимся пока рассмотрением полных систем элементов, содержащих элементы следующих двух типов: 1) элементы, каждый из которых имеет один выходной канал и реализует одну функцию алгебры логики; 2) элементы, каждый из которых имеет один входной канал и реализует какой-нибудь ограниченно-детерминированный оператор.

Общность последующих рассуждений обеспечивается тем, что ограничения, наложенные нами на системы элементов, являются мало стеснительными и обычно выполняются.

В качестве типичного примера можно указать систему, состоящую из электронных ламп (относящихся к элементам типа 1)) и задерживающего элемента, или одно-входного триггера (элемент типа 2)).

Заметим, что среди элементов типа 1) обязательно должен быть по крайней мере один такой, который реализует функцию алгебры логики, существенно зависящую больше чем от одной переменной. В противном случае

*) Напомним, что две величины μ, ν называются *асимптотически равными в некотором процессе* ($\mu \sim \nu$), если в этом процессе $\lim \frac{\mu}{\nu} = 1$; это означает, что относительная погрешность, порожденная заменой одной из этих величин другой, стремится к нулю.

система элементов не была бы полной, ибо из элементов такой системы нельзя было бы строить логические сети, реализующие функции алгебры логики, существенно зависящие более чем от одной переменной.

Условимся называть *удельной стоимостью* любого такого элемента \mathfrak{M}_v число $\frac{L(\mathfrak{M}_v)}{n-1}$, где $L(\mathfrak{M}_v)$ есть стоимость элемента, а n — число переменных, от которых существенно зависит функция, реализуемая им. Обозначим, далее, через ϱ наименьшую из удельных стоимостей элементов. Это число ϱ и является единственным параметром базисного набора и принятого способа оценки стоимостей, который участвует в формуле, выражающей асимптотическую оценку функции $L(m, n, k)$. Именно, может быть доказана следующая теорема.

П р е д е л ь н а я т е о р е м а . Если $m + k \rightarrow \infty$ и $\frac{\lg n}{m+n} \rightarrow 0$, то

$$L(m, n, k) \sim n\varrho \frac{2^{m \cdot 2^k}}{m+k} + k\varrho \frac{(2^m - 1) \cdot 2^k}{m+k}. \quad (7.5)$$

Вместе с тем при условиях, указанных в этой теореме (и даже при несколько более общих условиях), имеет место асимптотическое равенство *):

$$\lg N(m, n, k) \sim (2^m - 1) \cdot 2^k (n + k) + n \cdot 2^k. \quad (7.6)$$

Это позволяет преобразовать формулу (7.5) в следующий более обозримый вид:

$$L(m, n, k) \sim \varrho \frac{\lg N(m, n, k)}{\lg \lg N(m, n, k)}. \quad (7.7)$$

В частности, когда $k=0$ и $n=1$, формула (7.7) принимает вид:

$$L(m) = L(m, 1, 0) \sim \varrho \frac{2^m}{m}. \quad (7.8)$$

Функция $L(m)$ была введена К. Шенноном первоначально для оценки сложности реализации функций

*) Напомним, что $N(m, n, k)$ есть число всех операторов из $\theta_{m, n, k}$.

алгебры логики от m переменных в контактных схемах*). О. Б. Лупанов исследовал функцию $L(m)$ применительно к контактным схемам и к логическим сетям**) (именуемым им схемами из функциональных элементов) и установил для них асимптотическую оценку (7.8). Формула (7.5) получена нами путем распространения метода О. Б. Лупанова***), примененного им к исследованию реализации функций алгебры логики (то есть, по существу, истинностных операторов), на операторы с памятью****).

Формула (7.5) свидетельствует о том, что поведение функции $L(m, n, k)$ мало зависит от природы тех элементов, из которых строятся логические сети, и определяется лишь минимальной удельной стоимостью элементов типа 1). Что касается элементов типа 2), то их параметры вообще не отражены в асимптотической формуле. Это означает, например, что поведение функции $L(m, n, k)$ существенно не изменится, если наряду со стандартными элементами задержки будут применяться еще и одновходные триггеры, или элементы, порождающие периодические последовательности. Выбор того или иного базисного набора элементов может сильно отражаться на стоимости реализации операторов с небольшими параметрами m, n, k . При росте же этих параметров действует асимптотическая закономерность (7.5).

Предельная теорема является непосредственным следствием двух теорем: теоремы о верхней оценке и теоремы о нижней оценке, которые справедливы при тех же условиях, налагаемых на базисные элементы, что и в предельной теореме.

Первая из названных выше теорем устанавливает тот факт, что, как бы ни было мало фиксированное число $\varepsilon > 0$, существуют такие операторы $\theta \in \theta_{m, n, k}$, для которых

*) C. E. Shannon, The synthesis of two-terminal switching circuits. BSTJ, 1949, v. 28, № 1, p. 59—98.

**) О. Б. Лупанов, О возможности синтеза схем из произвольных элементов. Труды матем. ин-та им. Стеклова, т. 51, 1958.

***) О. Б. Лупанов, Об одном методе синтеза схем. Изв. высш. учебн. завед. Радиофизика, № 1, 1958.

****) Б. А. Трахтенброт, Асимптотическая оценка сложности логических сетей с памятью. ДАН СССР, т. 127, № 2, 1959.

справедливо:

$$L(\theta) \geq (1 - \varepsilon) \left[nq \frac{2^m \cdot 2^k}{m+k} + kq \frac{(2^m - 1) 2^k}{m+k} \right],$$

то есть:

$$L(\theta) \geq (1 - \varepsilon) q \frac{\lg N(m, n, k)}{\lg \lg N(m, n, k)}. \quad (7.9)$$

В действительности, однако, справедливо гораздо более сильное утверждение, составляющее содержание *теоремы о нижней оценке*.

Теорема о нижней оценке. Для любого фиксированного $\varepsilon > 0$ отношение числа операторов из класса $\theta_{m, n, k}$, для которых справедливо соотношение (7.9), к числу всех операторов из $\theta_{m, n, k}$ стремится к 1, коль скоро $m+k$ стремится к бесконечности, а отношение $\frac{\lg n}{m+k}$ стремится к нулю.

Таким образом, неравенство (7.9) справедливо для подавляющего большинства операторов из класса $\theta_{m, n, k}$.

Теорема о верхней оценке. Каково бы ни было $\varepsilon > 0$, существует такое $v > 0$, что при $m+k > v$ для любого оператора $\theta \in \theta_{m, n, k}$ имеет силу неравенство:

$$L(\theta) < (1 + \varepsilon) \left[nq \frac{2^m \cdot 2^k}{m+k} + kq \frac{(2^m - 1) 2^k}{m+k} \right]. \quad (7.10)$$

Доказательство этой теоремы заключается в описании эффективного метода синтеза (не сводящегося к перебору, как это имело место в случае тривиального алгоритма), который при больших $m+k$ (при $m+k \rightarrow \infty$) строит для любого $\theta \in \theta_{m, n, k}$ логическую сеть, стоимость которой удовлетворяет неравенству (7.10).

Сопоставление этих двух теорем не только обосновывает предельную формулу (7.5), но и устанавливает нечто большее. Именно, оказывается, что почти для всех операторов $\theta \in \theta_{m, n, k}$ функция $L(\theta)$ близка к функции $L(m, n, k)$, то есть почти все операторы из класса $\theta_{m, n, k}$ столь же сложно синтезируются, как и самый сложный оператор из класса $\theta_{m, n, k}$. Вместе с тем метод синтеза, гарантирующий верхнюю оценку (7.10), оказывается для почти всех этих операторов почти оптимальным. Таким образом, этот метод синтеза решает в некотором смысле задачу построения оптимальной сети. Правда,

полное решение этой задачи требует большего, а именно: требует такого метода синтеза, который для всех операторов позволял бы выявлять в точности оптимальные реализации.

Однако в свете последних результатов С. В. Яблонского становится правдоподобным, что полное решение в таком понимании невозможно иным способом, нежели тривиальный перебор множества сетей *).

В последующих параграфах основное внимание будет уделено изложению различных методов синтеза автоматов, на основе которых в конечном счете (§ 6) удастся получить теорему о верхней оценке. Доказательство теоремы о нижней оценке, а также доказательство формулы (7.6) мы не приводим, хотя и будем в дальнейшем прибегать к ссылкам на указанную теорему и формулу.

§ 2. Синтез сети, реализующей функцию алгебры логики; верхняя оценка сложности сети

В этом параграфе рассматриваются различные эффективные **) методы синтеза сетей, реализующих функции алгебры логики $f(x_1, \dots, x_m)$, и оценивается стоимость этих сетей при $m \rightarrow \infty$. Для каждого метода A обозначим через $L_A(m)$ наименьшее L , такое, что метод A доставляет для любой функции от m переменных реализующую ее сеть со стоимостью, не превышающей L . Метод B естественно считать сильнее метода A , если $L_B(m) \leq L_A(m)$ для всех m , начиная с некоторого m_0 . Исследуемые ниже методы располагаются в порядке роста их силы; самым сильным из них оказывается метод Лупанова, гарантирующий наилучшую возможную оценку в смысле предельной теоремы.

Условимся о некоторых терминах и обозначениях, которые будут применяться как в этом, так и в последующих параграфах. Выражения «&-ячейка», « \vee -ячейка», « \neg -ячейка» означают сети, реализующие соответственно функции $x_1 \& x_2$, $x_1 \vee x_2$, \bar{x} .

*) Яблонский С. В., Об алгоритмических трудностях синтеза минимальных контактных схем, Сб. «Проблемы кибернетики», № 2, Физматгиз, М., 1959.

**) Эффективные — в смысле: не сводящиеся к перебору множества сетей.

В частном случае некоторые (или даже все) ячейки такого рода могут оказаться элементами из рассматриваемого нами базиса. В общем же случае эти ячейки представляют собой некоторые блоки, построенные из элементов базиса; существование таких блоков обеспечивается полнотой базисного набора элементов. Стоимости этих ячеек мы будем обозначать так: $L_{\&}$, L_{\vee} , L_{\neg} .

Ниже, в пунктах 1°, 2°, 3°, 4°, мы рассмотрим методы синтеза сетей из \vee -, $\&$ - и \neg -ячеек (что соответствует обычному заданию функций алгебры логики посредством операций дизъюнкции, конъюнкции и отрицания). В пункте 5° будет рассмотрен самый общий случай, учитывающий специфику элементов, входящих в базис. В пункте 6° результаты, установленные ранее для функций алгебры логики, распространяются на системы таких функций (общий случай истинностного оператора).

1°. Моделирование совершенной дизъюнктивной нормальной формы (с.д.н.ф.). Реализация функции $f(x_1, x_2, \dots, x_m)$, исходя из ее с.д.н.ф., может потребовать не более m \neg -ячеек (для формирования отрицаний $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m$), не более $(m-1) \cdot 2^m$ $\&$ -ячеек (формирующих все члены вида $x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m}$) и не более 2^m \vee -ячеек. Поэтому стоимость непосредственного моделирования формулы подчиняется неравенству:

$$L_{1^\circ}(f) < mL_{\neg} + (m-1)2^m L_{\&} + 2^m L_{\vee}.$$

При $m \rightarrow \infty$ очевидно неравенство:*)

$$L_{1^\circ}(f) < m2^m L_{\&} [1 + o(1)]. \quad (7.11)$$

Эта оценка может быть улучшена, если учесть, что либо сама функция f , либо ее отрицание \bar{f} имеет с.д.н.ф.

*) Напомним обозначения:

$$\alpha = o(\beta) \text{ означает, что } \frac{\alpha}{\beta} \rightarrow 0,$$

$$\alpha = O(\beta) \text{ означает, что для некоторых}$$

констант m , M ($0 < m < M$) имеет место $m < \frac{\alpha}{\beta} < M$. В частности, $o(1)$ обозначает величину, стремящуюся к нулю в данном процессе.

с числом конъюнктивных членов $\leq \frac{2^m}{2}$ (см. гл. I). Моделирование f или \bar{f} (в зависимости от того, какая из них имеет более короткую с.д.н.ф.) с последующим взятием отрицания (во втором случае) снижает оценку (7.11) вдвое:

$$L_{1^0}(f) < \frac{m \cdot 2^m}{2} L_{\&} [1 + o(1)]. \quad (7.11')$$

2°. Применение универсального дешифратора. В методе 1° основная затрата стоимости падает на $\&$ -ячейки, необходимые для формирования всех конъюнкций вида $x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m}$. Сеть, реализующую все эти 2^m конъюнкций, мы будем называть *универсальным дешифратором* (у.д.) от m переменных x_1, x_2, \dots, x_m . Более экономный характер реализации универсального дешифратора (сравнительно с формированием в отдельности 2^m членов) явствует из следующего (см. также гл. VI).

Число $\&$ -ячеек, необходимых для реализации (в отдель-

ности) членов $x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m}$ и $x_m^{\frac{\sigma_m}{2} + 1} \dots x_m^{\sigma_m}$, не более чем

$$\frac{m}{2} \cdot 2^{\frac{m}{2}} + \frac{m}{2} \cdot 2^{\frac{m}{2}} = m \cdot 2^{\frac{m}{2}}.$$

Для получения «длинных» конъюнкций понадобятся еще не более чем 2^m $\&$ -ячеек (соединяющих всевозможные пары коротких конъюнкций). Итак, при $m \rightarrow \infty$:

$$L(\text{у.д.}) < (m \cdot 2^{\frac{m}{2}} + 2^m) L_{\&} + mL_{\neg} = 2^m L_{\&} [1 + o(1)]. \quad (7.12)$$

Для реализации функции $f(x_1, \dots, x_m)$ при наличии универсального дешифратора могут потребоваться еще не более чем 2^m \vee -ячеек.

Следовательно:

$$L_{2^0}(m) < 2^m [L_{\&} + L_{\vee}] [1 + o(1)]. \quad (7.13)$$

3°. Метод Шеннона*). Дальнейшее улучшение метода синтеза, приводящее к значительному понижению

*) С. E. Shannon, The synthesis of two-terminal [switching circuits. BSTJ, 1949, v. 28, № 1, p. 59-98.

оценки (7.13), может быть достигнуто путем разложения функции f .

Пусть число переменных $m = l + n$. Исследуем разложение (см. гл. I):

$$f(x_1, \dots, x_l, y_1, \dots, y_n) = \bigvee x_1^{\sigma_1} \dots x_l^{\sigma_l} f(\sigma_1, \dots, \sigma_l, y_1, \dots, y_n) \quad (7.14)$$

при $l \rightarrow \infty, n \rightarrow \infty$.

Для реализации конъюнкций вида $x_1^{\sigma_1} x_2^{\sigma_2} \dots x_l^{\sigma_l}$ будет израсходовано, согласно (7.12), не более $2^l [1 + o(1)]$ &-ячеек. Различных функций $f(\sigma_1, \dots, \sigma_l, y_1, \dots, y_n)$ существует не более 2^{2^n} , и реализация каждой стоит не более $2^n (L_{\&} + L_{\vee}) [1 + o(1)]$. Кроме того, потребуется еще 2^l &-ячеек и столько же \vee -ячеек для образования правой части соотношения (7.14).

Итак, мы имеем:

$$L_{3^{\circ}}(m) < [2^{l+1} L_{\&} + 2^l L_{\vee} + 2^{2^n} \cdot 2^n (L_{\&} + L_{\vee})] [1 + o(1)]. \quad (7.15)$$

Правая часть неравенства (7.15) зависит от способа разделения числа m на слагаемые l и n . Это разложение следует осуществлять так, чтобы минимизировать оценку для функции $L(m)$. Если брать $n = \lg(m - 3 \lg m)$, $l = m - \lg[m - 3 \lg m]$, то правая часть неравенства (7.15) примет вид: *)

$$\frac{2^m}{m - 3 \lg m} [2L_{\&} + L_{\vee}] + \frac{2^m}{m^3} (m - 3 \lg m) (L_{\&} + L_{\vee}) [1 + o(1)].$$

Следовательно,

$$L_3(m) < \frac{2^m}{m} [2L_{\&} + L_{\vee}] [1 + o(1)]. \quad (7.16)$$

Этот метод синтеза применен впервые К. Шенноном в теории контактных схем. Как видно из вышеизложенного, при применении метода Шеннона понижение стоимости сети достигается за счет двух факторов:

а) за счет экономной реализации универсального дешифратора от l переменных с общей стоимостью $\leq 2^l [1 + o(1)] L_{\&}$ в соответствии с пунктом 2° этого параграфа;

* Здесь, как и в дальнейшем, допуская некоторую вольность, мы берем логарифм вместо ближайшего целого числа, которое не меньше этого логарифма. Для наших оценок такая замена неуместивна.

б) за счет разложения реализуемой функции по некоторым из переменных; при этом число этих переменных подбирается надлежащим образом:

$$l = m - \lg(m - 3 \lg m).$$

4°. Перейдем теперь к рассмотрению *метода Лупанова**), дающего еще более экономную реализацию, чем метод Шеннона.

Таблица VII.1

	0	τ_1	1	x_1
	\vdots	\vdots	\vdots	\vdots
	0	τ_λ	1	x_λ
	0	σ_1	1	v_1
	\vdots	\vdots	\vdots	\vdots
$y_1 y_2 \dots y_v$	0	σ_μ	1	v_μ
0 0 ... 0	A_1			
	A_2			
	\vdots			
	\vdots			
$\eta_1 \eta_2 \dots \eta_v$	$f(\tau_1, \dots, \tau_\lambda, \sigma_1, \dots, \sigma_\mu, \eta_1, \dots, \eta_v)$			
	\vdots			
	\vdots			
1, 1, ... 1	A_p			

$$f(\tau_1, \dots, \tau_\lambda, \sigma_1, \dots, \sigma_\mu, \eta_1, \dots, \eta_v) = f(\tau, \sigma, \eta)$$

Метод Лупанова мы изложим сначала применительно к тому частному случаю, когда сеть строится, как и в предыдущих случаях, из $\&$ -, \vee - и \neg -ячеек. Затем уже рассмотрим общий случай, когда сеть строится из элементов произвольного базиса. Такое изложение облегчает понимание роли отдельных этапов построения.

*) О. Б. Лупанов, Об одном методе синтеза схем. Изв. высш. учебн. завед., Радиофизика, № 1. 1958.

Разобьем m переменных функции f на три группы

$$x_1, x_2, \dots, x_\lambda,$$

$$v_1, v_2, \dots, v_\mu,$$

$$y_1, y_2, \dots, y_\nu.$$

(где $\lambda + \mu + \nu = m$) и зададим f посредством матрицы, изображенной на таблице VII.1; строки матрицы занумерованы наборами значений переменных y_1, \dots, y_ν , а столбцы — наборами значений переменных $x_1, \dots, x_\lambda, v_1, \dots, v_\mu$, (ср. табл. I.10, в которой строкам соответствуют переменные x_1, x_2 , а столбцам — переменные x_3, x_4, x_5).

Там, где это не вызывает недоразумений, мы будем писать x вместо группы переменных x_1, \dots, x_λ ; точно так же v, y, τ будут обозначать другие группы переменных или наборы их значений. Так будут получаться такие выражения, как $f(x, v, y)$, $f(\tau, v, y)$, где τ есть набор $\tau_1, \dots, \tau_\lambda$, и т. п.

Первый этап построения сети. Строки матрицы разбиваются на группы A_1, A_2, \dots, A_p , по S строк в каждой (за исключением, быть может, одной группы, например группы A_p , которая будет содержать меньшее число строк в том случае, когда 2^l не делится нацело на S). Пусть $f_i(x, v, y)$ есть функция, совпадающая с $f(x, v, y)$ на группе A_i и равная нулю на остальных строках. Тогда имеет место очевидное представление

$$f(x, v, y) = \bigvee_{i=1}^p f_i(x, v, y); \quad \left(p < \frac{2^\nu}{S} + 1\right). \quad (7.17)$$

Примечание 1. Пусть $\tau_1, \dots, \tau_\lambda, \sigma_1, \dots, \sigma_\mu$ суть какие-нибудь фиксированные значения (0, или 1), подставляемые вместо переменных $x_1, \dots, x_\lambda, v_1, \dots, v_\mu$. В столбце значений любой из функций $f_i(\tau_1, \dots, \tau_\lambda, \sigma_1, \dots, \sigma_\mu, y_1, \dots, y_\nu)$ не более чем S единиц*). Следовательно: 1) в ее с.н.д.ф. существует не более чем S конъюнктивных членов; 2) при данном i число функций $f_i(\tau, \sigma, y) \leq 2^S$, а общее их число не более чем $p \cdot 2^S \leq \left(\frac{2^\nu}{S} + 1\right) \cdot 2^S$.

*) Ибо единицы могут находиться только на пересечении этого столбца с группой A_i .

Второй этап построения сети заключается в разложении каждой из функций $f_i(x, v, y)$ по переменным x :

$$f_i(x, v, y) = \bigvee_{\tau_1, \dots, \tau_\lambda} x_1^{\tau_1} \dots x_\lambda^{\tau_\lambda} f_i(\tau_1, \dots, \tau_\lambda, v, y). \quad (7.18)$$

Следовательно:

$$f(x, v, y) = \bigvee_{i=1}^p \bigvee_{\tau_1, \dots, \tau_\lambda} x^{\tau} f_i(\tau, v, y). \quad (7.19)$$

Примечание 2. Общее число всех функций $f_i(\tau, v, y)$ не превосходит $p \cdot 2^\lambda \leq \left(\frac{2^v}{S} + 1\right) 2^\lambda$.

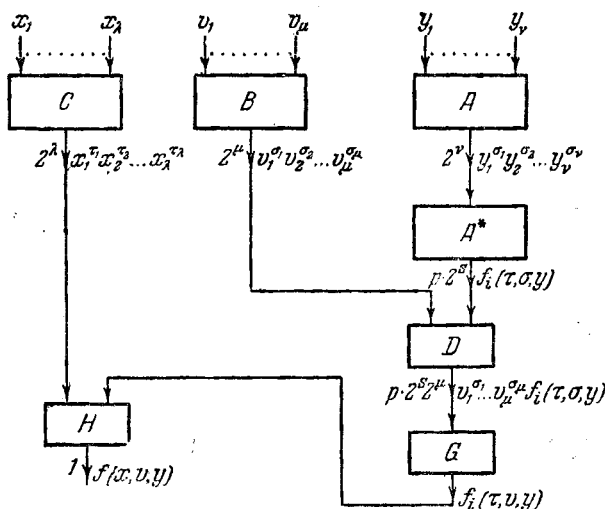


Рис. 7.1.

Третий этап. Для каждой из функций $f_i(\tau, v, y)$ рассматривается разложение по переменным v :

$$f_i(\tau, v, y) = \bigvee_{\sigma_1, \dots, \sigma_\mu} v_1^{\sigma_1} \dots v_\mu^{\sigma_\mu} f(\tau, \sigma_1, \dots, \sigma_\mu, y). \quad (7.20)$$

В соответствии с этим представлением схема, реализующая функцию f , строится из отдельных блоков (рис. 7.1). Входы каждого из них либо присоединены

к выходам блоков, построенных ранее, либо являются входами всей схемы. Ниже указаны функциональные характеристики этих блоков (см. рис. 7.1) и оценка их стоимости.

Блоки A , B , C являются универсальными дешифраторами по группам переменных y , v , x соответственно. Для их стоимостей, в предположении, что $v \rightarrow \infty$ и $\mu \rightarrow \infty$ и $\lambda \rightarrow \infty$, мы имеем следующие оценки (см. (7.12)):

$$L(A) < 2^v L_{\&} [1 + o(1)], \quad L(B) < 2^\mu L_{\&} [1 + o(1)],$$

$$L(C) < 2^\lambda L_{\&} [1 + o(1)].$$

Блок A^* реализует все функции вида $f_i(\tau, \sigma, y)$, число которых не больше $\left(\frac{2^v}{S} + 1\right) \cdot 2^S$ (ср. примечание 1). Каждая функция реализуется как дизъюнкция соответствующих конъюнкций, поступающих из A ; для этого требуется не более $S \vee$ -ячеек (ср. примечание 1).

Итак, мы имеем:

$$L(A^*) < \left(\frac{2^v}{S} + 1\right) 2^S S L_{\vee}.$$

Блок D реализует всевозможные функции вида

$$v_1^{\sigma_1} v_2^{\sigma_2} \dots v_\mu^{\sigma_\mu} f(\tau, \sigma_1, \dots, \sigma_\mu, y_1, \dots, y_v)$$

посредством конъюнктивного присоединения функций, реализуемых в блоке A^* , к функциям, реализуемым в дешифраторе B . Общее число таких функций не превосходит произведения числа всех функций из A^* на число всех функций из B .

Следовательно:

$$L(D) < \left(\frac{2^v}{S} + 1\right) 2^S \cdot 2^\mu L_{\&}.$$

Блок G реализует все функции $f_i(\tau, v, y)$, число которых не превосходит $\left(\frac{2^v}{S} + 1\right) 2^\lambda$ (ср. примечание 2), исходя из функций, реализуемых блоком D . Для реализации каждой функции $f_i(\tau, v, y)$ требуется не более $2^\mu \vee$ -ячеек.

Следовательно:

$$L(G) < \left(\frac{2^v}{S} + 1\right) 2^\lambda \cdot 2^\mu L_{\vee}.$$

Блок H реализует функцию f (см. (7.19)), исходя из функций блоков C и G . Он содержит не более $\left(\frac{2^v}{S} + 1\right) 2^\lambda$ $\&$ -ячеек для реализации каждой конъюнкции вида $x_1^{\tau_1} \dots x_\lambda^{\tau_\lambda} \cdot f_i(\tau_1, \dots, \tau_\lambda, v, y)$ и не более $\left(\frac{2^v}{S} + 1\right) 2^\lambda$ \vee -ячеек для реализации из этих конъюнкций функции f .

Следовательно:

$$L(H) < \left(\frac{2^v}{S} + 1\right) 2^\lambda (L_\& + L_\vee).$$

Оценим теперь стоимости отдельных блоков при $m \rightarrow \infty$ и при условии, что для параметров v, μ, λ, S взяты следующие значения:

$$\left. \begin{aligned} S &= [m - 5 \lg m], & \mu &= [\lg m], \\ v &= [2 \lg m], & \lambda &= m - v - \mu. \end{aligned} \right\} \quad (7.21)$$

Для блока G легко получается следующая оценка:

$$L(G) < \frac{2^m}{m} [1 + o(1)] L_\vee;$$

вместе с тем оказывается, что каждая из величин

$$L(A), L(B), L(C), L(A^*), L(D), L(H),$$

а значит и их сумма, есть $o\left(\frac{2^m}{m}\right)$.

Следовательно, для всей схемы получается оценка:

$$L(f) < \frac{2^m}{m} [1 + o(1)] L_\vee. \quad (7.22)$$

Эта оценка существенно лучше той, которую дает метод Шеннона.

5°. Опишем теперь общий случай, когда состав элементов в базе произволен. Заметим прежде всего, что в построении, рассмотренном в предыдущем пункте, вся стоимость приходится в основном за счет блока G , который совместно с блоком D осуществляет разложение (7.20) функций f_i по переменным v_1, v_2, \dots, v_μ . При этом блок G соответствует внешним функциям разложения *)

*) См. гл. I, § 4.

(в рассматривавшемся нами случае это были 2^μ -членные дизъюнкции), а блок D соответствует сопряженным функциям (то есть в нашем случае — конъюнкциям). Это подсказывает нам, что в общем случае целесообразнее вместо (7.20) использовать разложение следующего вида (ср. 1.14):

$$\begin{aligned} f_i(\tau, v, y) = & F\{\Psi_0[v_1, \dots, v_\mu, f_i(\tau, 0, \dots, 0, y)], \\ & \Psi_1[v_1, \dots, v_\mu, f_i(\tau, 0, \dots, 0, 1, y)], \dots \\ & \dots, \Psi_{2^\mu-1}[v_1, \dots, v_\mu, f_i(\tau, 1, \dots, 1, 1, y)]\}, \\ & X_{2^\mu}(v_1, \dots, v_\mu), \dots, X_{N-1}(v_1, \dots, v_\mu)\}, \quad (7.23) \end{aligned}$$

при котором внешняя функция разложения F допускает самую экономную реализацию.

Пусть, например, ячейка Φ , реализующая функцию $\Phi(x_1, \dots, x_r)$, существенно зависящую от r переменных, имеет стоимость L_Φ , причем ее удельная стоимость q , равная

$\frac{L_\Phi}{r-1}$, является наименьшей среди удельных стоимостей элементарных ячеек. Внешнюю функцию разложения F построим путем многократной суперпозиции этой функции Φ . Поскольку кратность этой суперпозиции $\leq \frac{2^\mu}{r-1} + 1$, то для стоимости L_F блока T_F , реализующего функцию F , исходя из ячеек Φ , мы имеем:

$$L_F < \left(\frac{2^\mu}{r-1} + 1 \right) L_\Phi = q \cdot 2^\mu + L_\Phi.$$

Если среди элементов базиса нет \neg -ячеек, $\&$ -ячеек или \vee -ячеек, то в силу полноты базиса можно изготовить соответствующие \neg -блоки, $\&$ -блоки и \vee -блоки, для стоимости которых мы сохраним обозначения L_\neg , $L_\&$, L_\vee . Поэтому можно воспроизвести — полностью или частично — построение предыдущего пункта, сохранив полученные там оценки; в частности, это относится к крупным блокам A , B , C , A^* , H , построение которых мы оставим в прежнем виде. Что касается блоков D и G , то вместо них мы построим блоки D' и G' , которые совместно должны реализовать все функции вида $f_i(\tau, v, y)$, в соответствии с представлением (7.23).

Блок D' реализует все функции $\Psi_i[v_1, \dots, v_\mu, f_i(\tau, \sigma_1, \dots, \sigma_n, y)]$, число которых *) не превышает $p2^S \cdot 2^\mu$, и все функции $\chi_i(v_1, \dots, v_\mu)$, число которых не более r^{**}). Если функции Ψ_i от $\mu + 1$ переменной и функции χ_i от μ переменных реализовать так, как указано в пункте 2°, то стоимость каждой из них будет не более чем $2^{\mu+1} [1 + o(1)] L_\&$. Поэтому справедливо следующее неравенство:

$$L(D') < p \cdot 2^S 2^\mu 2^{\mu+1} [1 + o(1)] L_\&.$$

Блок G' реализует все функции вида $f_i(\tau, v, y)$, исходя из функций, реализуемых блоком D' . Для реализации каждой функции $f_i(\tau, v, y)$ (см. соотношение (7.23)) требуется один блок T_F . Поэтому

$$L(G') < p \cdot 2^\lambda L_F < \left(\frac{2^v}{S} + 1\right) 2^\lambda [q \cdot 2^\mu + L_\Phi].$$

Сохраняя для параметров λ, μ, v, S значения (7.21), мы при $m \rightarrow \infty$ получаем оценку $L(G') < q \frac{2^m}{m} [1 + o(1)]$, в то время как $L(D') = o\left(\frac{2^m}{m}\right)$. Поскольку $L(A), L(B), L(C), L(A^*), L(H)$ по-прежнему являются $o\left(\frac{2^m}{m}\right)$, то мы имеем окончательно:

$$L(f) < q \frac{2^m}{m} [1 + o(1)], \quad (7.24)$$

чем и установлена теорема Лупанова.

П р и м е ч а н и е. В частном случае, если среди элементов базиса с минимальной удельной стоимостью q оказались бы также и \vee -ячейки, мы имели бы $q = \frac{L_V}{2-1} = L_V$;

в таком случае оценка (7.24) принимает вид (7.22).

6°. Рассмотрим теперь функцию $L(m, n) = L(m, n, 0)$, т. е. функцию, значение которой таково: наименьшее число L такое, что любой истинностный оператор $\theta \in \Theta_{m,n,0}$ может быть реализован логической сетью, имеющей сто-

*) Так же как и число функций, реализуемых блоком D .

**) Ибо в силу способа построения F из $\Phi(x_1, \dots, x_r)$ ясно, что число переменных в F превосходит 2^μ не более, чем на r .

имость, не превышающую L . Оператор, $\theta \in \theta_{m,n,0}$ определяется заданием системы n функций алгебры логики от m переменных каждая. Простейший способ реализации оператора θ заключается в раздельной реализации каждой из n функций-компонент $z_i = f_i(x_1, \dots, x_m)$, где $i=1, 2, \dots, n$, каким-либо из рассмотренных ранее методов. В частности метод Лупанова в этом случае применим с оценкой:

$$L(\theta) < n \frac{q \cdot 2^m}{m} [1 + o(1)]. \quad (7.25)$$

При изложении практических приемов синтеза многовыходной сети (см. гл. VI) было указано на то, что часто удается получить экономную сеть за счет объединения общих частей сетей, реализующих функции-компоненты, что возможно, например, посредством применения метода каскадов. Это обстоятельство наводит на мысль о том, что оценка (7.25) является завышенной, а также что улучшение этой оценки можно получить посредством отказа от принципа раздельной реализации и выработки надлежащих принципов эффективной совместной реализации функций f_i (где $i \leq n$).

Для отдельных подклассов класса всех истинностных операторов дело обстоит именно так. (В частности, это относится, например, к подклассу монотонных операторов, который рассматривается в следующем параграфе.)

Однако для класса всех истинностных операторов, при весьма общем ограничении, наложенном на рост параметров m, n , оценка (7.25) не может быть улучшена. Это ограничение сводится к требованию $\lim_{m \rightarrow \infty} \frac{\lg n}{m} = 0$, являющемуся частным случаем (при $k=0$) условия (7.4), естественность которого была мотивирована в § 1.

Действительно, при выполнении этого условия общая теорема об оценке снизу устанавливает, что с ростом m доля тех истинностных операторов, для которых

$$L(\theta) \geq q \frac{2^{m \cdot n}}{m} [1 - \varepsilon], \quad (7.25')$$

стремится к 1 ($\varepsilon > 0$ есть произвольное фиксированное число).

Таким образом:

$$L(m, n) \sim q \cdot \frac{n \cdot 2^m}{m} \quad \left(\text{при } \frac{\lg m}{n} \rightarrow 0, m \rightarrow \infty \right). \quad (7.25'')$$

§ 3. Операторы, допускающие простую реализацию

Тот факт, что оценки экспоненциального типа, установленные в предыдущем параграфе, не допускают улучшения, может привести к мало обнадеживающим выводам о невозможности получить простые реализации для подавляющего большинства мыслимых ограниченно-детерминированных операторов. В определенном смысле дело обстоит именно так. Однако при решении практических задач приходится преимущественно иметь дело не с любыми, наугад взятыми операторами, а со сравнительно небольшим запасом простых операторов, обладающих достаточно обозримой внутренней структурой. Для подобных операторов удается обычно находить сравнительно несложные реализации, зачастую со степенной (а в частных случаях и с линейной) относительно m и n оценкой. Выработка разнообразных частных методов синтеза применительно к отдельным классам операторов, имеющим практическое значение, является поэтому важной задачей. Значительные результаты в этом направлении получены С. В. Яблонским*), М. А. Гавриловым**), Г. Н. Поваровым***), О. Б. Лупановым и другими авторами.

В этом параграфе рассматриваются некоторые специальные классы истинностных операторов, реализуемых с оценкой стоимости существенно ниже той, которая установлена для класса всех истинностных операторов. При этом основная цель, которая здесь преследуется, заключается не в получении точных и окончательных асимптотических оценок, а в обзоре некоторых типичных ситуаций, в которых удается получать простые реализации.

1°. Пусть некоторый детерминированный оператор перерабатывает входные слова длины μ вида $x(1), x(2), \dots, x(\mu)$ в алфавите $\{0, 1\}$ в выходные слова $z(1), z(2), \dots, z(\mu)$. При фиксированном μ может быть поставлена

*) С. В. Яблонский, Об одном семействе классов функций, допускающих простую схемную реализацию, УМН, т. 12, 1957.

**) М. А. Гаврилов, Теория релейно-контактных схем. Изд. АН СССР, М.—Л., 1950.

***) Г. Н. Поваров, Новый метод синтеза симметричных контактных схем. ДАН УССР, т. 2, 1955, стр. 115—117.

задача о реализации этого преобразования информации в устройстве, воспринимающем одновременно по входным двоичным каналам слово x_1, x_2, \dots, x_μ и вырабатывающем слово z_1, \dots, z_μ *).

Таким образом, мы приходим к рассмотрению истинностного оператора $\langle z_1, \dots, z_\mu \rangle = \theta [x_1, \dots, x_\mu]$, характеризующегося тем, что z_i не зависит (существенно) от переменных $x_{i+1}, x_{i+2}, \dots, x_\mu$:

$$\begin{aligned} z_1 &= \varphi_1(x_1), \\ z_2 &= \varphi(x_1, x_2), \\ &\dots \dots \dots \\ z_\mu &= \varphi_\mu(x_1, x_2, \dots, x_\mu). \end{aligned}$$

В таком случае естественно ожидать, что раздельная реализация функций-компонент $\varphi_1, \varphi_2, \dots, \varphi_\mu$ пройдет с лучшей оценкой, чем оценка (7.25). Покажем, что это действительно имеет место.

Пусть $n < \mu$; тогда каждая из n функций $\varphi_1, \dots, \varphi_n$ зависит не более чем от n переменных, а каждая из остальных l функций (где $l = \mu - n$): $\varphi_{n+1}, \dots, \varphi_\mu$ — зависит от $n+1, n+2, \dots, \mu$ переменных соответственно. Положим $l = \lg \mu^2$, $n = \mu - \lg \mu^2$ и осуществим раздельную реализацию функций $\varphi_1, \dots, \varphi_\mu$. Тогда при $\mu \rightarrow \infty$ реализация осуществляется со следующей оценкой:

$$L(\theta) < \left[nq \frac{2^n}{n} + q \frac{2^{\mu - \lg \mu^2}}{\mu - \lg \mu^2} + \dots + q \frac{2^{\mu-1}}{\mu-1} + q \frac{2^\mu}{\mu} \right] (1 + o(1)),$$

а следовательно:

$$L(\theta) < \left[q \frac{2^\mu}{\mu^2} + q \frac{2^\mu + 2^{\mu-1} + \dots}{\mu - \lg \mu^2} \right] (1 + o(1));$$

и окончательно:

$$L(\theta) < \frac{q \cdot 2^{\mu+1}}{\mu} \cdot [1 + o(1)], \quad (7.26)$$

что существенно лучше, чем оценка (7.25) **).

*) Так обстоит, например, дело, когда вместо сумматора последовательного действия берется сумматор параллельного действия.

**) Имеет место также соответствующая теорема о нижней оценке, из которой следует, что оценка (7.26) не может быть улучшена.

Ясно, что изложенные выше соображения не связаны именно с «параллельной» реализацией детерминированного оператора, а относятся вообще к тому случаю, когда в операторе $\langle z_1, \dots, z_n \rangle = \theta [x_1, \dots, x_m]$ (не обязательно $n = m$) z_i не зависит от всех переменных x_j , а лишь от некоторого их числа $S(i)$. В зависимости от того, каковы

эти числа $S(i)$, могут получаться различные упрощенные оценки типа (7.26).

Рис. 7.2.

2°. В предыдущем пункте не был учтен вес оператора (точнее вес конечного дерева высоты μ). Предположим, что вес, о котором идет речь, не превышает 2^k , и следовательно, оператор может быть описан каноническими уравнениями:

$$\left. \begin{aligned} z(t) &= \Phi[x(t), q_1(t), \dots, q_k(t)], \\ q_j(t+1) &= \Psi_j[x(t), q_1(t), \dots, q_k(t)]. \end{aligned} \right\} \quad (7.27)$$

Переход от схемы последовательного действия, реализующей оператор (7.27) (изображена на рис. 7.2), к схеме

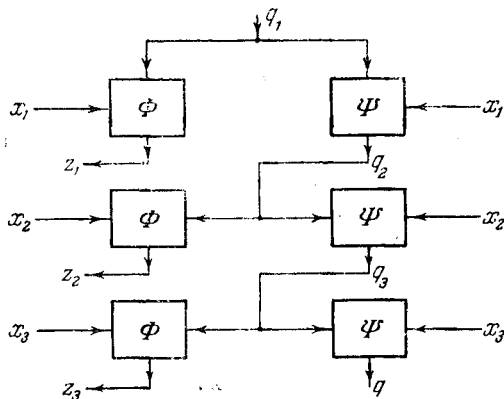


Рис. 7.3.

параллельного действия, реализующей соответствующий истинностный оператор $\langle z_1, \dots, z_\mu \rangle = \theta [x_1, \dots, x_\mu]$ (изображена на рис. 7.3), понятен из самих рисунков. Для про-

стоты мы на этих рисунках взяли $k = 1$ и $\mu = 3$; в этом случае (7.27) принимает вид:

$$\left. \begin{aligned} z(t) &= \Phi[x(t), q(t)], \\ q(t) &= \Psi[x(t), q(t)], \end{aligned} \right\} \quad (7.27')$$

а соответствующий истинностный оператор описывается условиями, не содержащими переменной t :

$$z_1 = \Phi[x_1, q_1], \quad z_2 = \Phi[x_2, q_2], \quad z_3 = \Phi[x_3, q_3],$$

$$q_2 = \Psi[x_1, q_1], \quad q_3 = \Psi[x_2, q_2], \dots$$

Если стоимость схемы, изображенной на рис. 7.2 (не считая элемента задержки), обозначить через C , то стоимость схемы, изображенной на рис. 7.3, будет равна $C \cdot \mu$, причем C есть константа, не зависящая от μ . Можно сказать, что сокращение времени работы в μ раз сопряжено с μ -кратным увеличением оборудования.

Таким образом, истинностные операторы

$$z_1 = \theta_1[x_1],$$

$$\langle z_1, z_2 \rangle = \theta_2[x_1, x_2],$$

$$\dots \dots \dots$$

$$\langle z_1 z_2 \dots z_\mu \rangle = \theta_\mu[x_1, x_2, \dots, x_\mu] \dots$$

и т. д., соответствующие деревьям высоты $1, 2, \dots, \mu, \dots$ заданного оператора с конечным весом, реализуются с линейной относительно μ оценкой стоимости. Разумеется, это справедливо и в том случае, когда входными и выходными буквами исходного оператора являются не просто 0 и 1, а наборы из 1 и 0.

П р и м е р ы.

А. Определить оценку стоимости логической сети оператора $z = x + y$, перерабатывающего двоичные записи μ -разрядных чисел x, y в младшие μ разрядов их суммы (заметим, что одновременно с реализацией этого оператора реализуется и оператор переноса). При решении этой задачи будем исходить из того, что для двоичных цифр суммы (z_1, z_2, \dots, z_μ) и переноса (S_1, S_2, \dots, S_μ) имеются

рекуррентные соотношения:

$$z_1 = x_1 \oplus S_1 \oplus y_1, \quad z_2 = x_2 \oplus S_2 \oplus y_2,$$

$$\dots$$

$$z_\mu = x_\mu \oplus S_\mu \oplus y_\mu,$$

$$S_1 = 0, \quad S_2 = x_1 (y_1 \vee S_1) \vee y_1 S_1,$$

$$\dots$$

$$S_\mu = x_{\mu-1} (y_{\mu-1} \vee S_{\mu-1}) \vee y_{\mu-1} \cdot S_{\mu-1}.$$

Из этих соотношений непосредственно усматривается линейная оценка стоимости сети рассматриваемого оператора с коэффициентом $C < 2(L_\oplus + L_\& + L_\vee)$.

В частности, для оператора $z = x + 1$ мы имеем:

$$z_i = x_i \oplus S_i, \quad S_{i+1} = x_i S_i, \quad S_1 = 1 \quad (\text{где } i = 1, 2, \dots, \mu).$$

Следовательно, $C < L_\oplus + L_\&$. Аналогично обстоит дело и с оператором $z = x - y$.

Б. Рассмотрим схему параллельного действия для сравнения чисел. Она должна реализовать функцию $z_n = f_n(x_1, \dots, x_n, y_1, \dots, y_n)$, равную 1, в том и только в том случае, если для чисел x, y с двоичными записями $x_n x_{n-1} \dots x_1$ и $y_n y_{n-1} \dots y_1$ справедливо $x < y$. Исходя из соответствующего оператора с памятью

$$z(t) = \bar{x}(t) \cdot y(t) \vee [\bar{x}(t) \oplus y(t)] \cdot z(t-1),$$

мы имеем рекуррентные соотношения:

$$z_i = \bar{x}_i \cdot y_i \vee [\bar{x}_i \oplus y_i] \cdot z_{i-1},$$

откуда и следует линейная оценка с коэффициентом:

$$C < L_\neg + 2L_\& + L_\vee + L_\oplus.$$

В. Обратимся теперь к схемам параллельного действия для умножения и деления. Осуществление этих операций над парой μ -значных двоичных чисел сводится, в соответствии с общеизвестными алгоритмами, к μ -кратной итерации операций с линейной оценкой сложности

(сложение, вычитание, сравнение...). Отсюда легко уже вытекает квадратичная оценка стоимости для умножения и деления. Точно так же можно установить кубическую оценку для операции x^y . Выкладки мы здесь полностью опускаем. В дальнейшем мы будем опираться на тот факт, что функция алгебры логики $f(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n)$, имеющая смысл: «число, имеющее двоичную запись $\langle x_1 \dots x_n \rangle$, делится нацело на число, имеющее двоичную запись $\langle y_1 \dots y_n \rangle$ », реализуема с квадратичной оценкой сложности.

3°. Пусть во множестве A^s всех наборов длины s из нулей и единиц выделено подмножество \tilde{A}^s наборов, которые мы будем называть существенными наборами. Пусть, далее, задан истинностный оператор $\tilde{\theta}$, для которого входным алфавитом является \tilde{A}^s , а входной алфавит есть подмножество \tilde{A}^n некоторого A^n . Естественно считать, что логическая сеть R реализует оператор $\tilde{\theta}$, если она реализует какой-нибудь истинностный оператор θ , определенный всюду на A^s и совпадающий с $\tilde{\theta}$ на \tilde{A}^s . Поскольку поведение θ на «несущественных» наборах этим определением не регламентируется, то желательно осуществить доопределение на несущественных наборах так, чтобы упростить, насколько это возможно, стоимость реализации. В связи с этой задачей мы рассмотрим в пунктах 3° и 4° некоторые классы функций алгебры логики и истинностных операторов, допускающих более простую реализацию, чем в общем случае.

Сначала рассмотрим классы функций алгебры логики, которые мы будем обозначать K_α , K_β , K_γ , $K_{\beta'}$, $K_{\gamma'}$. Каждый из них характеризуется условием, налагаемым на выбор некоторого множества \tilde{A} существенных наборов, и содержит все функции, произвольным образом определенные на \tilde{A} , но специальным образом определенные на множестве A/\tilde{A} несущественных наборов. Общая закономерность, которая устанавливается здесь, заключается в том, что вместо общей оценки для функций $f(x_1, \dots, x_s)$, имеющей вид

$$L(f) < \varrho \frac{2^s}{s} [1 + o(1)] = \varrho \frac{\text{число всех наборов из } A^s}{s} [1 + o(1)],$$

имеет место лучшая оценка:

$$L(f) < \varrho \frac{\text{число всех наборов из } \tilde{A}^s}{s} [1 + o(1)].$$

В дальнейшем $|x_1, \dots, x_m|$ обозначает число с двоичной записью $x_1 \dots x_m$. В классах K_α , K_β , K_γ на несущественных наборах функции равны нулю.

Класс K_α функций $f(x_1, \dots, x_s)$. A^s состоит из λ наборов ($2^{s-1} < \lambda \leq 2^s$), для которых $|x_1 \dots x_s| < \lambda$:

$$L(f) < \frac{\lambda}{s} [1 + o(1)] \quad \text{при } f \in K_\alpha, \quad s \rightarrow \infty.$$

Класс K_β функций $f(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}})$. $\tilde{A}^{m+\tilde{k}}$ состоит из $2^m \cdot \lambda$ наборов $x_1 \dots x_m q_1 \dots q_{\tilde{k}}$, для которых $|q_1 \dots q_{\tilde{k}}| < \lambda$ ($2^{\tilde{k}-1} < \lambda \leq 2^{\tilde{k}}$):

$$L(f) < \frac{2^m \lambda}{m + \tilde{k}} [1 + o(1)] \quad \text{при } f \in K_\beta, \quad m + \tilde{k} \rightarrow \infty.$$

Класс K_γ функций $f(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}})$. $\tilde{A}^{m+\tilde{k}}$ состоит из $(2^m - 1) \lambda$ наборов $x_1 \dots x_m q_1 \dots q_{\tilde{k}}$, для которых $|q_1 \dots q_{\tilde{k}}| < \lambda$ ($2^{\tilde{k}-1} < \lambda \leq 2^{\tilde{k}}$) и $|x_1 \dots x_m| \neq 0$.

$$L(f) < \frac{(2^m - 1) \lambda}{m + \tilde{k}} [1 + o(1)] \quad \text{при } f \in K_\gamma, \quad m + \tilde{k} \rightarrow \infty.$$

Классы $K_{\beta'}$, $K_{\gamma'}$ отличаются от K_β , K_γ лишь способом доопределения на несущественных наборах. Именно, выбираем набор $\tau_1 \dots \tau_{\tilde{k}}$ такой, что $|\tau_1 \dots \tau_{\tilde{k}}| < \lambda$, и полагаем $f(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}}) = f(x_1, \dots, x_m, \tau_1, \dots, \tau_{\tilde{k}})$ всякий раз, когда $|q_1 \dots q_{\tilde{k}}| \geq \lambda$ (можно сказать, что такие наборы $q_1 \dots q_{\tilde{k}}$ функционально неотличимы от $\tau_1 \dots \tau_{\tilde{k}}$). В предположении, что для K_β , K_γ метод синтеза и оценки уже установлены, покажем сразу же, что для $K_{\beta'}$ и $K_{\gamma'}$ синтез проходит с теми же оценками. Пусть, например, $f' \in K_{\beta'}$, а f отличается от f' лишь на несущественных наборах (причем $f \in K_\beta$). Пусть далее θ есть оператор, перерабатывающий набор $q_1 \dots q_{\tilde{k}}$ в $\tau_1 \dots \tau_{\tilde{k}}$ при $|q_1 \dots q_{\tilde{k}}| \geq \lambda$ и в тот же набор $q_1 \dots q_{\tilde{k}}$ в прочих случаях. Очевидно, $L(f') < L(f) + L(\theta)$; вместе с тем

слагаемое $L(\theta)$ мажорируется линейной функцией (см. схему сравнения), и им можно пренебречь.

Проведем теперь синтез и оценку для K_a ; перенесение этих результатов на K_β , K_γ не представляет труда и здесь не рассматривается.

Пусть $\lambda_v = \lambda' \cdot 2^v$ — наименьшее число, кратное 2^v , которое само не меньше λ . Тогда мы имеем*):

$$2^{s-1} \leq \lambda \leq \lambda_v = \lambda' \cdot 2^v \leq 2^s.$$

Положим $\mu = s - v$; при этом $\lambda' \leq 2^\mu$.

Функция f представима в виде:

$$\begin{aligned} f(x_1, \dots, x_\mu, x_{\mu+1}, \dots, x_{\mu+v}) &= \\ &= \bigvee_{|\sigma_1 \dots \sigma_\mu| < \lambda'} x_1^{\sigma_1} \dots x_\mu^{\sigma_\mu} f(\sigma_1, \dots, \sigma_\mu, x_{\mu+1}, \dots, x_{\mu+v}), \end{aligned}$$

ибо $f(\sigma_1, \dots, \sigma_\mu, x_{\mu+1}, \dots, x_{\mu+v}) = 0$ при $|\sigma_1 \dots \sigma_\mu| \geq \lambda'$. Исходя из этого представления, ясно, что

$$\begin{aligned} L(f) < \sum L[f(\sigma_1, \dots, \sigma_\mu, x_{\mu+1}, \dots, x_{\mu+v})] + \\ &+ L(D_\mu) + \lambda'(L_\& + L_v), \end{aligned}$$

где D_μ — универсальный дешифратор, зависящий от переменных x_1, \dots, x_μ , и следовательно (см. (7.12)) $L(D_\mu) = O(2^\mu)$ при $\mu \rightarrow \infty$.

Если положить $\mu = 3 \lg s$, $v = s - 3 \lg s$ и учесть, что при $s \rightarrow \infty$ $\lambda' \cdot 2^v = \lambda_v \sim \lambda$, то мы легко получаем:

$$L(f) < \left[\lambda' \frac{2^v}{s - 3 \lg s} + O(s^3) \right] [1 + o(1)] = \frac{\lambda}{s} [1 + o(1)].$$

4°. Пусть Q — множество всех $2^{\tilde{k}}$ наборов длины \tilde{k} из нулей и единиц и $Q' \subset Q$ — его подмножество, состоящее из λ наборов. Рассмотрим отображение T множества, содержащего $2^m \cdot \lambda$ наборов длины $m + \tilde{k}$ (существенных наборов) $x_1 \dots x_m q_1 \dots q_{\tilde{k}}$, для которых $(q_1 \dots q_{\tilde{k}}) \in Q'$, во множество наборов $(q'_1 \dots q'_{\tilde{k}}) \in Q'$.

Для произвольного истинностного оператора θ

$$q'_j(t) = \Psi_j(x_1(t), \dots, x_m(t), q_1(t), \dots, q_{\tilde{k}}(t))$$

*) Далее предполагается, что $v < s$.

(где $j = 1, 2, \dots, \tilde{k}$), совпадающего с T на существенных наборах, можно гарантировать лишь реализацию с оценкой (при $m + \tilde{k} \rightarrow \infty$):

$$L(\theta) < q \tilde{k} \frac{2^{m+\tilde{k}}}{m+\tilde{k}} [1 + o(1)]. \quad (*)$$

В связи с этим мы рассмотрим специальный класс операторов (класс K_δ), поведение которых на несущественных наборах характеризуется описываемым ниже условием.

Существует набор $\tau_1 \dots \tau_{\tilde{k}}$ во множестве Q' такой, что для всякого набора $q_1 \dots q_{\tilde{k}}$, не вошедшего в Q' , справедливо:

$$\Psi_j(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}}) = \Psi_j(x_1, \dots, x_m, \tau_1, \dots, \tau_{\tilde{k}}) \quad (**)$$

(что означает, что все такие наборы неотличимы от $\tau_1 \dots \tau_{\tilde{k}}$).

Укажем метод синтеза для операторов θ класса K_δ , гарантирующий оценку:

$$L(\theta) < q \lg \lambda \frac{2^{m\lambda}}{m + \lg \lambda} [1 + o(1)] \quad (7.28)$$

при $m + \tilde{k} \rightarrow \infty$, $2^{\tilde{k}} < \lg m$. (То, что (7.28) лучше, чем (*), можно проверить хотя бы для случая: $m \rightarrow \infty$, $\tilde{k} = 2$, $\lambda = 3$. В этом же случае (7.28) лучше и того результата, который можно было бы получить ранее указанным методом, если бы функции Ψ_j принадлежали классу $K_{\delta'}$.)

Выбор параметров. Разобьем группу переменных $x_1 \dots x_m$ на две подгруппы: x_1, \dots, x_π и $x_{\pi+1} \dots x_m$. При любой фиксации $x_{\pi+1} = \sigma_{\pi+1}, \dots, x_m = \sigma_m$ оператор θ индуцирует оператор $\theta_{\sigma_{\pi+1} \dots \sigma_m}$, отображающий наборы $x_1 \dots x_\pi q_1 \dots q_{\tilde{k}}$ в наборы $q'_1 \dots q'_{\tilde{k}}$ из Q' . Учитывая (**), легко видеть, что число различных операторов $\theta_{\sigma_{\pi+1} \dots \sigma_m}$ (обозначим его p) не более $\lambda^{\lambda \cdot 2^\pi}$, в то время как число способов фиксации набора $x_{\pi+1} \dots x_m$ равно $2^{m-\pi}$. В наших условиях (т. е. когда $\lambda < \lg m$, $m + \tilde{k} \rightarrow \infty$) имеет место $m \rightarrow \infty$, и если взять $2^\pi = \lg m$, то $\lambda^{\lambda \cdot 2^\pi}$ не больше $(\lg m)^{(\lg m)^2}$, и окажется значительно

меньше, чем $2^{m-\pi}$, которое равно $2^m/m$. Именно это обстоятельство и использовано при синтезе.

Метод синтеза. Пусть $\theta_1, \dots, \theta_v, \dots, \theta_p$ — какая-нибудь нумерация всех (попарно различных) операторов $\theta_{\sigma_{\pi+1} \dots \sigma_m}$. Сопоставим каждому оператору θ_v :

а) систему функций алгебры логики:

$$\Psi^v(x_1, \dots, x_\pi, q_1, \dots, q_{\tilde{k}}), \quad \text{где } j=1, 2, \dots, \tilde{k},$$

описывающую θ_v ;

б) функцию $V_v(x_{\pi+1}, \dots, x_m)$, равную 1 на тех и только тех наборах $\sigma_{\pi+1} \dots \sigma_m$, для которых $\theta_{\sigma_{\pi+1} \dots \sigma_m}$ совпадает с θ_v .

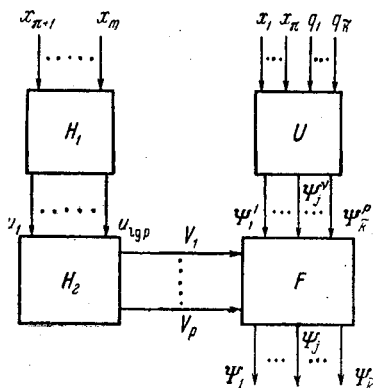


Рис. 7.4.

Тогда очевидно, что функции $\Psi_j (j=1, 2, \dots, \tilde{k})$, задающие оператор θ , допускают представление:

$$\begin{aligned} \Psi_j(x_1, \dots, x_\pi, x_{\pi+1}, \dots, x_m, q_1, \dots, q_{\tilde{k}}) = \\ = \bigvee_{v=1}^p V_v(x_{\pi+1}, \dots, x_m) \cdot \Psi_j^v(x_1, \dots, x_\pi, q_1, \dots, q_{\tilde{k}}). \end{aligned}$$

В соответствии с этим представлением логическая сеть строится из следующих блоков (см. рис. 7.4).

Блок H_1 . Этот блок реализует оператор, перерабатывающий любой из наборов $\sigma_{\pi+1} \dots \sigma_m$ в двоичную

запись $u_1 \dots u_{\lg p}$ номера v ($v \leq p$), отнесенного оператору $\theta_{\pi+1 \dots \sigma_m}$. При раздельной реализации $\lg p$ функций от $m - \pi$ переменных

$$L(H_1) < \lg p \cdot q^{\frac{m-\pi}{m-\pi}} \cdot [1 + o(1)].$$

Блок H_2 . Для реализации функций $V_v(x_{\pi+1}, \dots, x_m)$ остается еще построить универсальный дешифратор со входными переменными $u_1, \dots, u_{\lg p}$ и выходами $V_1, \dots, V_v, \dots, V_p$. Это и есть блок H_2 , для которого (см. (7.12)):

$$L(H_2) < q \cdot p \cdot [1 + o(1)].$$

Блок U . Этот блок реализует все $2^{2\pi+\tilde{k}}$ функций от переменных $x_1, \dots, x_\pi, q_1, \dots, q_{\tilde{k}}$ (в том числе и все функции Ψ_j^v); поэтому

$$L(U) < q \cdot 2^{2\pi+\tilde{k}} \cdot 2^{\pi+\tilde{k}}.$$

Блок F . Этот блок реализует функции Ψ_j , исходя из функций V_v , выработанных блоком H_2 , и функций Ψ_j^v , выработанных блоком U ; в состав F входят $k \cdot p$ $\&$ -ячеек и столько же \vee -ячеек. Итак:

$$L(F) = kp [L_{\&} + L_{\vee}].$$

Пусть теперь, как было указано выше:

$$m + \tilde{k} \rightarrow \infty, \quad \lambda < 2^{\tilde{k}} < \lg m, \quad 2^\pi = \lg m, \quad p \leq \lambda^{\lambda \cdot 2^\pi},$$

а значит: $(m - \pi) \sim m \sim (m + \tilde{k})$.

Тогда, как легко проверить:

$$L(H_1) < q\lambda \cdot 2^\pi \cdot \lg \lambda^{\frac{m-\pi}{m-\pi}} [1 + o(1)] = q \lg \lambda^{\frac{2^m \cdot \lambda}{m}} [1 + o(1)],$$

$$\text{в то время как } L(H_2) + L(U) + L(F) = o\left(\lg \lambda \cdot \frac{2^m \lambda}{m}\right).$$

Поскольку в наших условиях $m + \lg \lambda \sim m$, то этим и доказано (7.28).

Отметим еще, что как (7.28), так и оценки, установленные применительно к классам $K_\alpha, K_\beta, K_\gamma, K_{\beta'}, K_{\gamma'}$, не могут быть улучшены. Это следует из надлежащей

модификации теоремы о нижней оценке, которая — как доказательство самой этой теоремы — нами не излагается.

5°. Сравнительно простая схемная реализация арифметических операций может послужить основой для получения простых реализаций таких операторов, которые благодаря свойственным им особенностям могут быть описаны (хотя бы и частично) посредством арифметических операций и соотношений.

Проиллюстрируем это на примере *монотонных операторов*.

Определение. Истинностный оператор $\langle z_1, \dots, z_n \rangle = \theta [x_1, \dots, x_n]$ называется *монотонным*, если индуцируемая им числовая функция $z = \beta(x)^*$ является монотонной (не обязательно строго монотонной).

Метод синтеза, который учитывает монотонность оператора (для определенности мы будем рассматривать монотонно-возрастающий оператор), мы рассмотрим для случая, когда $m = n = k$.

Обозначим через $\beta^{-1}(v)$ наименьшее значение переменной x , для которого $\beta(x) \geq v$. Число v мы будем задавать в двоичной записи.

Таблица VII.2 задает монотонно-возрастающий оператор (для аргумента x задана одновременно и десятичная запись). В данном случае $\beta^{-1}(1000) = 7$, ибо, начиная с $x = 7$, $\beta(x)$ становится больше или равной (1000); $\beta^{-1}(1100) = 10$.

Для компоненты z_k (старшего разряда числа z) имеет место очевидное соотношение $z_k = [x \geq \beta^{-1}(10 \dots 0)]$.

Аналогично, для компонент z_{k-1}, z_{k-2}, \dots получается:

$$\left. \begin{aligned} z_{k-1} &= [\beta^{-1}(010 \dots 0) \leq x < \beta^{-1}(100 \dots 0) \vee [\beta^{-1}(110 \dots 0) \leq x], \\ z_{k-2} &= [\beta^{-1}(0010 \dots 0) \leq x < \beta^{-1}(010 \dots 0)] \vee \\ &\quad \vee [\beta^{-1}(0110 \dots 0) \leq x < \beta^{-1}(100 \dots 00)] \vee \\ &\quad \vee [\beta^{-1}(1010 \dots 0) \leq x < \beta^{-1}(110 \dots 0)] \vee \\ &\quad \vee [\beta^{-1}(1110 \dots 0) \leq x]. \end{aligned} \right\} \quad (7.29)$$

*) При интерпретации $\langle z_1, \dots, z_n \rangle$ и $\langle x_1, \dots, x_n \rangle$ как двоичных записей чисел z, x .

Таблица VII.2

x	x_4	x_3	x_2	x_1	z_4	z_3	z_2	z_1
0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	1
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	0
9	1	0	0	1	1	0	0	1
10	1	0	1	0	1	1	0	0
11	1	0	1	1	1	1	0	0
12	1	1	0	0	1	1	0	0
13	1	1	0	1	1	1	0	0
14	1	1	1	0	1	1	1	1
15	1	1	1	1	1	1	1	1

Вернувшись к нашему примеру, отметим, что для оператора, заданного таблицей VII.2, мы будем иметь:

$$z_4 = [7 \leq x], \quad z_3 = [3 \leq x < 7] \vee [10 \leq x].$$

Пусть мы выразили таким образом j старших разрядов числа z . В правых частях соответствующих определяющих равенств будут фигурировать:

а) функции вида $x \geq \beta^{-1}(v)$ и их отрицания вида $x < \beta^{-1}(v)$; общее число различных функций $x \geq \beta^{-1}(v)$ не превосходит, как легко видеть, числа

$$1 + 2 + 2^2 + \dots + 2^{j-1} = 2^j - 1 < 2^j;$$

б) дизъюнкции, общее число которых $\leq 1 + 2 + \dots + 2^{j-1} < 2^j$;

в) конъюнкции, общее число которых также $< 2^j$.

Исходя из этого, оценка совместной реализации j старших компонент, в соответствии с формулой (7.29), будет иметь (если вспомнить, что сравнения $x > \beta^{-1}(v)$ реализуются с линейной оценкой $\leq C \cdot k$) следующий вид:

$$L'_j < 2^j (C \cdot k + L_{\&} + L_{\vee} + L_{\neg}).$$

Для остальных $m - j$ разрядов можно взять отдельную реализацию по Лупанову с оценкой (при $k - j \rightarrow \infty$)

$$L''_{k-j} < q \cdot (k - j) \frac{2^k}{k} [1 + o(1)].$$

Если положить $j = k - \lg k$, то для стоимости $L'_j + L''_{k-j} = L(\theta)$ реализации оператора θ получается оценка:

$$L(\theta) < 2q \frac{2^k \lg k}{k} [1 + o(1)], \quad (7.30)$$

в то время как для операторов $\theta \in \theta_{k, k, o}$ можно гарантировать вообще лишь то, что:

$$L(\theta) \sim q \frac{k 2^k}{k} = q 2^k \text{ (см. (7.25'')).}$$

П р и м е ч а н и е. При получении оценки стоимости монотонного оператора свойство монотонности не было использовано до конца. Это видно хотя бы из того, что найденные применительно к оператору, заданному таблицей VII.2, выражения для z_4 и z_3 останутся справедливыми и в том случае, если, не меняя столбца x_1, x_2, x_3, x_4 , в столбце для z_1, z_2, z_3, z_4 мы произвольным образом переставим строки между двумя соседними горизонтальными чертами (тем самым монотонность будет нарушена). Класс операторов, допускающих представление вида (7.29) для j старших разрядов, шире класса монотонных операторов, и для этого класса операторов оценка (7.30) не может быть улучшена. Что касается монотонных операторов, то, по-видимому, возможен синтез с лучшей оценкой*).

*) Недавно О. Б. Лупанов получил лучшую оценку: $L(\theta) < 2q \frac{2^k}{k} [1 + o(1)]$; см. О. Б. Лупанов, О принципе локального кодирования и реализации функций из некоторых классов схемами из функциональных элементов, ДАН СССР, т. 140, № 2, 1961.

§ 4. Выбор кода

1°. Оценки, установленные в предыдущих параграфах, относятся к тому случаю, когда входной и выходной алфавиты закодированы в двоичном коде. Представляется естественным, что в тех случаях, когда выбор кодовых групп для входных и выходных букв не предрешен условиями задачи, можно надеяться на упрощение схемы благодаря удачному выбору кодовых групп. На практике это обстоятельство всегда учитывается. Так, например, в арифметических устройствах, работающих на десятичной системе счисления, надлежащий выбор четырехзначных двоичных кодовых групп (так называемых *тетрад*) для цифр 0, 1, 2, ..., 9 (играющих одновременно роль входных и выходных букв) позволяет достичь определенных конструктивных упрощений. В этом параграфе мы рассмотрим подробнее ситуацию, возникающую при обобщении задачи о выборе тетрад, и выясним возможности упрощения схем за счет надлежащего кодирования.

Пусть истинностный оператор θ имеет один алфавит A , представляющий собой множество из K букв; пусть этот алфавит является одновременно входным и выходным. Оператор θ задает однозначное отображение множества букв оператора A в себя (то есть в часть множества A) или на себя (то есть на все буквы множества A). Если $2^{h-1} < K \leq 2^h$, то эти буквы могут быть закодированы k -значными кодовыми группами, причем, каков бы ни был способ кодирования, отдельная реализация по Лупанову гарантирует (асимптотически) оценку стоимости:

$$L(\theta) < q \cdot 2^h [1 + o(1)]. \quad (7.31)$$

Возникает вопрос: можно ли существенно улучшить оценку путем такого выбора кодовых групп, который учитывает специфику данного оператора?

Утвердительный ответ на этот вопрос вытекает из тех методов кодирования, которые будут описаны ниже; мы ограничимся далее рассмотрением того случая, когда K есть степень двойки: $K = 2^h$.

2°. Зададим истинностный оператор (отображение) θ с помощью ориентированного графа следующим образом.

Пусть буквы алфавита A изображаются точками (вершинами) графа, а стрелки, которые их соединяют, указывают для каждой точки, в какую точку она отображается.

Примеры. Отображениям, заданным таблицами VII.3 и VII.4, соответствуют ориентированные графы, изображенные соответственно на рис. 7.5, а и б.

Таблица VII.3

Вход	Выход	Вход	Выход
a	c	i	i
b	a	j	i
c	b	k	p
d	e	l	k
e	f	m	l
f	g	n	m
g	h	o	n
h	d	p	o

Таблица VII.4

Вход	Выход	Вход	Выход
a	k	i	e
b	a	j	m
c	b	k	b
d	g	l	l
e	d	m	o
f	g	n	m
g	k	o	p
h	l	p	o

Первое отображение является взаимно однозначным отображением множества A на себя, и поэтому его граф распадается на отдельные циклы; в данном случае мы имеем по одному циклу длины 6, 5, 3 и два одноэлементных цикла (длины 1).

Второе отображение не является взаимно однозначным, и поэтому его граф состоит из циклов, к некоторым

из вершин которых примыкают корни ориентированных деревьев (ориентация ребер в дереве — к корню); так, например, к циклу длины 3 примыкают два дерева (именно, в вершинах b и k).

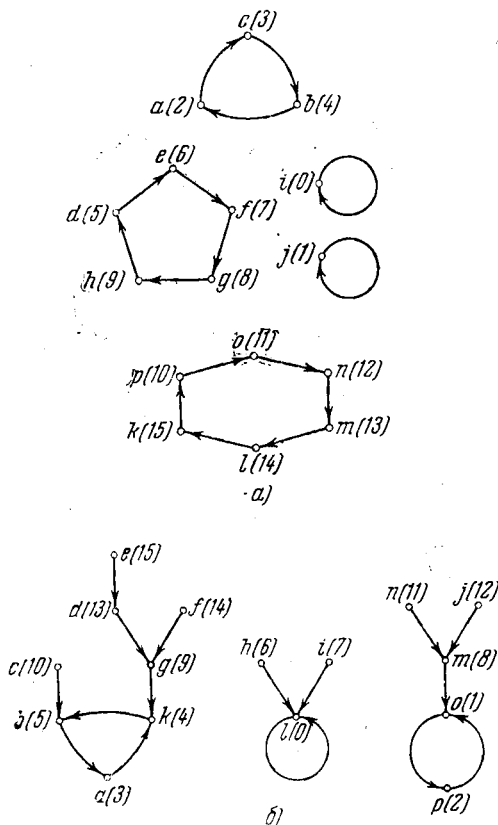


Рис. 7.5.

Припишем вершинам полученных ориентированных графов ранги следующим образом:

1) вершинам, лежащим на циклах, припишем нулевой ранг;

2) всякой вершине, из которой выходит стрелка, ведущая к вершине ранга i (из каждой вершины выходит только одна стрелка), приписывается ранг $i+1$.

Иными словами, ранг вершины указывает длину ориентированного пути, ведущего от данной вершины до вершины цикла. Например, на рис. 7.5, б вершина e имеет наибольший ранг, равный 3. Ранги вершин f, i, p равны соответственно 2, 1, 0.

Очевидно, что кодирование алфавита A равносильно установлению нумерации вершин соответствующего графа; при этом двоичные записи этих номеров и являются кодами букв. Мы опишем ниже такую нумерацию вершин, при которой индуцируемое отображение номеров будет описываться посредством арифметических операций и монотонного оператора; тем самым, в соответствии с результатами предыдущего параграфа, будет достигнуто определенное упрощение схемной реализации.

3°. Н у м е р а ц и я в е р ш и н. Пусть в графе имеется K_i вершин i -го ранга (где $i=0, 1, \dots, S$). Отнесем вершинам нулевого ранга номера $0, 1, 2, \dots, K_0-1$; вершинам первого ранга отнесем номера $K_0, K_0+1, \dots, K_0+K_1-1$, и т. д.

Распределение номеров между вершинами данного фиксированного ранга производится следующим образом.

1) Прежде всего рассматриваются вершины нулевого ранга (то есть вершины, лежащие на циклах). Циклы располагаются в порядке возрастания их длин. В первом из них выбирается вершина (любая), которой приписывается номер 0; тогда стрелка указывает вершину, к которой нужно отнести номер 1 и т. д., до тех пор, пока ко всем вершинам цикла не будут отнесены номера. Потом выбирается вершина в следующем цикле и ей приписывается наименьший из не занятых еще номеров и т. д.

2) Пусть вершинам ранга i и меньших чем i рангов уже отнесены номера $0, 1, 2, \dots, K_0+K_1+\dots+K_i-1$. Тогда K_{i+1} вершинам ранга $i+1$ приписываются номера $K_0+K_1+\dots+K_i, K_0+K_1+\dots+K_i+1, \dots, K_0+K_1+\dots+K_i+K_{i+1}-1$; при этом соблюдается следующее требование: если стрелки, выходящие из вершин a, b (ранга $i+1$), ведут в вершины $a'b'$ (ранга i) такие, что номер a' меньше номера b' , то и номер a меньше номера b .

На рис. 7.5 в скобках выписаны номера вершин, установленные в соответствии с указанным правилом.

4°. Отметим теперь свойства оператора (отображения) θ , вытекающие из характера принятой нумерации.

а) Оператор θ осуществляет монотонно-возрастающее отображение номеров $K_0, K_0+1, \dots, K-1$, отнесенных вершинам не нулевого ранга, во множество всех номеров.

Обозначим через θ_1 отображение, совпадающее с θ на множестве номеров $\geq K_0$ и переводящее все номера, меньшие чем K_0 , в нуль. Очевидно, что θ_1 является монотонным отображением, определенным на множестве всех номеров $0, 1, \dots, K-1$.

б) Предположим, что среди всех циклов рассматриваемого отображения имеются в точности ν циклов, длины которых попарно различны; эти длины обозначим через l_1, l_2, \dots, l_ν . Тогда существует возрастающая последовательность чисел:

$$0 = \sigma_0, \sigma_1, \dots, \sigma_{\nu-1}, \sigma_\nu = K_0$$

таких, что номера x вершин всех циклов длины l_i и только они удовлетворяют неравенству $\sigma_{i-1} \leq x < \sigma_i$.

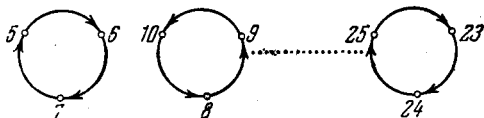


Рис. 7.6.

Исходя из этого, взаимно однозначное отображение числового отрезка $0, 1, 2, \dots, K_0-1$ на себя, осуществляемое оператором θ , может быть охарактеризовано следующим образом.

Пусть $\sigma_{i-1} \leq x < \sigma_i$. Тогда: 1) если x есть одно из чисел $\sigma_{i-1} + l_i - 1, \sigma_{i-1} + 2l_i - 1, \dots, \sigma_i - 1$, то есть если $(x - \sigma_{i-1} + 1) : l_i^*$, то $T(x) = x - (l_i - 1)$; 2) в противном случае $T(x) = x + 1$. На рис. 7.6 эта ситуация проиллюстрирована для $\sigma_{i-1} = 5, l_i = 3, \sigma_i = 26$, где $\theta(10) = 8, \theta(24) = 25$ и т. п.

*) $a : b$ означает, что a делится нацело на b .

Указанные свойства отображения θ подсказывают следующий метод синтеза.

5°. Синтез и оценка. Схема, реализующая оператор $z = \theta(x)$, состоит из описываемых ниже крупных блоков А, В, С (см. рис. 7.7).

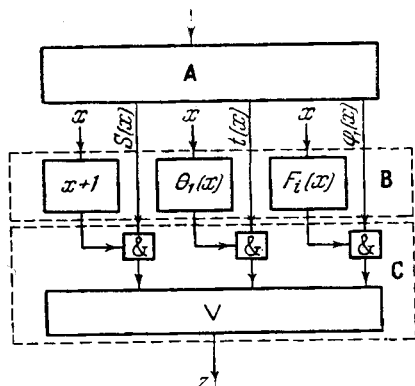


Рис. 7.7.

Б л о к А. Для любого набора $x_1 x_2 \dots x_k$, являющегося кодом (номером) одной из $K = 2^k$ букв алфавита, применение оператора θ сводится к применению одной из следующих операций:

- 1) вычитания числа $l_i - 1$ ($i = 1, 2, \dots, v$);
- 2) добавления единицы;
- 3) применения монотонного оператора θ_1 .

Какую именно из этих операций следует применять — это «выясняет» блок А; это «выяснение» осуществляется посредством реализации функций алгебры логики:

$$\varphi_i(x_1, \dots, x_k) \quad (\text{где } i = 1, 2, \dots, v),$$

$$S(x_1, \dots, x_k),$$

$$t(x_1, \dots, x_k).$$

(обозначаемых нами сокращенно через $\varphi_i(x)$, $S(x)$, $t(x)$); при этом:

$$\varphi_i(x) = [\sigma_{i-1} \leq x < \sigma_i] \& [(x - \sigma_{i-1} + 1) : l_i],$$

$$S(x) = [x < \sigma_v \& \bigwedge_{i \leq v} \overline{\varphi_i(x)}],$$

$$t(x) = [\sigma_v \leq x].$$

Для этого реализуются сначала v сравнений $\sigma_i < x$ и v проверок делимости на l_i , а потом посредством $4v$ $\&$ -ячеек и $2v$ \neg -ячеек (в том числе v \neg -ячеек для образования сравнений $x \geq \sigma_i$ в виде $\sigma_i > x$) образуются уже функции φ_i , t , S . Учитывая линейную оценку для стоимости сравнения и квадратичную — для стоимости проверки делимости, мы получаем:

$$L(A) < v \cdot [C_1 k + C_2 k^2 + 4L_{\&} + 2L_{\neg}].$$

Б л о к В. Этот блок реализует операторы: $F_i(x) = [x - l_i + 1]$, где $i = 1, 2, \dots, v$; $\theta_1(x)$ и $x + 1$. Учитывая линейную оценку для вычитания и сложения, а также оценку для монотонного оператора θ_1 , мы имеем (при $k \rightarrow \infty$) следующую оценку:

$$L(B) < v \cdot C_3 k + C_1 k + 2q \frac{2^k \lg k}{k} [1 + o(1)].$$

Б л о к С. Этот блок реализует оператор θ , исходя из функций алгебры логики и операторов, реализуемых в блоках А и В, и состоит из $v + 2$ k -членных конъюнкций и столько же k -членных дизъюнкций.

Следовательно:

$$L(C) = (v + 2) \cdot k [L_{\&} + L_{\vee}].$$

Очевидно, что при $k \rightarrow \infty$, оценивая величину $L(\theta)$, равную $L(A) + L(B) + L(C)$, можно пренебречь всеми слагаемыми, за исключением слагаемых $C_2 v k^2$ и $2q \frac{2^k \lg k}{k}$.

Исследуем первое из них. Очевидно, наибольшее возможное значение числа v циклов различной длины получится, если считать, что имеется по одному циклу длины 1, по одному циклу длины 2 и т. д.; в то же время сумма длин циклов не превосходит 2^k (числа всех букв в алфавите). Следовательно,

$$1 + 2 + 3 + \dots + v = \frac{(1+v)v}{2} \leq 2^k,$$

откуда получается, что

$$v < 2^{\frac{k+1}{2}}.$$

Окончательно получаем:

$$L(\theta) < C_2 k^2 2^{\frac{k+1}{2}} + 2q \frac{2^k \lg k}{k} [1 + o(1)].$$

Отсюда видно, что если отображение θ является взаимно однозначным (все вершины — нулевого ранга), то

$$L(\theta) < C k^2 \cdot 2^{\frac{k+1}{2}}. \quad (7.32)$$

В общем же случае мы имеем:

$$L(\theta) < 2q \frac{2^k \lg k}{k} [1 + o(1)]. \quad (7.33)$$

Эти оценки на самом деле лучше, чем оценка (7.31), которая может быть гарантирована при произвольном способе кодирования.

§ 5. Синтез сетей с памятью

В этом параграфе мы опишем метод синтеза для операторов θ класса $\theta_{m, n, k}$, проходящий с верхней оценкой $q \frac{\lg N_{m, n, k}}{\lg \lg N_{m, n, k}}$, приведенной в § 1. Начнем с одного замечания, касающегося способа кодирования букв в алфавитах рассматриваемых операторов. Коды входных и выходных букв синтезируемой сети обычно бывают заданными заранее: первые приспособлены к источнику, из которого поступает в сеть информация, а вторые — к приемнику информации, выдаваемой сетью. Что же касается кодов состояний, то они предназначены для изображения информации, которая циркулирует в самой системе, и их выбор может быть, в частности, подчинен соображениям, касающимся упрощения сети.

Предположим, что входные и выходные буквы некоторого оператора закодированы соответственно последовательностями (x_1, x_2, \dots, x_m) и (z_1, z_2, \dots, z_n) . Предположим, далее, что буквы состояний закодированы каким-то образом (пока произвольным) с помощью наборов $q_1 q_2 \dots q_{\tilde{k}}$, где \tilde{k} — наименьшее натуральное число, такое, что вес оператора не больше $2^{\tilde{k}}$. Канонические уравнения оператора

имеют вид:

$$\left. \begin{aligned} z_i(t) &= \Phi_i(x_1(t), \dots, x_m(t), q_1(t), \dots, q_{\tilde{k}}(t)), \\ q_j(t+1) &= \Psi_j(x_1(t), \dots, x_m(t), q_1(t), \dots, q_{\tilde{k}}(t)), \end{aligned} \right\} \quad (7.34)$$

где $i = 1, 2, \dots, n$, $j = 1, 2, \dots, \tilde{k}$.

Оценим стоимость L сети, полученной путем раздельной реализации методом Лупанова указанных $n + \tilde{k}$ функций от $m + \tilde{k}$ переменных каждая, с последующим образованием обратных связей через \tilde{k} задерживающих элементов (через L_3 обозначим стоимость одного элемента задержки). Очевидно, при $m + \tilde{k} \rightarrow \infty$, пренебрегая величиной $\tilde{k}L_3$, мы имеем:

$$L(\theta) < \varrho \frac{(n + \tilde{k}) 2^{m + \tilde{k}}}{m + \tilde{k}} \cdot (1 + o(1)). \quad (7.35)$$

Допустим теперь, что выполнены условия:

1) вес оператора является степенью двойки и, следовательно, в точности равен $2^{\tilde{k}}$;

2) $m \rightarrow \infty$ (условие более сильное, чем $m + k \rightarrow \infty$).

При этих условиях $k = \tilde{k}$, $(2^m - 1) \sim 2^m$ и поэтому сопоставление с теоремой о нижней оценке (см. (7.9) и (7.10)) показывает, что оценка (7.35) не может быть улучшена, а следовательно, имеет место асимптотика:

$$L(m, n, k) \sim \varrho \frac{\lg N(m, n, k)}{\lg \lg N(m, n, k)}.$$

В этом случае раздельная реализация функций Φ_i, Ψ_j в представлении (7.34) доставляет почти для всех операторов реализующие их сети с асимптотически минимально возможной стоимостью. Однако если нарушено хотя бы одно из условий 1) или 2), то получается пробел между нижней и верхней оценками. Это наводит на мысль, что метод синтеза может быть улучшен за счет привлечения каких-нибудь дополнительных принципов, относящихся к способу кодирования алфавита состояний. К изложению такого метода синтеза мы сейчас и перейдем.

Рассмотрим оператор θ , вес которого λ , быть может, и не является степенью двойки: $2^{\tilde{k}-1} < \lambda \leq 2^{\tilde{k}}$, где

\tilde{k} — целое число. Пусть по-прежнему входные и выходные буквы закодированы соответственно наборами $x_1 \dots x_m$ и $z_1 \dots z_n$, а алфавит состояний Q' содержит буквы:

$$q_0, q_1, \dots, q_{\lambda-1}.$$

Канонические уравнения оператора пусть имеют вид:

$$\begin{aligned} z_i(t) &= \Phi_i[x_1(t), \dots, x_m(t), q(t)], \\ q(t+1) &= \Psi[x_1(t), \dots, x_m(t), q(t)], \\ q(t) &\in Q'. \end{aligned}$$

Расширим алфавит Q' до алфавита Q , насчитывающего $2^{\tilde{k}}$ букв, добавлением μ букв (где $\mu = 2^{\tilde{k}} - \lambda$) $q_\lambda, q_{\lambda+1}, \dots, q_{2^{\tilde{k}}-1}$, задающих состояния, не отличимые от состояния q_τ , определяемого следующим образом.

Рассматривается отображение $\Psi(0, \dots, 0, q)$ множества Q' в себя и нумерация вершин соответствующего ориентированного графа (см. § 4). В качестве q_τ берется состояние, изображенное вершиной с максимальным номером. Неотличимость присоединенных состояний $q_\lambda, \dots, q_{2^{\tilde{k}}-1}$ мы обеспечим путем следующего доопределения функций Φ_i и Ψ :

$$\left. \begin{aligned} \Phi_i(x_1, \dots, x_m, q_j) &= \Phi_i(x_1, \dots, x_m, q_\tau), \\ \Psi(x_1, \dots, x_m, q_j) &= \Psi(x_1, \dots, x_m, q_\tau), \end{aligned} \right\} \quad (7.36)$$

где $j = \lambda, \lambda + 1, \dots, 2^{\tilde{k}} - 1$.

В соответствии с процедурой, описанной в § 4, напишем теперь $2^{\tilde{k}}$ буквам из Q такие коды длины \tilde{k} , при которых:

(I) отображение $\Psi(0, \dots, 0, q)$ множества Q в себя реализуемо логической сетью со стоимостью $O\left(\frac{2^{\tilde{k}} \cdot \lg \tilde{k}}{\tilde{k}}\right)$;

(II) кодом (номером) каждой буквы $q_j (j \geq \lambda)$ является двоичная запись числа j . (Условно (II) обеспечивается тем, что буквы $q_\lambda, q_{\lambda+1}, \dots$, так же как и q_τ , изображены в ориентированном графе вершинами максимального ранга.)

Покажем, что при таком способе кодирования состояний оператор θ , имеющий теперь уже канонические уравнения вида

$$\left. \begin{aligned} z_i(t) &= \Phi_i(x_1(t), \dots, x_m(t), q_1(t), \dots, q_{\tilde{k}}(t)), \\ q_j(t+1) &= \Psi_j(x_1(t), \dots, x_m(t), q_1(t), \dots, q_{\tilde{k}}(t)), \end{aligned} \right\} \quad (7.37)$$

реализуем со стоимостью:

$$L(\theta) < \left[nq \frac{2^m \cdot \lambda}{m+k} + kq \frac{(2^m-1) \cdot \lambda}{m+k} \right] \cdot (1 + o(1)). \quad (7.38)$$

Логическая сеть, реализующая оператор θ , состоит из блока, реализующего систему функций $\{\Phi_i\}_{i \leq n}$, блока, реализующего систему функций $\{\Psi_j\}_{j \leq \tilde{k}}$, и из \tilde{k} элементов задержки; следовательно:

$$L(\theta) < L\{\Phi_i\}_{i \leq n} + L\{\Psi_j\}_{j \leq \tilde{k}} + \tilde{k}L_3. \quad (7.38')$$

Рассмотрим в отдельности каждую из составных частей:

А. Для каждой из функций Φ_i ($i = 1, 2, \dots, n$), в соответствии с (7.36) и с принятым способом кодирования, при фиксированном наборе $(\tau_1 \dots \tau_{\tilde{k}})$ справедливо:

$$\Phi_i(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}}) = \Phi_i(x_1, \dots, x_m, \tau_1, \dots, \tau_{\tilde{k}}),$$

если $|q_1 \dots q_{\tilde{k}}| < \lambda$.

Таким образом, функции Φ_i принадлежат классу $K_{\beta'}$. Следовательно (см. стр. 374):

$$\begin{aligned} L\{\Phi_i\}_{i \leq n} &\leq \sum_{i \leq n} L(\Phi_i) \leq n \cdot q \frac{2^m \cdot \lambda}{m+k} [1 + o(1)] \sim \\ &\sim nq \frac{2^m \cdot \lambda}{m+k} [1 + o(1)], \end{aligned} \quad (7.39)$$

ибо $m + \tilde{k} \sim m + k$.

В. При реализации системы $\{\Psi_j\}_{j \leq \tilde{k}}$ рассмотрим две конструкции, зависящие от того, имеет ли место (при $m+k \rightarrow \infty$) $2^{\tilde{k}} \geq \lg m$ или $2^{\tilde{k}} < \lg m$.

В₁. Пусть $2^{\tilde{k}} < \lg m$, а значит при $m + \tilde{k} \rightarrow \infty$ также и $m \rightarrow \infty$. Из принятого способа кодирования вытекает, что система функций $\{\Psi_j\}_{j \leq \tilde{k}}$ задает оператор типа K_0

(см. пункт 4° в § 3). Следовательно:

$$L\{\Psi_j\}_{j \leq \tilde{k}} < q \cdot \lg \lambda \cdot \frac{2^m \cdot \lambda}{m + \lg \lambda} \cdot [1 + o(1)]. \quad (7.40)$$

Поскольку при $m \rightarrow \infty$ ($2^m - 1$) $\sim 2^m$, то из (7.39) и (7.40) уже получается (7.38), — если учесть, что $m + k = (m + \lg \lambda) \sim (m + \tilde{k})$ и что в (7.38') слагаемым $\tilde{k}L_\lambda$ можно пренебречь.

В₂. Рассмотрим, наконец, случай, когда $2^{\tilde{k}} \geq \lg m$, а следовательно, при $m + \tilde{k} \rightarrow \infty$ имеют место $2^{\tilde{k}} \rightarrow \infty$ и $\lambda \rightarrow \infty$.

Пусть $\Psi'_j(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}})$ равна 0 при $x_1 = x_2 = \dots = x_m = 0$ и совпадает с $\Psi_j(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}})$ в прочих случаях. Тогда при $j = 1, 2, \dots, \tilde{k}$ мы имеем:

$$\Psi_j(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}}) = \bar{x}_1 \bar{x}_2 \dots \bar{x}_m \Psi_j(0, \dots, \dots, 0, q_1, \dots, q_{\tilde{k}}) \vee \Psi'_j(x_1, \dots, x_m, q_1, \dots, q_{\tilde{k}}).$$

Из этого представления непосредственно следует:

$$L\{\Psi_j\}_{j \leq \tilde{k}} \leq L\{\Psi'_j\}_{j \leq \tilde{k}} + L\{\Psi_j(0, \dots, 0, q_1, \dots, q_{\tilde{k}})\}_{j \leq \tilde{k}} + mL_{\neg} + k(L_{\&} + L_{\vee}).$$

Последними двумя слагаемыми можно пренебречь.

С другой стороны, в соответствии с принятым способом кодирования состояний и определением функций Ψ'_j , эти функции принадлежат классу K_V (см. стр. 374). Следовательно:

$$L\{\Psi'_j\}_{j \leq \tilde{k}} \leq \sum_{j \leq \tilde{k}} L(\Psi'_j) \leq \tilde{k}q \cdot \frac{(2^m - 1)\lambda}{m + \tilde{k}} \cdot [1 + o(1)].$$

Поскольку $\lambda \rightarrow \infty$, то $\tilde{k} \sim k$ (в частности, $\tilde{k} \rightarrow \infty$). Но в то же время в соответствии с (I) (стр. 391) при $\tilde{k} \rightarrow \infty$ мы имеем:

$$L\{\Psi_j(0, \dots, 0, q_1, \dots, q_{\tilde{k}})\}_{j \leq \tilde{k}} \leq O\left(\frac{2^{\tilde{k}} \cdot \lg \tilde{k}}{\tilde{k}}\right).$$

Поэтому это слагаемое есть $o\left(\tilde{k}q \frac{(2^m-1)\lambda}{m+\tilde{k}}\right)$, и им можно пренебречь. Итак:

$$L\{\Psi_j\}_{j \leq \tilde{k}} < \tilde{k} \cdot q \cdot \frac{(2^m-1)\lambda}{m+\tilde{k}} \cdot [1 + o(1)]. \quad (7.41)$$

Как уже указывалось, $k \sim \tilde{k}$. Поэтому из (7.39) и (7.41) опять вытекает (7.38).

Сделаем еще одно замечание.

Пусть $\theta' \supset \theta$ означает: для всякого состояния оператора θ существует неотличимое от него состояние оператора θ' (обратное утверждение, вообще говоря, не обязательно). Логическая сеть, реализующая θ' , в каком-то смысле реализует и θ . Однако при таком ослабленном условии реализуемости, вообще говоря, улучшение верхней оценки не получается.

ДОБАВЛЕНИЕ

Как мы указывали во введении, настоящая книга не охватывает всей проблематики теории конечных автоматов. Вместе с тем не оказалось возможным изложить и ряд исследований (в основном выполненных или опубликованных в самое последнее время), которые непосредственно примыкают к рассматриваемым в ней вопросам. Ниже будут пояснены в самых общих чертах тематика некоторых исследований такого рода и связь их с содержанием данной книги; будет также указана соответствующая литература.

Обратимся сначала к наиболее прикладному и вместе с тем наиболее раннему направлению в теории конечных автоматов — к так называемой структурной теории автоматов, предметом которой является изучение способов построения сложных автоматов из элементарных автоматов (см. §§ 4—6 гл. II, § 5 гл. IV и гл. VI—VII данной книги). Результаты, устанавливаемые в этой теории для схем одного типа, в значительной мере удается переносить и на схемы другого типа. Однако до сих пор еще не создана общая теория схем и исследования проводятся применительно к схемам конкретного типа*). Ряд общих результатов, относящихся к схемам из произвольных элементов, содержится в работах О. Б. Лупанова^[26] **). А. Н. Колмогоров разработал круг понятий, удобных для классификации и изучения схем автоматов, исходя из того, изменяется ли с течением времени состав элементов в схеме и способ их соединения или нет.

*) В данной книге — в применении к логическим сетям.

**) См. литературу в конце книги.

По инициативе А. Н. Колмогорова его учениками предпринято такое исследование синтеза автоматов, которое основано на совместном применении принципов, относящихся к асимптотической оценке сложности схем (см. гл. VII) и к оценке энтропии функциональных пространств (такой подход позволяет, например, исследовать наилучшие способы вычисления автоматами значений аналитических функций).

О. Б. Лупановым [28] разработан принцип локального кодирования, позволяющий выделять многие классы операторов, которые допускают простую схемную реализацию (ср. § 3 гл. VII).

Установлению критериев полноты для систем элементарных автоматов посвящены работы В. Б. Кудрявцева [20]; другой подход изложен в статье А. А. Летичевского [24].

Сравнительно более новым направлением в теории конечных автоматов является так называемая абстрактная теория автоматов, в которой изучается поведение автоматов (т. е. характер осуществляемого ими преобразования информации) при отвлечении от специфики их внутреннего устройства (ср. §§ 1—3 гл. II, §§ 1—4 и п. 5, § 5 из гл. IV, гл. V). В рамках абстрактной теории автоматов содержание понятий «автомат» и «машина» исчерпывается, по существу, стандартным описанием преобразования информации, которое осуществляется автоматом. В нашем изложении этой концепции соответствуют понятия ограниченно-детерминированного оператора *) (заданного своими каноническими уравнениями) или элементарного автомата. Однако большинство авторов склоняется, по-видимому, к тому, чтобы употреблять в этом смысле термин «конечный автомат» [13,2]; другими синонимами являются: «последовательностная машина», «автомат Мили», «автомат Мура» **). Типичными задачами абстрактной теории автоматов являются задачи с экспериментами над автоматами (ср. § 3 гл. IV), исследование которых продолжено в работах Гинзбурга [67], А. А. Карацубы [16],

*) Ограниченно-детерминированный оператор называется иногда автоматным оператором.

**) Эти термины употребляются (причем неодинаково всеми авторами) для указания тех или иных особенностей канонических уравнений.

В. В. Мартынюка^[29] и др. Преобразование периодических последовательностей (ср. § 4 гл. IV) и обратимость ограниченно-детерминированных операторов изучены Гинзбургом^[68]. В ряде работ рассмотрены приемы минимизации алфавита состояний (ср. § 5 гл. IV).

Проблематика синтеза в абстрактной теории автоматов (ср. гл. V) предполагает предварительное создание удобного языка для записи условий работ автомата с удобными алгоритмами перехода от записи к каноническим уравнениям. По-видимому, в настоящее время наиболее удобными являются алгебраические языки (язык регулярных выражений С. К. Клини^[18] и его усовершенствования в работах В. М. Глушкова*), Копи, Эльгота и Райта^[64]) и логические языки, основанные на логике однометных предикатов (ср. гл. V). На возможность применения последних было впервые указано Б. А. Трахтенбротом^[42] и А. Черчем^[63]. В последнее время появился ряд статей, посвященных исследованию логических языков (Р. Бюхи^[58], Б. А. Трахтенброт^[46], К. Эльгот^[65]), в которых обобщаются и усиливаются результаты § 4 гл. V. При этом ставится следующая задача: создать алгоритм, который по любой формуле данного языка 1) выясняет, существует ли ограниченно-детерминированный оператор, удовлетворяющий этой формуле; 2) если да, то строит канонические уравнения какого-либо одного такого оператора или всех таких операторов.

Ряд алгоритмических задач, связанных с распознаванием тех или иных свойств операторов, исходя из заданных регуляторных выражений, содержится в статье М. Рабина и Д. Скотта^[75].

В работах Ю. В. Глебского, В. И. Левенштейна, Д. Гилберта и Э. Ф. Мура исследуются методы кодирования и декодирования сообщений, осуществимые в конечных автоматах ^[11,22,9].

До недавнего времени вероятностные методы применялись в теории автоматов в основном лишь к исследованию надежности схем, построенных из ненадежных элементов ^[34]. В работе М. Л. Цетлина^[50] такие методы

*) В. М. Г л у ш к о в, Об одном алгоритме синтеза абстрактных автоматов. Укр. матем. журнал, т. 12, 1960, № 2, стр. 147—156.

ыли применены в задаче о поведении автоматов в случайных средах.

Интересно отметить, что исследования в области абстрактной теории автоматов привели к взаимному проникновению понятий и методов теории автоматов, с одной стороны, и алгебры и математической логики — с другой стороны. На этой почве появились исследования, представляющие интерес с точки зрения тематики, традиционной для алгебры и для логики. К ним относятся, например, работы В. М. Глушкова^[12], Ю. И. Соркина^[40] и др., в которых изучается связь между автоматами и полугруппами. Р. Бюхи^[58] *) впервые обратил внимание на то, что идеи теории автоматов адекватны для решения чисто логических задач. Отметим еще, что понятие конечного автомата и его аналоги оказались полезными также в математической лингвистике^[62] и в динамическом программировании^[54].

Из этого краткого обзора видно, что в своем развитии теория конечных автоматов, возникшая первоначально под влиянием запросов вычислительной техники, вступает в контакт с постоянно расширяющимся кругом математических теорий и вырабатывает свою собственную математическую проблематику.

*) См. также работу Эльгота^[65].

ЛИТЕРАТУРА

1. Автоматы. Сб. статей, под ред. К. Э. Шеннона и Дж. Маккарти. Пер. с англ., под ред. А. А. Ляпунова, ИЛ, М., 1956.
2. Айзерман М. А., Гусев Л. А., Розоноэр Л. И., Смирнова И. М., Таль А. А., Конечные автоматы. I. «Автоматика и телемеханика», 1960, т. XXI, № 2; Конечные автоматы. II. «Автоматика и телемеханика», 1960, т. XXI, № 3.
3. Ауфенкамп Д., Хон Ф., Анализ последовательностей машин. Периодич. сб. переводов иностранных статей. «Математика», 3:3, ИЛ, М., 1959.
4. Базилевский Ю. Я., Вопросы теории временных логических функций. Сб. «Вопросы теории математических машин», Физматгиз, М., 1958.
5. Блох А. Ш., О задачах, решаемых последовательностными машинами. Проблемы кибернетики, вып. 3, Физматгиз, М., 1960.
6. Бонч-Бруевич А. М., Применение электронных ламп в экспериментальной физике, изд. 4-е, Гостехиздат, М., 1956.
7. Гаврилов М. А., Теория релейно-контактных схем. Изд-во АН СССР, М.—Л., 1950.
8. Гаврилов М. А., Основные формулы синтеза релейно-контактных схем. Автоматика и телемеханика, 1954, т. 15, № 6.
9. Гилберт Д. и Мур Э., Двоичные кодовые системы переменной длины. Кибернетический сборник, № 3, ИЛ, М., 1961.
10. Гилберт Д. и Аккерман В., Основы теоретической логики, ИЛ, М., 1947.
11. Глебский Ю. В., Кодирование с помощью автоматов, ДАН СССР, 1961, т. 141, № 5.
12. Глушков В. М., Абстрактная теория автоматов, УМН, т. XVI, вып. 5, 1961.
13. Глушков В. М., Некоторые проблемы синтеза цифровых автоматов, Вычисл. матем. и матем. физика, № 3, 1961.
14. Журавлев Ю. И., Об отделимости подмножеств n -мерного единичного куба. Тр. Математич. ин-та им. В. А. Стеклова, т. 51, Изд-во АН СССР, М.—Л., 1958.
15. Заволокина З. И., Магнитные элементы в цифровых вычислительных устройствах, Госэнергоиздат, М.—Л., 1958.
16. Карацуба А. А., Решение одной задачи из теории конечных автоматов, УМН, т. XV, вып. 3, 1960.
17. Клини С. К., Введение в метаматематику, перев. с англ., ИЛ, М., 1957.

18. Клипи С. К., Представление событий в нервных сетях и конечных автоматах. Сб. «Автоматы», ИЛ, М., 1956.
19. Кобринский Н. Е. и Трахтенброт Б. А., О построении общей теории логических сетей. Сб. «Логические исследования», Изд-во АН СССР, М., 1949.
20. Кудрявцев В. Б., Теорема полноты для одного класса автоматов без обратных связей, ДАН СССР, 1960, т. 132, № 2.
21. Кудрявцев В. Б., Вопросы полноты для систем автоматов, ДАН СССР, 1960, т. 130, № 6.
22. Левенштейн В. И., Самонастраивающиеся автоматы для декодирования сообщений, ДАН СССР, 1961, т. 141, № 6.
23. Летицевский А. А., О синтезе конечных автоматов, ДАН УССР, № 2, 1961.
24. Летицевский А. А., Условия полноты для конечных автоматов, Вычисл. матем. и матем. физика, № 4, 1961.
25. Лунц А. Г., Алгебраические методы анализа и синтеза контактных схем, Изв. АН СССР, серия матем., 1952, т. 10, № 5.
26. Лупанов О. Б., О возможностях синтеза схем из произвольных элементов. Тр. Математич. ин-та им. В. А. Стеклова, т. 51, Изд-во АН СССР, 1958.
27. Лупанов О. Б., Об одном методе синтеза схем. Изв. высших учебных заведений, Радиофизика, № 1, 1958.
28. Лупанов О. Б., О принципе локального кодирования и реализации функций из некоторых классов схемами из функциональных элементов, ДАН СССР, 1961, т. 140, № 2.
29. Мартынюк В. В., Соотношение между памятью и некоторыми возможностями конечного автомата. Проблемы кибернетики, вып. 5, Физматгиз, М., 1961.
30. Медведев Ю. Т., О классе событий, допускающих представление в конечном автомате. Сб. «Автоматы». Добавления, ИЛ, М., 1956.
31. Меерович Л. А., Зеличенко Л. Г., Импульсная техника. Изд. «Советское радио», М., 1952.
32. Муррей Ф. Д., Механизм и автоматы. Кибернетический сборник, вып. 1, ИЛ, М., 1960.
33. Мур Э. Ф., Умозрительные эксперименты с последовательными машинами. Сб. «Автоматы», ИЛ, М., 1956.
34. Нейман Дж., Вероятностная логика и синтез надежных организмов из ненадежных компонент. Сб. «Автоматы», ИЛ, М., 1956.
35. Поваров Г. Н., О логическом синтезе электронных вычислительных и управляющих схем. Сб. «Логические исследования», Изд. АН СССР, М., 1959.
36. Поваров Г. Н., Метод синтеза вычислительных и управляющих контактных схем. Автоматика и телемеханика, 1957, т. 18, № 2.
37. Рогинский В. Н. и Харкевич А. Д., Релейные схемы в телефонии, Связьиздат, М., 1955.
38. Синтез электронных вычислительных и управляющих схем, пер. с англ., ИЛ, М., 1954.

39. Скорняков Л. А., Об одном классе автоматов (нервные системы). Проблемы кибернетики, вып. 4, Физматгиз, М., 1960.
40. Сортин Ю. И., Алгебра автоматов. Проблемы кибернетики, вып. 5, Физматгиз, М., 1961.
41. Трахтенброт Б. А., Об операторах, реализуемых в логических сетях, ДАН СССР, 1957, т. 112, № 6.
42. Трахтенброт Б. А., Синтез логических сетей, операторы которых описаны средствами исчисления одноместных предикатов, ДАН СССР, 1958, т. 118, № 4.
43. Трахтенброт Б. А., К теории неповторных контактных схем. Тр. Математич. ин-та им. В. А. Стеклова, т. 51, ИЛ, М.—Л., 1958.
44. Трахтенброт Б. А., Асимптотическая оценка сложности логических сетей с памятью, ДАН СССР, 1959, т. 127, № 2.
45. Трахтенброт Б. А., Некоторые построения в логике одноместных предикатов, ДАН СССР, 1961, т. 138, № 2.
46. Трахтенброт Б. А., Конечные автоматы и логика одноместных предикатов, ДАН СССР, 1961, т. 140, № 2.
47. Фельдбаум А. А., Введение в теорию нелинейных цепей, Госэнергоиздат, М., 1948.
48. Цетлин М. Л., О непримитивных схемах. Проблемы кибернетики, № 1, Физматгиз, М., 1958.
49. Цетлин М. Л., Шехтман Л. М., Двухтактные ферротранзисторные схемы и алгебраический метод их синтеза. Проблемы кибернетики, № 2, Физматгиз, М., 1959.
50. Цетлин М. Л., Поведение конечных автоматов в случайных средах, Автоматика и телемеханика, 1961.
51. Шестаков В. И., Алгебраический метод синтеза многотактных релейных систем, ДАН СССР, 1954, т. 99, № 6.
52. Шестаков В. И., Моделирование операций исчисления высказываний посредством релейно-контактных схем. Сб. «Логические исследования», Изд-во АН СССР, М., 1959.
53. Шехтман Л. М., Алгебраический метод синтеза ферродиодной дроссельной схемы. Сб. «Вопросы радиоэлектроники», сер. 12, вып. 18, 1959.
54. Шрейдер Ю. А., Задача динамического планирования и автоматы. Проблемы кибернетики, вып. 5, Физматгиз, М., 1961.
55. Яблонский С. В., Функциональные построения в k -значной логике. Тр. Математич. ин-та им. В. А. Стеклова, т. 51, Изд-во АН СССР, М.—Л., 1958.
56. Яблонский С. В., Об алгоритмических трудностях синтеза минимальных контактных схем. Проблемы кибернетики, № 2, Физматгиз, М., 1959.
57. Яблонский С. В., Об одном семействе классов функций, допускающих простую схемную реализацию, УМН, т. 1, 1957.
58. Büchi J. R., Weak second-order arithmetic and finite automata. Zeitschr. Math. Logik und Grundl. der Math., B. 6, N1, 1960.
59. Burks A. W., Wright G. B., Theory of logical nets. Proc. IRE, v. 41, N 4, 1953.
60. Burks A. W., Copey J. M., The logical design of an idealized general-purpose computer, J. Franklin Inst., v. 261, N 3—4, 1956.

61. Burks A. W., Wang H., The logic of automata, Univ. of Michigan, Ann. Arbor, Mich. Eng. Res. Inst., 1956.
 62. Chomsky N., On the certain formal properties of grammars. Inform. and Control, v. 2, 1959.
 63. Church A., Application of recursive arithmetic in the theory of computers and automata. Univ. Michigan, p. 1—68, 1959.
 64. Copy J. M., Elgot C., Wright J. B., Realization of events by logical nets. J. Ass. Comp. Mach., v. 5, N 2, 1958.
 65. Elgot C., Decision problems of finite automata design and related arithmetics. Trans. Amer. Math. Soc., v. 98, N 1, 1961.
 66. Elgot C., Wright J., Quantifier elimination in a problem of logical design. Michigan Mathem. Journal, v. 6, N 1, 1959.
 67. Ginsburg S., On the length of the smallest uniform experiment which distinguishes the terminal states of machine. J. Ass. Comp. Mach., v. 5, 1958.
 68. Ginsburg S., Some remarks on abstract machines. Trans. Amer. Math. Soc., v. 96, N 3, 1960.
 69. Huffman D., The synthesis of sequential switching circuits. Franklin Inst., 1954, v. 257, N 3.
 70. Huffman D., The design and use of hazard-free switching networks. J. Ass. Comp. Mach., v. 4, N 1, 1957.
 71. Keister W., Richi A., Wachburn S., The design of switching networks. Journ. Ass. Comp. Mach., v. 4, N 1, 1957.
 72. Moisil Gr. C., Interbuintarea logicilor trivalente in teoria mecanismelor automate. Comunic rile Acad. RPR, t. 6, N 2, 3, 1956.
 73. Nelson E., Mathematical models of the logical structure of digital computers abstract. Comp. Automation, v. 5, N 10, 1956.
 74. Quine W., The problem of simplifying boolean functions. The Amer. Math. Monthly, v. 58, N 8, 1952.
 75. Rabin H., Scott D., Finite automata and their decision problems. IBM Research J., v. 3, 1959.
 76. Svoboda A., Synthesa releovych siti. Stroje na zpracovani informaci, Sborn. 2, Praha, 1954.
 77. Shannon C., The synthesis of two-terminal switching circuits. BSTJ, v. 28, N 1, 1949.
-

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

- Автомат 9
 - без памяти 10
 - дискретного действия 9
 - конечный 11, 84, 107
 - непрерывного действия 13
 - элементарный (см. также элемент) 107
- Автоматов теория 12
- Алгебра логики (см. также логика суждений) 27
- Алгебры логики формула 27
 - — функции 24
- Алфавит 85
 - внутренний (алфавит состояний) 99
- Анализ автомата 15, 16, 127, 190, 210
- Базис дерева 99
 - детерминированного оператора 99
 - конечного дерева 227
 - элементов логической сети 348
- Вес оператора 98 и д.
 - — логарифмический 350
- Ветвь дерева 97, 231
- Вынос иррегулярностей 259, 261
- Дерево (информативное, бесконечное) 92, 225 и д.
 - конечное 224
 - нагруженное 94
- Дешифратор 289, 290, 330, 332, 358 и д.
- Диаграмма переходов 193
- Дизъюнктивная нормальная форма 46
 - — — минимальная 46
 - — — совершенная 34
 - — — тупиковая 51
- Дизъюнкция (см. также логическое сложение) 25, 35, 46
 - элементарная 46
- Диод 137
- Дополнение (см. также отрицание) 25
- Достижимость 194, 228 и д.
- Инвертор 133
- Истинностно-полный набор элементов 126
- Исчисление предикатов (см. также логика предикатов) 69
- Исчисление предикатов расширенное 79
 - — — с ограниченными кванторами 80
- Канонические таблицы 103, 193
 - уравнения 99 и д.
- Каскад 122
- Катодный повторитель 136
 - — двойной 136
- Квантор 69, 151
 - общности 69
 - ограниченный 81, 274
 - существования 69
- Кодовые группы 24, 381 и д.
- Конечный автомат 11, 84, 107
- Конъюнктивная нормальная форма 46
 - — — минимальная 46
 - — — совершенная 36
- Конъюнкция (см. также логическое умножение) 25, 36, 46, 50, 51
 - элементарная 46
- Логика предикатов (см. также исчисление предикатов) 69
 - суждений (см. также алгебра логики) 27
- Логическая сеть 14, 115, 279 и д.
- Логическое сложение (см. также дизъюнкция) 25
- Логическое умножение (см. также конъюнкция) 25
- Матрица перехода 238, 300
- Машина Тьюринга 11
- Метод каскадов 317
 - неопределенных коэффициентов 53
- Минимизация алфавита состояний 191, 216
- Образование петли 122

- Объем внутренней памяти 108 и д.
 Оператор детерминированный (оператор без предвосхищения) 89 и д.
 — истинностный (без памяти) 89 и д.
 — класса g, n, k 350
 — константный 92, 208
 — монотонный 376
 — ограниченно-детерминированный (с конечным весом) 96, 98 и д.
 — остаточный 97
 Отрицание (см. также дополнение) 25
- Пентод 131, 135
 Переменная пропозициональная 28
 — свободная 70, 79
 — связанная 70
 Периодичность 204
 Подключение к выходу 122
 Полная система функций алгебры логики 41
 — — элементов 127, 348 и д.
 Полное дерево 225
 Полюса ранг (в схеме) 213
 Полюсы элемента (схемы) 108, 110
 Правильная петля 111
 Предвосхищение 89
 Предикат 68, 69, 245 и д.
 Преобразование тождественное 45, 76
- Различимость (операторов, состояний, букв, вершин, ветвей) 97, 199, 200, 203
 Разложение по регулярным спутникам 258
 — функций алгебры логики 33, 59
 Ранг вершины (ребра в дереве) 92
 — полюса (в схеме) 215
 Реализация оператора в масштабе времени 129, 183, 187
 — — со сдвигом 129
- Реализуемость операторов 84, 99, 123, 124
 Регулярная формула 257
 Регулярное выражение 272
 — событие 272
- Синтез автомата 15, 19, 128, 219
 — — логической сети 281 и д.
 Состояние 10, 84 и д.
 Спуск кванторов 252, 254
 Спутник формулы 256 и д.
 Степень достижимости 196, 228
 — различимости 201, 226
 Существенная зависимость функции от аргумента 59, 108
 Схема (автомата) 84, 109
 — пирамидальная 287, 317
 Счетчик четности 110
- Теория автоматов 12
 Триггер 154, 167
 Триод 86, 131, 142
 t -контролируемость 255
- Ферродиодный дроссельный элемент 178, 311
 Ферромагнитная трансформаторная схема 169, 172, 303
 Фиктивная переменная 108
 — ячейка 114, 150
 Формула алгебры логики 27
 Формулы эквивалентные 29
 Функции алгебры логики 24
 Функция класса $k_\alpha, k_\beta, k_\gamma, k_{\beta'}, k_{\gamma'}, 374$ и д.
 — — Шеннона $L(m, n, k)$ 351
 — — Шеффера 41
- Эксперименты 200
 Элемент (см. также автомат элементарный) 12, 107, 127, 348 и д.
 — задержки 101
 — функциональный 12

Опечатки

Стр.	Строчка	Напечатано	Должно быть
14	2 стр.	G. B. Wright	J. B. Wright
52	1 стр.	$x_2 \bar{x}_3 \vee x_2 \bar{x}_3$	$x_2 \bar{x}_3 \vee \bar{x}_2 x_3$
60	5 стр.	\bar{v}_i	\bar{v}_i
109	4 стр.	G. B. Wright,	J. B. Wright,
270	10—11 стр.	из-под формулы	из подформулы
306	8 стр.	$\bar{\Phi}_2(t) \vee$	$\Phi_2(t) \vee$
399	17 стр.	10. Гилберт	10. Гильберт
401	4 стр.	Wright G. B.	Wright J. B.

Н. Е. Кобринский и Б. А. Трахтенброт

Зак. 1246.