# КОМПЬЮТЕРНАЯ МАТЕМАТИКА

Д. КУК, Г. БЕЙЗ

Д. КУК, Г. БЕЙЗ

# КОМПЬЮТЕРНАЯ МАТЕМАТИКА

Перевод с английского Г. М. КОБЕЛЬКОВА

МОСКВА «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1990

ББК 22.18
К89
УДК 519.6

# COMPUTER MATHEMATICS

D. J. COOKE AND H. E. BEZ

Cambridge University Press
Cambridge

LONDON · NEW YORK · NEW ROCHELLE ·
· MELBOURNE · SYDNEY

Кук Д., Бейз Г. Компьютерная математика: Пер. с англ.— М.: Наука, Гл. ред. физ.-мат. лит., 1990.— 384 с.— ISBN 5-02-014216-6.

На основе фундаментальных понятий математики, введенных в начале, математически строго описывается ряд проблем и дается их решение. Изложение, где это возможно, носит строгий математический характер. Доказательства утверждений проводятся на конструктивном уровне. Дается большое количество примеров и упражнений, результаты которых, как правило, используются в дальнейшем.

Для студентов, аспирантов и научных работников, занимающихся вопросами компьютерной математики и ее приложениями.

Табл. 28. Ил. 164.

К $\frac{1602110000-056}{053(02)-90}$ 20-90

ISBN 5-02-014216-6

## ОГЛАВЛЕНИЕ

## ПРЕДИСЛОВИЕ

Вычисления являются точной наукой, и систематическое изучение всех аспектов, включая такие различные области, как разработка баз данных, проверка систем и создание математического обеспечения, с необходимостью вызывает использование математических моделей. С этой точки зрения многие учебные программы по вычислениям в университетах и институтах содержат специальные курсы, знакомящие студентов с соответствующими математическими структурами и методами. Содержание этой книги является таким курсом и занимает примерно 100 лекционных часов; курс читался студентам факультета компьютерных наук Технологического университета в Лафбаро. Материал книги соответствует первым двум годам обучения. Лекции первого года посещают все студенты. Курс лекций второго года является необязательным; он обеспечивает основу для курсов, читающихся на последнем году обучения. Содержание книги совершенствовалось в течение ряда лет. В течение этого времени мы ощущали помощь многих из окружающих нас людей, включая многочисленных коллег (как в Лафбаро, так и в других местах) и студентов, которые помогали при

подготовке и проверке материала. Мы особенно хотим поблагодарить наших жен — Крис и Кэрис — за постоянную поддержку и понимание в течение всего процесса написания книги. Особо благодарим Крис Кук за многочисленные часы, которые она провела, помогая нам в создании проекта рукописи, Орнеллу Ларднер за аккуратно напечатанный окончательный вариант и Алана Бенсона, который прочитал всю работу и сделал много конструктивных предложений. Ответственность за все оставшиеся ошибки и неточности принадлежит исключительно нам.

*Д. Кук, Г. Бейз*
Лафбаро, 1982

## ВВЕДЕНИЕ

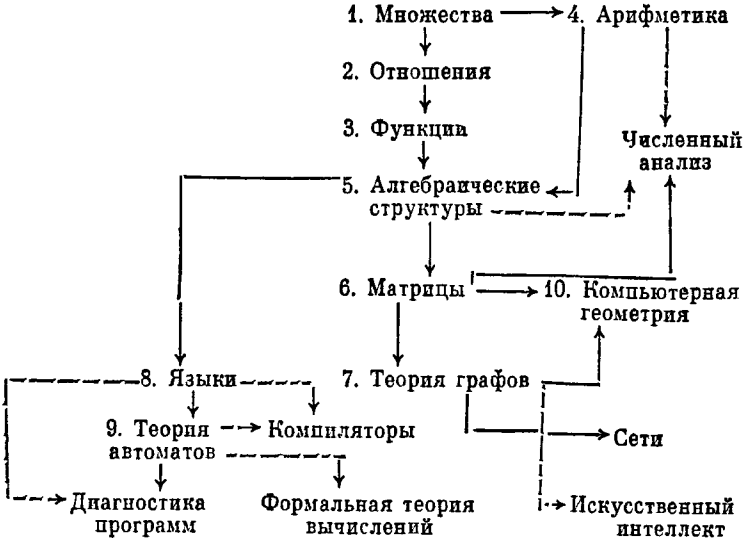
Книга в основном представляет собой курс лекций по компьютерной математике для университетов и институтов; однако она также может быть полезной для специалистов, работающих в этой области и желающих получить более глубокие знания предмета.

Книга содержит материал из тех областей современной математики, которые имеют отношения к вычислениям, и, как следствие, обеспечивает читателя средством для сжатого и точного описания многих проблем компьютерной науки. Несмотря на нашу приверженность излагать материал, непосредственно относящийся к компьютерной науке, в книге сделана попытка дать его разумное и строгое представление, приемлемое с математической точки зрения. Изложение по возможности носит конструктивный характер. Везде, где возможно, в каждой новой теме используются понятия и термины из предыдущих тем; материал сопровождается многочисленными упражнениями и примерами. Следует подчеркнуть, что разбор примеров и решение упражнений являются составной частью изучения предлагаемого материала.

Преследуя эту цель, мы должны с чего-то начать. Нашим исходным неопределяемым понятием является понятие множества, описываемое перечислением свойств, которыми оно обладает. Исходя из этого, можно определить все последующие понятия конструктивным и математически приемлемым образом. Такой подход необходим, поскольку любую ошибку легко можно проследить, вернувшись назад к неправильному предложению где-то в цепочке рассуждений. Это также означает, что часть или же вся рассматриваемая теория может быть запрограммирована.

Множества обсуждаются в гл. 1 и используются далее во всей книге. Главы 2, 3 и 5—7 образуют основу

строгого обсуждения многих разделов компьютерной математики; материал этих глав выбран таким образом, чтобы его можно было широко использовать. В гл. 4 описывается математическая модель арифметической системы, используемой в цифровых компьютерах при работе с целыми числами. Главы 8—10 связывают математическую теорию из предыдущих глав с разделами компьютерной математики. В частности, здесь выделены такие области, как теория языков, теория автоматов и компьютерная геометрия. Эти темы отражают одновременно и интересы авторов, и их желание изложить важные разделы современной математики, представляющие общий интерес, а не только прикладной. Другие области, где может применяться излагаемый в книге материал, включают в себя базы данных, сети, программную проверку и численный анализ. Главы 8—10 являются исходными для дальнейшего изучения таких тем, как компиляция, системы моделирования, теория вычислений, компьютерная графика, вычислительная геометрия и автоматическое проектирование. Логические внутренние связи между главами и другими изучаемыми областями показаны на следующей диаграмме:




Терминология и обозначения обычно строго вводятся в соответствующем месте текста. Однако иногда в примерах мы используем термины, которые раньше не опре-

делялись. В таких случаях обычно следует руководствоваться интуицией, а не формальными соображениями. Так, в одном из случаев в гл. 1 используется термин «конечная машина», в то время как определение дано лишь в гл. 9. В § 5 гл. 3 некоторые свойства действительных чисел используются без доказательства. Например, мы используем неравенство треугольника $|x+y| \leqslant |x|+|y|$, где $|\cdot|$ обозначает абсолютную величину числа, в некоторых доказательствах, относящихся к пределам. При желании читатель может опустить эти доказательства до прочтения из п. 3.4 гл. 5. Повсюду в книге символ // будет означать конец определения, примера, доказательства и т. п., а символ # — обнаружение логической ошибки, т. е. противоречие. Также по причинам, которые станут ясными в § 5 гл. 1, мы обычно будем использовать для обозначения операции умножения символ * (а не $\times$), хотя, когда мы будем иметь дело с обычными числами, символ * иногда будет опускаться с целью упрощения записи в больших выражениях.

ГЛАВА 1

# МНОЖЕСТВА

## § 1. Множества и их спецификация

Как уже отмечалось во введении, нам хотелось бы построить математическую теорию достаточно строго, однако при этом возникают некоторые трудности с обоснованием. Вместо этого мы начнем с описания таких начальных понятий, как множество. Хотя материал может показаться простым, это делается для того, чтобы читатель мог свободно разобрать примеры, приведенные в конце параграфа.

Множество — это совокупность определенных различаемых объектов таких, что для любого объекта можно установить, принадлежит этот объект данному множеству или нет.

Множество, которое подчиняется лишь такому ограничению, может содержать объекты почти любой природы. Например:

— множество всех подземных станций Лондона;

— множество левых ботинок;

— множество натуральных чисел: 1, 2, 3, 4 и т. д.;

— множество символов, доступных специальному печатающему устройству;

— множество кодов операций конкретного компьютера;

— множество зарезервированных слов языка Паскаль.

В конечном счете нас будут интересовать следующие множества:

— множество идентификаторов, встречающихся в определенной программе;

— множество операций в той же самой программе;

— множество операций, которые могут быть выполнены после данной инструкции в той же программе.

Однако эти множества также являются достаточно сложными для исследования. Поэтому для большинства примеров мы будем использовать некоторые абстрактные множества, такие как множества чисел.

Множества обычно обозначают прописными буквами, например $A$, и специфицируют одним из двух путей. Если множество содержит несколько элементов, то мы просто записываем все его элементы. Например, если мы определим $A$ как множество всех целых чисел строго между 6 и 10, то это можно записать следующим образом:

$$A = \{7, 8, 9\}$$

и прочитать как

«$A$ — множество, содержащее 7, 8, 9».

Здесь символ «=» используется в определенном смысле: $A$ равно множеству... Далее будет использоваться высказывание «равно ли $A$...». Поэтому мы должны предложить процедуру установления справедливости этого утверждения. Другими словами, множество можно охарактеризовать определенными свойствами, и, следовательно, множество $A$ можно определить как

$$A = \{x\colon\ x \text{ — целое число и } 6 < x < 10\}$$

и прочитать как

«$A$ есть множество всех $x$ таких, что...».

Множеству $A$ принадлежат только те элементы, которые являются целыми числами, большими 6 и меньшими 10, т. е. 7, 8 и 9, и, следовательно, мы имеем 7, 8, 9, как и ранее.

Множества часто рассматривают как «неупорядоченные совокупности элементов», хотя иногда полезно подчеркнуть, что, например,

$$\{7, 8, 9\} = \{8, 9, 7\} = \{9, 8, 7\} = \ldots;$$

мы не делаем никакой оговорки о порядке, в котором рассматриваются элементы, поэтому было бы неправильным допускать какой-либо определенный порядок.

Для любого заданного объекта можно определить, принадлежит ли он множеству $A$. В частности, если число принадлежит множеству, то будем говорить, что «оно является элементом множества». Так, например, если 7 является элементом множества $A$, то это утверждение может быть записано следующим образом:

«$7 \in A$».

Утверждение «6 не является элементом $A$» будем обозначать как

«$6 \notin A$».

Символ $\in$ происходит от греческой буквы $\varepsilon$. Отрицание обозначается через $\notin$. Такое обозначение отрицания операции (или операционного символа) является общим в математике и часто будет использоваться в дальнейшем.

Надо подчеркнуть, что следовало бы обратить большее внимание на спецификацию множеств. Для каждого множества должна быть записана его спецификация. Процесс образования множества может продолжаться бесконечно долго. В результате получается множество, соответствующее определению. Частичная спецификация может быть полезной в том случае, когда не ясно, принадлежит данный элемент множеству или нет.

До сих пор нам встречались символы $\{:\}$, $\{,,,\}$, $\in$ и $\notin$. Их применение кажется достаточно простым, однако оно требует определенных навыков, которые будут проиллюстрированы на следующих примерах.

Пример 1.1. Какие из приведенных определений множества являются правильными:

$A = \{1, 2, 3\}$,

$B = \{5, 6, 6, 7\}$,

$C = \{x: x \notin A\}$,

$D = \{A, C\}$,

$E = \{x: x = 1$ или $x = \{y\}$ и $y \in E\}$,

$F = \{$множества, которые не являются элементами самих себя$\} = \{x: x$ — множество и $x \notin x\}$?

Если число членов множества $A$ легко вычисляется и среди элементов множества нет повторений, то определение верно.

Множество $B$ выглядит также правильным, за исключением лишь того, что число 6 встречается дважды. Мы можем проверить, принадлежит ли элемент множеству или нет. Таким образом, это наиболее важное требование в определении множества выполнено. Следовательно, мы может рассматривать эту запись как верную и эквивалентную $\{5, 6, 7\}$. Однако в этой ситуации возникают следующие проблемы. Если мы рассмотрим первоначальное определение $B$ и выбросим одно из чисел 6 из множества, то мы, очевидно, будем иметь $6 \in B$ и $6 \notin B$. Возникает противоречие. Поэтому мы будем рассматривать повторение символов в определении множеств как упоминание одного и того же символа, а его дублирование как недо-

смотр; удаление повторяющихся символов образует основу для некоторых дальнейших математических рассуждений.

Множество $A$ содержит числа; это может вызвать недоумение, так как числа не существуют. Более точно, мы используем символы чисел; эти символы называются так же, как и числа. Поэтому $B$ — множество имен, и мы обычно используем имена, чтобы представить объекты (элементы), на которые ссылаемся. В вычислениях имена имеют особое значение, особенно в изучении семантики программных языков (смысла программ). Здесь не место входить в детальное обсуждение этих проблем; достаточно указать ловушки и необходимость адекватных спецификаций рассматриваемых объектов.

Возьмем, например, множество

$X =$ {«Введение в Паскаль»,

«Основы структурных данных», «Введение в Паскаль»}.

Это — множество названий двух книг с одним элементом, по невнимательности записанным дважды, или же это — множество трех книг, две из которых имеют одно и то же название? Если верно последнее, то две книги «Введение в Паскаль» следует разделить каким-либо способом. Из данной информации нельзя выяснить правильный ответ, поэтому в данном случае следует быть осторожным.

Определение $C$ также справедливо как $A$, так как, если $x \in A$, то $x \notin C$ и, если $x \notin A$, то $x \in C$. Множество $C$ очень велико: оно содержит «всё», за исключением чисел 1, 2 и 3. Обозначение «всё» выделено и, как мы скоро увидим, «опасно» с математической точки зрения.

Так как определения $D$, $A$ и $C$ представляют множества, то отсюда получаем, что определение $D$ также справедливо. Заметим, что это — множество множеств (ничего неверного в этом нет!) такое, что оно имеет только два элемента, в частности $1 \notin D$, даже если $1 \in A$ и $A \in D$. Это легко проверить, так как $1 \neq A$, $1 \neq C$ и только $A$ и $C$ являются элементами $D$.

Множество $E$ является первым примером рекурсивно определяемого множества; оно определяется (частично) в терминах самого себя. Конструктивный процесс продолжается бесконечно, поэтому мы должны иметь правило для определения элементов. Мы не можем записать их явно. Заметим, что $E$ не определяется полностью в терминах $E$. Мы должны знать о множестве что-то, что не

зависит от определения; в данном случае это то, что $1 \in E$. Имеем:

$$1 \in E, \quad \text{поэтому} \quad \{1\} \in E,$$
$$\{1\} \in E, \quad \text{поэтому} \quad \{\{1\}\} \in E,$$
$$\{\{1\}\} \in E, \quad \text{поэтому} \quad \{\{\{1\}\}\} \in E \text{ и т. д.}$$

Хотя конструктивный процесс неограничен, беря любой элемент и располагая достаточным временем, можно определить, содержится ли этот элемент в $E$.

Перейдем теперь к $F$; это достаточно трудная задача. Чтобы увидеть, почему $F$ не может существовать, мы сначала допустим существование, а затем продемонстрируем, что существует особый элемент (обозначим его через $y$) такой, что мы не можем определить, $y \in F$ или $y \notin F$. Вообще исследование «неудобного» примера, на котором мы можем показать логический изъян, проводится нелегко; однако в данном случае мы можем использовать само множество $F$. Чтобы прояснить дело, давайте обозначим это множество через $G$. Если, как мы предполагаем, $F$ — искомое множество, то или $G \in F$, или $G \notin F$. Рассмотрим два возможных случая:

а) $G \in F$. Тогда $G$ удовлетворяет условию содержания, т. е. $G \notin G$, и, следовательно, $G \notin F$;

б) $G \notin F$ говорит о том, что $G$ не удовлетворяет условию вхождения в $F$, и, следовательно, $G \in F$.

Следовательно, во всех случаях мы приходим к противоречию. Поэтому $F$ не может существовать. Где же была сделана ошибка? Множества множеств, вероятно, разрешаются, и бесконечно большие множества (например, рассмотренное выше множество $E$) также разрешаются; однако с «множеством всех множеств» нельзя работать в обычной теории множеств — это требует другого рода математики. Эта аномалия теории множеств известна как парадокс Рассела. Если мы уже имеем множество $H$, то можно определить $J$:

$$J = \{x\colon\ x \in H \text{ и } x \in x\}. \ //$$

Таким образом, мы будем использовать только множества, которые могут быть явно записаны или же построены путем хорошо определенных процессов. Поэтому множества не так тривиальны, как они могли вначале показаться. Однако, следуя приведенным выше правилам, работа с ними не будет особо трудной. Попытайтесь сделать самостоятельно следующее упражнение.

Упражнение 1.1.

1. Рассмотреть по крайней мере две возможные интерпретации множества

{Смит, Смит, Браун}.

Определить каждую из них настолько однозначно, насколько это возможно.

2. Рассмотреть следующие четыре множества. Выяснить, как записи этих множеств могут быть упрощены и какие из них эквивалентны. Предложить все возможные интерпретации, удовлетворяющие описанным выше предположениям:

а) {1, 2, 3, 4}; б) {I, II, III, IV, V};

в) {1, один, one, uno, ein}; г) {5, V, пять, five}.

3. Проверить справедливость утверждений

«Это утверждение неверно» и «я лгун».

Какие слова в утверждении требуют собственного (т. е. математически точного) определения для того, чтобы сделать ответы математически строгими?

4. Пусть $X$ — множество {1, 2}, а $Y$ — множество $\{x\colon\ x = y + z;\ y,\ z \in X\}$. Определить в явном виде множество $Y$. Какие это множества:

$\{y\colon\ y = x + z;\ x,\ z \in X\}$ и $\{y\colon\ x = y + z;\ x,\ z \in X\}$?

5. Предположим, что $x$ является элементом, а не множеством. Тогда $y \notin x$ для любого $y$, и отсюда следует, что $x \notin x$. Можно ли упростить множество $\{x, \{x\}, \{\{x\}\}\}$? Что можно сказать относительно $\{x, y, \{x, y\}\}$?

6. Пусть $A$ — множество всех целых чисел. Описать словами множество

$$X = \{x\colon\ x \in A \text{ и } x = 1 \text{ или } (x - 2) \in X\}.$$

## § 2. Простейшие операции над множествами

Как мы видели из рассмотрения «ошибочного множества» $F$ в примере 1.1 необходимо проявлять внимательность при определении множества. Тем не менее, строя новые множества из старых по простым правилам, можно безопасно получить много интересных примеров.

Позднее будут выписаны формальные правила, которым должны удовлетворять операции с множествами, а сейчас введем некоторые обозначения. Начнем с простейших операций.

Определение. Пусть даны множества $A$ и $B$. *Пересечением* множеств $A$ и $B$ называется множество всех элементов, принадлежащих $A$ и $B$, и обозначается $A \cap B$; таким образом,

$$A \cap B = \{x: x \in A \text{ и } x \in B\}.$$

Аналогично *объединение* $A$ и $B$ обозначается $A \cup B$ и определяется следующим образом:

$$A \cup B = \{x: x \in A \text{ или } x \in B\}.$$

Значения этих обозначений нетрудно запомнить, но иногда бывают ошибки. Один из путей запомнить, какой символ обозначает какую операцию,— объединить символы в слове и записать «Пересечение» и «Объединение». Эти определения выводятся из слов «и» и «или», и, как следствие, мы имеем

$$A \cup B = B \cup A, \quad A \cap B = B \cap A$$

и, что, вероятно, менее очевидно,

$$A \cup A = A, \quad A \cap A = A.$$

Эти тождества важны по двум причинам. Во-первых, из дальнейших математических рассуждений будет видно, что иногда следует сводить $A \cup A$ (соответственно $A \cap A$) к $A$ или, наоборот, расширять $A$ до $A \cup A$ (соответственно $A \cap A$). Во-вторых, можно не обратить на это внимание из-за того, что, выраженные словами, эти тождества могут казаться лишенными смысла даже тогда, когда они логически верны.

Заметим также, что определение объединения использует включение «или», называемое так потому, что оно включает «и» так, что

$$\{1, 2\} \cup \{2, 3\} = \{1, 2, 3\},$$
$$\{1, 2\} \cap \{2, 3\} = \{2\}.$$

Элементы в пересечении множеств (в данном случае это — единственное число 2) включаются в объединение. Это обычная математическая договоренность, и существует пример, в котором математическое значение является более точным, чем при общем употреблении.

Пример 2.1. В предположении, что каждый день или дождливый, или ясный, математическим (или логическим) ответом на вопрос

«Ясно или дождливо сегодня?»

будет

«Да».

Определение. *Разность множеств* $A$ и $B$ (также называемая *дополнением* $B$ до $A$) записывается в виде $A\backslash B$ и определяется соотношением

$$A\backslash B = \{x:\ x \in A \text{ и } x \notin B\}.$$

Поэтому, если $A = \{1, 2, 3\}$ и $B = \{2, 3, 4\}$, то $A\backslash B = \{1\}$ и $B\backslash A = \{4\}$.

Следующее определение включено для полноты. Хотя мы будем редко использовать его непосредственно, однако, как мы увидим в дальнейшем, этот оператор имеет большое значение в машинной арифметике.

Определение. *Симметрическая разность* множеств $A$ и $B$, т. е. $A\Delta B$, определяется как

$$A\Delta B = (A \cup B)\backslash(A \cap B).$$

Возможно, читателя запутали обозначения $\cup$, $\cap$, $\backslash$, $\Delta$, или, наоборот, он поверил, что они настолько элементарны, что не имеют никакого практического применения. Следующие примеры помогут в этом разобраться.

Пример 2.2. Предположим, что мы имеем две программы, называемые $P$ и $Q$, и что $A$ — множество всех значений данных, доступных $P$, а $B$ — множество всех значений данных, доступных $Q$. Тогда $A \cap B$ — множество всех данных, доступных $P$ и $Q$; $A \cup B$ — множество всех данных, доступных по крайней мере или $P$, или $Q$; $A\backslash B$ — множество всех данных, доступных $P$, но недоступных $Q$; $B\backslash A$ — множество всех данных, доступных $Q$, но недоступных $P$; $A\Delta B$ — множество всех данных, доступных только одной из программ $P$ или $Q$.

Чтобы полностью определить $A$ и $B$, мы должны знать некоторые данные о вычислениях, связанные с $P$ и $Q$. В нашем случае достаточно сказать, что они производятся на некоторой конечной ЭВМ.

Перед дальнейшим изложением будет удобно определить два специальных множества. Первое из них — пустое множество.

Определение. *Пустое множество* (обозначается $\varnothing$) есть множество, обладающее свойством

$$x \notin \varnothing \quad \text{при любом } x.$$

Второе множество, определение которого зависит от задачи, называют универсальным множеством.

Определение. *Универсальное множество* (обозначается $\mathscr{E}$) есть множество всех рассматриваемых в данной задаче элементов.

Ограничение $\mathscr{E}$ в этом случае помогает избежать трудностей, подобных тем, которые возникали при рассмотрении «множества» $F$ примера 1.1; в любом случае большинство элементов несущественно в каждой данной задаче. Например, размеры «Английского словаря», несомненно, неинтересны, если рассматривается поведение отдельной программы на Фортране.

Определение. Два множества $A$ и $B$ *не пересекаются*, если $A \cap B = \varnothing$.

Определение. В каждом случае, когда $\mathscr{E}$ задано, определим *дополнение* множества $A$ (обозначается $A'$) как

$$A' = \mathscr{E} \backslash A = \{x\colon\ x \notin A\}.$$

Из определений $\varnothing$, $\mathscr{E}$ и $A'$ следует справедливость тождеств

$$A \cup A' = \mathscr{E}, \quad A \cap A' = \varnothing.$$

В § 4 будет показано, что для данного $\mathscr{E}$ эти тождества достаточны, чтобы однозначно определить $A'$.

Пример 2.3. Пусть

$$\mathscr{E} = \{1, 2, 3, 4\}, \quad A = \{1, 3, 4\}, \quad B = \{2, 3\}, \quad C = \{1, 4\}.$$

Из определений легко найти, например, $A'$, $B \cap C$, $C \backslash A$ и т. д. Однако может понадобиться исследовать более сложные выражения, включающие две или более операций. Поскольку в этих случаях встает вопрос, как определить порядок, в котором мы должны осуществлять элементарные операции над множествами, будем использовать скобки. Каждое выражение, заключенное в скобки, должно быть выполнено перед тем, как его результат может быть использован в других вычислениях. Например, в $(A \cap B)'$ пересечение ($\{3\}$) вычисляется раньше, чем дополнение ($\{1, 2, 4\}$). Этого соглашения, очевидно, достаточно. Однако, чтобы избежать такого множества скобок, мы не будем требовать скобок, когда хотим произвести операцию дополнения перед любой из операций с множествами $\{\cup, \cap, \backslash, \Delta\}$. Поэтому $A \Delta B'$ означает $A \cap (B')$ и т. д. Следовательно,

$$A \cap B' = A \cap \{1, 4\} = \{1, 4\},$$
$$(A \cap B)' = \{3\}' = \{1, 2, 4\},$$
$$(B \backslash A) \cup C = \{2\} \cup C = \{1, 2, 4\}. \ //$$

Читатель может быть удивлен, почему изложение построено на понятии множества, а не числа. Действительно, до сих пор мы использовали числа только как элементы множеств. Это делалось лишь для того, чтобы читатель познакомился с объектами, с которыми ему придется работать. Дело в том, что существуют множества более сложные, чем числа; мы можем получить числа из множеств, но не наоборот. Однако для многих приложений последующей теории необходимо сделать точные утверждения о некоторых специальных множествах чисел. Чтобы обеспечить основу, с помощью которой будут конструироваться такие множества, определим множество **N** целых положительных чисел (натуральных чисел):

$$\mathbf{N} = \{1, 2, 3, \ldots\}.$$

Точное определение множества **N** вместе с арифметическими операциями $+$ и $*$ и его упорядочивание будут даны ниже. Однако в настоящей главе мы будем предполагать, что читатель знаком с некоторыми свойствами **N**. Аналогично **Z** определяют как множество всех целых чисел:

$$\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}.$$

Конечно, множества **N** и **Z** не могут быть выписаны явно (они достаточно велики), но в настоящее время мы должны понимать «...» как «и так далее».

Рассмотрим теперь множество

$$A = \{1, 2, \ldots, n\} = \{x\colon x \in \mathbf{N},\ 1 \leqslant x \leqslant n\}.$$

Оно имеет $n$ элементов. Будем говорить, что *мощность* (или *размер*, *норма*, *длина*) этого множества есть $n$. Это обозначается как

$$|A| = \operatorname{card}(A) = n.$$

Далее любое множество $B$, которое имеет то же число элементов, что и $A$, имеет такую же мощность, и, конечно, эти элементы не надо пересчитывать. Для небольших множеств достаточно легко пересчитать элементы, но для других множеств (например, **N**) это может быть невозможно. Далее мы дадим строгое, но в то же время неформальное правило для вычисления количества элементов.

Определение. Говорят, что множество $X$ *конечно*, если $X = \varnothing$ или если для некоторого $n \in \mathbf{N}$ существует множество $\{1, 2, \ldots, n\}$ такое, что оно имеет то же самое

число элементов, что и $X$. Если $X \neq \varnothing$ и никакого $n$ не может быть найдено, то $X$ называют *бесконечным*.

Сейчас, когда мы ввели некоторые определения, можно сформулировать несколько упражнений. Чтобы сделать множества легко записываемыми, мы снова будем использовать числа и буквы для элементов, но будем помнить, что те же операции могут быть применены к произвольным множествам.

Упражнение 1.2.

1. Пусть

$$\mathscr{E} = \{1, 2, 3, 4\}, \quad X = \{1, 5\},$$
$$Y = \{1, 2, 4\}, \quad Z = \{2, 5\}.$$

Найти множества:

а) $X \cap Y'$; б) $(X \cap Z) \cup Y'$;
в) $X \cup (Y \cap Z)$; г) $(X \cup Y) \cap (X \cup Z)$;
д) $(X \cup Y)'$; е) $X' \cap Y'$; ж) $(X \cap Y)'$;
з) $(X \cup Y) \cup Z$; и) $X \cup (Y \cup Z)$; к) $X \backslash Z$;
л) $(X \backslash Z) \cup (Y \backslash Z)$.

2. Пусть

$$\mathscr{E} = \{a, b, c, d, e, f\}, \quad A = \{a, b, c\},$$
$$B = \{f, e, c, a\}, \quad C = \{d, e, f\}.$$

Найти множества:

а) $A \backslash C$; б) $B \backslash C$; в) $C \backslash B$;
г) $A \backslash B$; д) $A' \cup B$; е) $B \cap A'$;
ж) $A \cap C$; з) $C \cap A$; и) $C \Delta A$.

3. Даны два произвольных множества $A$ и $B$ такие, что $A \cap B = \varnothing$. Что представляют собой множества $A \backslash B$ и $B \backslash A$?

4. Даны два произвольных множества $C$ и $D$ такие, что $C \cap D' = \varnothing$. Что можно сказать о $C \cap D$ и $C \cup D$?

5. Дано произвольное множество $X$. Найти множества:

а) $X \cap X'$; б) $X \cup X'$; в) $X \backslash X'$.

6. Какие из следующих утверждений справедливы:

а) $0 \in \varnothing$; б) $\varnothing = \{0\}$; в) $|\{\varnothing\}| = 1$;
г) $\{\{\varnothing\}\} \in \{\{\{\varnothing\}\}\}$; д) $|\{\{\varnothing\}\}| = 2$?

Это вопрос коварный. Хотя это может показаться простым или надуманным, пустое множество и его свойства являются достаточно важными. Если вы не совсем уверены в ответе, проработайте вопрос, используя аналогию портфеля вместо множества. Таким образом, $\{\{\}, \{\}\}$ — портфель, содержащий два пустых портфеля, и, следовательно, $|\{\{\}, \{\}\}| = 2$ и т. д.

7. Пусть $M$ и $N$ — два конечных компьютера (см. пример 2.2) с фиксированными программами. Далее пусть $A$ — множество значений данных, доступных $M$ и таких, что если $x \in A$ и машина $M$ работает с входным словом $x$, то $M$ останавливается и выдает результат. Аналогично пусть $B$ — множество значений данных, которые приводят $N$ к остановке и выдаче результата. Если любой элемент $A$ доступен $M$ и $N$, что мы может сказать об элементах $B'$? Объяснить эту ситуацию с помощью символов и пояснить бесполезность этой информации.

8. Объяснить в терминах множеств, почему пример 2.1 верен.

9. При определении операции объединения подчеркивалось, что мы использовали включение «или». Как в терминах множеств можно выразить исключающее «или»?

10. Часто в вычислениях будут использоваться арифметические операции для образования новых множеств. Так, если $A$ и $B$ — множества чисел, то

$$A + B = \{x\colon\ x = a + b,\ a \in A,\ b \in B\}.$$

Аналогично определяются операции $*$, $-$, $/$ между множествами чисел. Найти следующие множества:

а) $\{1, 2\} + \{1, 3\}$;
б) $\{1, 2\} \cup \{1, 3\}$;
в) $\{1, 2\} * \{1, 3\}$;
г) $\{1, 2\} \cap \{1, 3\}$;
д) $\{1, 2\} - \{1, 3\}$;
е) $\{1, 2\} \setminus \{1, 3\}$;
ж) $\{2, 4\}/\{2\}$;
з) $\{2, 4\} \setminus \{2\}$;
и) $\{2, 4\} - \{2\}$.

11. Для того чтобы быть в состоянии применить технику операций с множествами к конкретной задаче, мы неизбежно должны на некотором этапе взять «нематематическое» утверждение и перевести его в математическое. Обычно (но не всегда) это делает описание более компактным. Однако математическое выражение будет всегда математически строгим, тогда как исходное выражение может таким не быть. (Где это случается, требуется найти, чего же недостает в первоначальной формулировке.) Теперь надо:

а) попытаться сформулировать следующие утверждения на языке множеств:

— даны множества $A$, $B$ и $C$; определить множество, включающее в себя только два из этих множеств;

— решить предыдущую задачу при условии, что $A$, $B$ и $C$ взаимно не пересекаются;

— даны множества $V$, $W$, $Y$, $X$ и $Z$. Определить множество, включающее по крайней мере два из множеств $V$, $W$, $X$ и $Y$ и не включающее $Z$;

б) аналогично описать словами следующие множества:

$$(J \cap (K \cup L))' \cup (H \backslash L),$$

$$(P \cup R \cup Q) \backslash (P \cap (Q \backslash R)),$$

$$((E \backslash F) \cup (F \backslash E))' \cup G.$$

## § 3. Диаграммы Венна

Уже можно было заметить некоторые специфические свойства операций над множествами, в особенности то свойство, что одно и то же множество может быть определено различными путями. Далее в этой главе мы обсудим способы доказательства этих свойств формальным путем, однако часто полезно иметь геометрические представления множеств. Такие представления не могут заменить доказательства, но могут быть полезны, чтобы быстро и просто убедиться, справедливо ли конкретное утверждение и, следовательно, доказательство его возможно или же оно неверно. В этом случае можно заметить, как следует строить пример, чтобы доказать, что оно неверно. Диаграммы, которые мы будем использовать, называют диаграммами Венна (по имени английского математика Джона Венна) и строят, как это описано ниже.

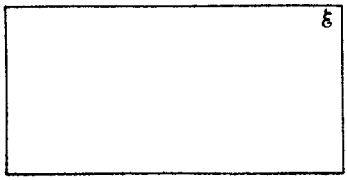



Рис. 1.1

Во-первых, начертим большой прямоугольник, представляющий $\mathscr{E}$ (рис. 1.1). Во-вторых, начертим круги (или какие-либо другие подходящие замкнутые кривые) внутри прямоугольника, чтобы представить множества. Они должны пересекаться в наиболее общем случае, требуемом в задаче, и должны быть соответствующим образом обозначены (рис. 1.2). Точки, которые лежат внутри различных областей диаграммы, сейчас могут рассматриваться как элементы соответствующих множеств. Если число элементов в множествах мало, тогда отдельные элементы могут быть записаны внутри подходящих областей, как это показано в примере 3.1.

Пример 3.1. Пусть $\mathscr{E} = \{b, c, d, e\}$, $A = \{b, c, d\}$, $B = \{c, e\}$. Соответствующая диаграмма изображена на

рис. 1.3. Этот рисунок полностью иллюстрирует пример 3.1, обеспечивая знание элементов $\mathscr{E}$. Если же, например, $A \in \mathscr{E}$, тогда неясно, что предполагалось изобразить на диаграмме. В тех случаях, когда используются

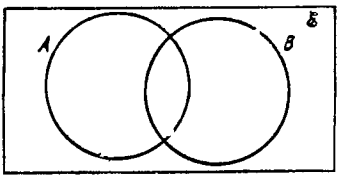



Рис. 1.2

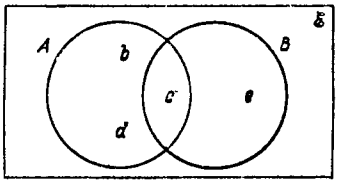



Рис. 1.3

более сложные конструкции множеств, следует избегать изображения их в виде диаграмм.

Имея построенную подходящим образом диаграмму, мы можем заштриховать определенные области для обозначения вновь образованных множеств.

Пример 3.2. Чтобы представить множество $A \cup (B' \cap C)$, начнем с общей диаграммы, показанной на рис. 1.4. Заштрихуем $B'$ диагональными линиями в одном направлении, а $C$ диагональными линиями в другом

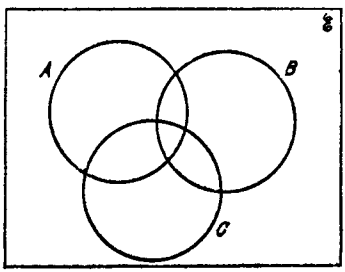



Рис. 1.4

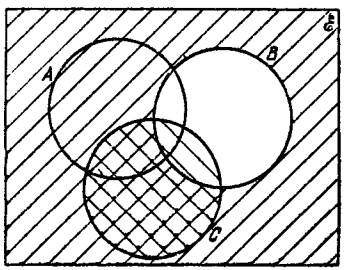



Рис. 1.5

направлении (рис. 1.5). Площадь с двойной штриховкой представляет собой множество $B' \cap C$.

На новой копии диаграммы заштрихуем эту область горизонтальными линиями, а $A$ вертикальными. Вся заштрихованная на рис. 1.6 область представляет множество $A \cup (B' \cap C)$. Если в отдельных случаях мы имеем дополнительную информацию о рассматриваемых множествах, то ее можно использовать для упрощения диаграммы Венна.

Пример 3.3. Пусть $A \cap B = \varnothing$; это соответствует диаграмме на рис. 1.7. //

Заметим, что в большинстве случаев множества содержат довольно много элементов, и, следовательно, эти элементы не могут быть представлены отдельно. Поэтому

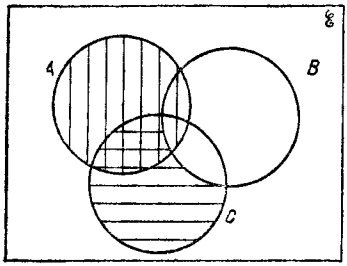



Рис. 1.6

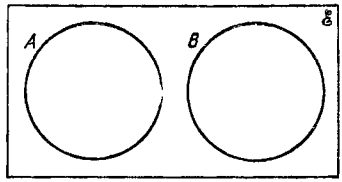



Рис. 1.7

более удобно в этом случае говорить о каждом из множеств как о целом и не упоминать отдельных элементов.

Упражнение 1.3.

1. Начертить диаграмму, иллюстрирующую построение множеств, рассматриваемых в задаче 1 упражнения 1.2.

2. Как можно представить следующие множества, используя диаграммы Венна:

$$\{A, \{A\}\}, \{\{a\}, \{b\}\}, \{X, Y, Z\},$$

где

$$X = \{x\colon\ x = 1 \text{ или } (x-2) \in X\},$$

$$Y = \{x\colon\ x = 3 \text{ или } (x-3) \in Y\},$$

$$Z = \{x\colon\ x = 2 \text{ или } (x-2) \in Z\}?$$

## § 4. Подмножества и доказательства

Операции пересечения, объединения, разности и дополнения позволяют нам формировать новые множества. Однако, как правило, мы не можем сказать, как одно множество соотносится с другими. Например, пусть даны два множества $X$ и $Y$; пересечение $X \cap Y$ в некотором смысле «меньше» (или по крайней мере не больше), чем $X$. Действительно, все элементы множества $X \cap Y$ принадлежат также множеству $X$. Из этого наблюдения можно формально определить равенство множеств и различных выражений для того же самого множества. С по-

мощью этих определений мы также в состоянии написать подходящие логические доказательства важных фактов, относящихся к множествам. Эти результаты, хотя и очевидны, обеспечивают подходящие ситуации, в которых можно ввести некоторые из основных способов доказательств, используемых в дальнейшем.

Определение. Пусть множества $A$ и $B$ таковы, что из принадлежности $x \in A$ следует, что $x \in B$. Тогда говорят, что $A$ есть *подмножество* $B$, и обозначают это как $A \subseteq B$. Соответствующая диаграмма Венна изображена на рис. 1.8. Далее, если существует элемент $B$, который не принадлежит $A$, то $A$ называют *собственным подмножеством* $B$ и записывают в виде $A \subset B$. Это означает, что в некотором смысле $B$ больше, чем $A$, но, как мы впоследствии увидим, такие термины могут вводить в заблуждение. Следовательно, при употреблении этого термина требуется проявлять осторожность. Эти отношения могут также быть записаны в обратном порядке, или

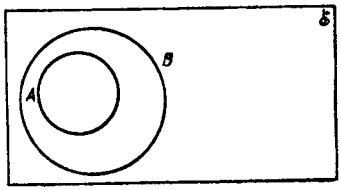


Рис. 1.8

$$B \supset A \quad \text{и} \quad B \supseteq A;$$

тогда говорят, что $B$ — (собственное) *надмножество* $A$.

Очевидно, что для любого множества $A$ справедливы следующие три соотношения:

$$\varnothing \subseteq A, \quad A \subseteq A, \quad A \subseteq \mathscr{E}.$$

Второе из них является наиболее важным. Говорят, что множества $A$ и $B$ *эквивалентны* (записывается как $A = B$), если

$$A \subseteq B \quad \text{и} \quad B \subseteq A.$$

Это означает, что все элементы $A$ являются элементами $B$, а все элементы $B$ — элементами $A$. //

Используя определение эквивалентности множеств, докажем теперь идентичность некоторых множеств, разобрав серию из пяти примеров. Они являются примерами доказательств. Поэтому мы обязаны кратко рассмотреть вопрос о том, почему следует заботиться о доказательствах в компьютерной науке. Строго говоря, как это обычно делается, мы должны доказать что-либо только один раз. Затем мы исходим из того, что эта часть инфор-

мации является правильной и, следовательно, может рассматриваться как факт. Однако, как бывает в большинстве аспектов вычислительной науки, метод, которым мы получаем результат, по крайней мере так же важен, как и сам результат. После анализа доказательства становятся ясными сделанные предположения и последствия, к которым они приводят, а также становятся понятными процессы вывода, которые можно использовать при решении других задач. (Аналогично проведению эксперимента на ЭВМ с подобными структурами данных при их использовании в программировании.)

Можно также заметить, что доказательства теорем формируют основу всех решающихся автоматически задач, но это будет обсуждаться позднее. Рассмотрим несколько примеров.

В примере 4.1 непосредственно доказывается справедливость следующих двух соотношений:

а) $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$;
б) $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$.

Каждое из этих доказательств состоит из последовательности утверждений вида

«если $P$, то $Q$»

(если справедливо $P$, то справедливо и $Q$). Для удобства запишем это утверждение как «$P \Rightarrow Q$» и будем читать

«из $P$ следует $Q$».

Следовательно, если имеется последовательность $P_0$, $P_1, \ldots, P_n$ такая, что $P_0 \Rightarrow P_1, P_1 \Rightarrow P_2, \ldots, P_{n-1} \Rightarrow P_n$ (из $P_0$ следует $P_1$, из $P_1$ следует $P_2$, ..., из $P_{n-1}$ следует $P_n$), то мы имеем прямое доказательство $P_0 \Rightarrow P_n$.

Пример 4.1. Относительно множеств $A$, $B$ и $C$ докажем, что

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

Доказательство.

$$x \in A \cap (B \cup C) \Rightarrow x \in A \text{ и } x \in (B \cup C) \Rightarrow$$

(Такая группировка необходима, так как скобки означают, что объединение следует вычислить перед пересечением.)

$$\Rightarrow x \in A \text{ и } (x \in B \text{ или } x \in C) \Rightarrow$$
$$\Rightarrow (x \in A \text{ и } x \in B) \text{ или } (x \in A \text{ и } x \in C) \Rightarrow$$
$$\Rightarrow (x \in A \cap B) \text{ или } (x \in A \cap C) \Rightarrow x \in (A \cap B) \cup (A \cap C).$$

Таким образом,

$$A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C).$$

Сейчас необходимо доказать включение $\subseteq$ в обратную сторону:

$$x \in (A \cap B) \cup (A \cap C) \Rightarrow$$
$$\Rightarrow (x \in A \cap B) \text{ или } (x \in A \cap C) \Rightarrow$$
$$\Rightarrow (x \in A \text{ и } x \in B) \text{ или } (x \in A \text{ и } x \in C) \Rightarrow$$
$$\Rightarrow x \in A \text{ и } (x \in B \text{ или } x \in C) \Rightarrow$$
$$\Rightarrow x \in A \text{ и } x \in B \cup C \Rightarrow x \in A \cap (B \cup C).$$

Следовательно,

$$(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C).$$

Поэтому

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C). \; //$$

В этом частном случае вторая часть доказательства точно совпадает с первой, и, следовательно, можно записать

$$x \in A \cap (B \cup C) \Leftrightarrow x \in A \text{ и } x \in B \cup C \text{ и т. д.}$$

Здесь символ «$\Leftrightarrow$» («$P \Leftrightarrow Q$») означает, что $P \Rightarrow Q$ и $Q \Rightarrow P$. Он может читаться как «тогда и только тогда» (иногда также читают «если и только если») и означает эквивалентность двух утверждений $P$ и $Q$. Однако не всегда так просто обратить аргумент и следствие. В общем случае мы должны провести доказательства в обе стороны раздельно. Заметим также, что эквивалентность может быть легко получена (хотя и не доказана) из подходящей диаграммы Венна, однако не всегда можно начертить диаграмму, относительно которой можно быть уверенным, что она настолько отвечает требованиям, насколько необходимо. Поэтому непосредственное доказательство необходимо. Это доказательство зависит от внутренних взаимосвязей между значениями «и» и «или» и, следовательно, может быть выбрано для каждого случая своим. Далее, когда будут определены некоторые алгебраические структуры, мы покажем, что это не составляет трудностей.

Примеры 4.2—4.5 также используют прямые доказательства, однако эти доказательства записаны в несколько другом виде.

Пример 4.2. Относительно данного множества $\mathscr{E}$ дополнение любого множества $A (A \subseteq \mathscr{E})$ единственно.

Доказательство. Предположим, что существует два множества $B$ и $C$, каждое из которых удовлетворяет требованиям дополнения $A$, т. е.

$$B \cap A = C \cap A = \varnothing \quad \text{и} \quad B \cup A = C \cup A = \mathscr{E}.$$

Тогда

$$B = B \cap \mathscr{E} = B \cap (C \cup A) = (B \cap C) \cup (B \cap A) =$$
$$= (B \cap C) \cup \varnothing = B \cap C;$$

поэтому

$$x \in B \Rightarrow x \in B \text{ и } x \in C \Rightarrow B \subseteq B \cap C \Rightarrow B \subseteq B \text{ и } B \subseteq C.$$

Однако мы знаем, что $B \subseteq B$. Поэтому отсюда следует, что

$$B \subseteq C.$$

Аналогично (меняя ролями $B$ и $C$) получаем

$$C \subseteq B,$$

откуда

$B = C$, т. е. $B = C = A'$, и $A'$ единственно. //

Приведенный выше пример содержит в себе основной математический подход, употребляемый для доказательства единственности,— сначала предполагается, что существуют два таких объекта, а затем доказывается, что они совпадают.

В следующем примере мы опять прибегнем к предположениям об «или», чтобы иметь возможность написать выражение $A \cup B \cup C$ как $A \cup (B \cup C)$ или $(A \cup B) \cup C$, когда это будет более удобно для требуемых преобразований.

Пример 4.3. Даны множества $A$, $B$ и $C$ такие, что

$$A \cup B \cup C = \mathscr{E}$$

и $A$, $B$ и $C$ попарно не пересекаются. Тогда

$$A' = B \cup C,\ B' = A \cup C \text{ и } C' = A \cup C.$$

Доказательство.

$$A \cup B \cup C = A \cup (B \cup C) = \mathscr{E},$$
$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) = \varnothing \cup \varnothing = \varnothing.$$

Следовательно, $B \cup C$ удовлетворяет условиям для $A'$, которое единственно. Поэтому $A' = B \cup C$. Аналогично проводятся доказательства для $B'$ и $C'$. //

Пример 4.4. Для произвольных множеств $X$ и $Y$ справедливо соотношение

$$(X \cap Y)' = (X \cap Y') \cup (X' \cap Y) \cup (X' \cap Y').$$

Доказательство. Предположим, что мы имеем множества $A$, $B$ и $C$ из примера 4.3 и

$$C = D \cup E \quad \text{и} \quad D \cap E = \varnothing.$$

(Диаграмма Венна на рис. 1.9 может разбить $\mathscr{E}$, как требуется.) Тогда множества $A$, $B$, $D$ и $E$ взаимно не пересекающиеся и

$$A \cup B \cup D \cup E = \mathscr{E}.$$

Более того,

$$A' = B \cup C,$$

поэтому

$$A' = B \cup D \cup E.$$

Сейчас легко показать, что если положить

$$A = X \cap Y, \quad B = X \cap Y', \quad D = X \cap Y$$

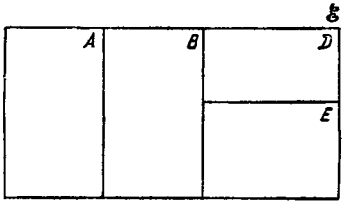



Рис. 1.9

и $E = X' \cap Y'$, то требуемые условия будут выполнены, и поэтому нужный результат немедленно следует из предыдущих рассмотрений. //

Пример 4.5. Для любых множеств $X$ и $Y$ верно соотношение

$$(X \cap Y)' = X' \cup Y'.$$

Доказательство.

$$x \in (X \cap Y)' \Leftrightarrow x \in (X \cap Y)' \cup (X' \cap Y) \cup (X' \cap Y') \Leftrightarrow$$
$$\Leftrightarrow x \in (X \cap Y)' \cup (X' \cap Y') \cup (X' \cap Y) \cup (X' \cap Y') \Leftrightarrow$$

(Один член здесь продублирован. Это разрешается, так как $A = A \cup A$ для любого $A$.)

$$\Leftrightarrow x \in ((X \cap Y') \cup (X' \cap Y')) \cup ((X' \cap Y) \cup (X' \cap Y')) \Leftrightarrow$$
$$\Leftrightarrow x \in ((X \cup X') \cap Y') \cup (X' \cap (Y \cup Y')) \Leftrightarrow x \in (\mathscr{E} \cap Y') \cup$$
$$\cup (X' \cap \mathscr{E}) \Leftrightarrow x \in Y' \cup X' \Leftrightarrow x \in X' \cup Y'.$$

Следовательно, $(X \cap Y) = X' \cup Y'$. //

Результат, полученный в примере 4.5, и подобные ему (см. упражнение 4.5) называют *законами де Моргана.*

Они играют важную роль в математической логике. Наиболее непосредственные приложения к вычислениям находятся в области комбинаторных цепей в логике.

Последовательность примеров 4.1—4.5 иллюстрирует, как можно развивать математическую теорию путем последовательности доказательств простых теорем и выводить такие важные результаты, как законы де Моргана. Перед тем как перейти к заключительной части этой главы, попытаемся переписать доказательства задач, разобранных в примерах 4.2—4.5, формальным образом, как это делалось в примере 4.1. В частности, каждый шаг должен быть проверен ссылкой на доказанный результат из предыдущих работ или же непосредственно выведен. Позднее мы введем необходимую терминологию, которая позволит использовать более краткие обозначения. Дадим сначала два определения.

Определение. Говорят, что два множества $A$ и $B$ *неэквивалентны*, если они не эквивалентны. Это свойство равносильно тому, что одно из множеств $A\backslash B$ или $B\backslash A$ непусто.

Определение. Множество всех подмножеств данного множества $X$ назовем *степенью множества* $X$ и будем обозначать через $\mathscr{P}(A)$. (Некоторые авторы используют обозначение $2^X$; причина этого будет ясна немного позднее, когда мы разберем несколько примеров.) Формально

$$\mathscr{P}(X)=\{Y\colon\ Y\subseteq X\}.$$

В частности, заметим, что поскольку $\varnothing\subseteq X$ и $X\subseteq X$, то

$$\varnothing\in\mathscr{P}(X),\quad X\in\mathscr{P}(X).\ //$$

Пример 4.6. Пусть $A=\{1,2,3\}$. Тогда

$$\mathscr{P}(A)=\{\varnothing,\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},A\}.\ //$$

Завершим этот параграф упоминанием о двух косвенных методах доказательства. Первый из них — доказательство от противного. Вспомним парадокс Рассела (пример 1.1):

$$F\in F\Rightarrow F\notin F\quad\text{и}\quad F\notin F\Rightarrow F\in F.$$

Если обозначить утверждение $F\in F$ через $P$, то получим $P$ справедливо $\Rightarrow P$ ложно и $P$ ложно $\Rightarrow P$ справедливо. Основой математики является предположение о том, что не может быть утверждения, которое является истинным и ложным одновременно (т. е. логическая система должна быть содержательной; мы отвергаем множество Рас-

села потому, что его определение несостоятельно в рассматриваемом смысле), и мы используем это положение как основу доказательства от противного. Предположим, что мы имеем совокупность высказываний $P_1$, $P_2$, ..., $P_n$ и хотим доказать ($P_1$ истинно и $P_2$ истинно и... и $P_n$ истинно) $\Rightarrow$ $Q$ истинно, или же более просто

$$(P_1 \text{ и } P_2 \text{ и } \ldots \text{ и } P_n) \Rightarrow Q.$$

Если мы допустим высказывание ($P_1$ и $P_2$ и ... и $P_n$ и не $Q$), т. е. что $P_1$, ..., $P_n$ истинно, а $Q$ ложно, и отсюда сможем вывести некоторое утверждение $P$, которое одновременно является и истинным. и ложным, то логическая система, основанная на

$$(P_1 \text{ и } P_2 \text{ и } \ldots \text{ и } P_n \text{ и не } Q),$$

является недопустимой. Таким образом, доказано, что если имеет место

$$(P_1 \text{ и } P_2 \text{ и } \ldots \text{ и } P_n),$$

то мы можем заключить, что

$$(P_1 \text{ и } P_2 \text{ и } \ldots \text{ и } P_n) \Rightarrow Q,$$

так как предположение не $Q$ приводит к противоречию. Для иллюстрации вышесказанного, рассмотрим пример.

Пример 4.7. Докажем, что для произвольных множеств $A$ и $B$ имеет место соотношение

$$A \subseteq B \Leftrightarrow B' \subseteq A'.$$

Доказательство. Допустим, что свойства (т. е. определения) множеств $\mathscr{E}$, $\varnothing$ и т. д. выполнены и что $A \subseteq B$ и $B' \not\subseteq A$. (В терминах описанной выше общей ситуации $Q$ это $B' \subseteq A'$.) Тогда

$$A \subseteq B \Rightarrow \text{если } x \in A, \text{ то } x \in B; \qquad (*)$$

$B' \not\subseteq A' \Rightarrow$ существует некоторый элемент $y$ такой, что

$$y \in B' \text{ и } y \notin A'.$$

Из (*) следует соотношение

$$y \in A \Rightarrow y \in B \Rightarrow y \in B' \text{ и } y \in B \Rightarrow y \in B' \cap B = \varnothing$$

(противоречие).

Следовательно, полученное утверждение $B' \not\subseteq A'$ ложно, и поэтому $B' \subseteq A'$. Аналогично можно показать, что $B' \subseteq A' \Rightarrow A \subseteq B$, и, следовательно,

$$A \subseteq B \Leftrightarrow B' \subseteq A'. \; //$$

Этот пример также распространяется и на второй метод косвенного доказательства.

Пример 4.8. Пусть $P$ означает высказывание «сегодня четверг», а $Q$ — «сегодня день недели». Тогда $(P \Rightarrow Q)$ означает «если сегодня четверг, то это день недели», а (не $Q \Rightarrow$ не $P$) означает «если сегодня не день недели, то это не четверг».

Следует убедиться, что эти два высказывания эквивалентны (т. е. что они оба одновременно являются либо истинными, либо ложными). //

С этой точки зрения, хотя диаграммы Венна могут использоваться для прояснения ситуации, все результаты должны быть выведены из предположений, данных в задаче. Следует помнить, что если какое-либо утверждение истинно, то мы должны быть в состоянии доказать его. Если это действительно очевидно, то мы легко докажем его, если же нет, то, по-видимому, это не так очевидно, как мы думали, и вполне возможно, что это даже неверно.

Упражнение 1.4.

1. Доказать, что

$$A \cap (B \cap C) = (A \cap B) \cap C.$$

2. Пусть даны множества $A$, $B$ и $C$: $C \subseteq B$. Доказать, что:

а) $A \cap C \subseteq A \cap B$; б) $A \cup C \subseteq A \cup B$;

в) $A \setminus B \subseteq A \setminus C$; г) $C \setminus A \subseteq B \setminus A$;

д) $B' \setminus A \subseteq C' \setminus A$.

3. Доказать, что если $A \subseteq B$, то $\mathscr{P}(A) \subseteq \mathscr{P}(B)$.

4. Показать справедливость равенства

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C).$$

5. Доказать, что $(A \cup B)' = A' \cap B'$. (Указание: показать, что $(A \cup B) \cup (A' \cap B') = \mathscr{E}$ и $(A \cup B) \cap (A' \cap B') = \varnothing$.)

6. Доказать эквивалентность следующих утверждений, т. е. что из каждого следует другое:

а) $A \cup B = \mathscr{E}$; б) $A' \subseteq B$; в) $A' \cap B' = \varnothing$.

7. Какие из следующих утверждений справедливы:

а) $0 \in \varnothing$; б) $\{\varnothing\} \subseteq \varnothing$; в) $\varnothing \subseteq \{\varnothing\}$;

г) $\varnothing \subseteq \mathscr{E}$; д) $\{\varnothing\} \subseteq \{\{\varnothing\}\}$?

Сравните ответы на этот вопрос с ответами к упражнению 1.2,6. Существует связь между символами $\in$ и $\subseteq$ однако это не одно и то же. Как аналогия «портфеля» связана с символом $\subseteq$?

8. Показать, что для конечного множества $A$

$$|2^A| = 2^{|A|}.$$

(Указание: выписать множество $A = \{a_1, \ldots, a_n\}$ и рассмотреть его подмножества.)

## § 5. Произведения множеств

Пока мы в основном занимались построением из существующих множеств множеств меньшего размера. Сейчас будет рассмотрен один из наиболее общих способов конструирования больших множеств. Рассмотрим для иллюстрации множество размеченных клеток шахматной доски (рис. 1.10). Рассмотрим множество столбцов, которые обозначим буквами $a$, $b$, ..., $h$ (слева направо), и

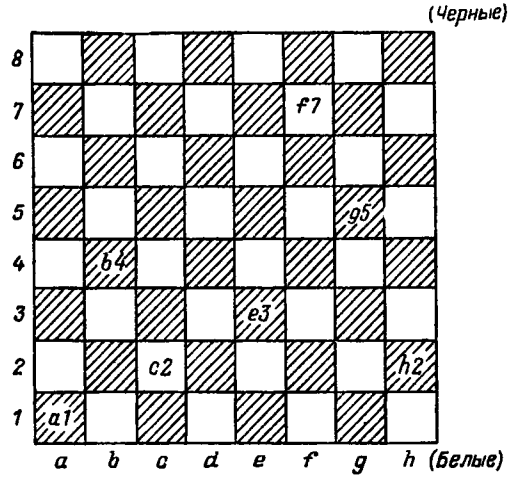



Рис. 1.10

множество строк от 1 до 8 (снизу вверх). Следовательно, каждая клетка может быть однозначно задана двумя символами: один — из множества $F = \{a, b, \ldots, h\}$, другой — из множества $R = \{1, 2, \ldots, 8\}$, например $a1$, $f7$, $e3$ и т. д. Таким образом, из множества столбцов $F$ и множества строк $R$ мы образовали множество всех клеток доски.

Этот пример содержит в себе новые идеи, которые используются при построении произведений множеств. Однако для того, чтобы быть в состоянии обобщить рассматриваемую ситуацию, следует быть немного более точными.

Определение. Обозначим последовательность из $n$ элементов $x_1, x_2, \ldots, x_n$ через $(x_1, x_2, \ldots, x_n)$. Здесь круглые скобки используются для того, чтобы указать на порядок, в котором записаны элементы. Например, если $x_1 \neq x_2$, то последовательность $(x_2, x_1, \ldots, x_n)$ не совпадает с исходной. Будем называть такую последовательность *набором длины n*; набор длины 2 будем называть *парой*.

Пусть даны $n$ множеств $A_1, A_2, \ldots, A_n$; множество всех наборов $(x_1, x_2, \ldots, x_n)$ таких, что $x_1 \in A_1, \ldots, x_n \in A_n$, называют *прямым произведением* $A_1, \ldots, A_n$ и обозначают $A_1 \times A_2 \times \ldots \times A_n$. Используя другие обозначения, это произведение запишем более кратко: $\prod_{i=1}^{n} A_i$. //

Пример 5.1. Пусть $X = \{0, 1\}$, $Y = \{x, y\}$, $Z = \{0, 1, 2\}$. Тогда

$$X \times Y = \{(0, x), (0, y), (1, x), (1, y)\},$$

$$Y \times X = \{(x, 0), (x, 1), (y, 0), (y, 1)\}.$$

Таким образом, $X \times Y \neq Y \times X$. При рассмотрении снова примера с шахматной доской становится ясно, что будет, если написать выражение $3e$. Множества $F$ и $R$ не пересекаются и $(e, 3) \in F \times R$, а $(3, e) \notin F \times R$. Например,

$$X \times Y = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2)\}$$

и $(0, 1)$ и $(1, 0)$ — различные элементы $X \times Y$. Следовательно, с математической точки зрения нам следует отвергнуть запись $3e$ как неверную для шахматной доски. //

Мы часто будем использовать прямое произведение для одинаковых множеств. В этом случае будет удобнее записывать $A \times A \times \ldots \times A$ как $A^n$.

Упражнение 1.5.

1. Пусть $X = \{a, b, c\}$ и $Y = \{a, b, e, f\}$. Найти $X \times Y$ и $Y^2$.

2. Доказать: при $A \subseteq X$ и $B \subseteq Y$, $A \times B \subseteq X \times Y$.

3. Доказать, что $A \times (B \cap C) = (A \times B) \cap (A \times C)$.

4. Доказать, что для любых непустых конечных множеств $A$ и $B$ выполняются соотношения:

а) $\varnothing \times A = \varnothing$; б) $\mathscr{B} \times A \neq A$; в) $A \subseteq A \times A$;

г) $|A \times \{x\}| = |A|$;

д) $A \times B = B \times A$ тогда и только тогда, когда $A = B$.

# ГЛАВА 2
# ОТНОШЕНИЯ

Часто в вычислениях необходимо выбирать элементы множеств, которые удовлетворяют некоторому «отношению». Это понятие довольно общее. Поэтому оно широко применимо. Естественно, что при соответствующем выборе отношения его аргументы могут быть связаны достаточно просто. Они не обязательно должны быть связаны какой-либо простой или очевидной формулой, хотя в ситуациях, когда требуется осуществить некоторые вычисления, иногда можно найти удачное описание отношения.

Перед тем как подойти к этому вопросу с математической позиции, рассмотрим несколько идей, возникающих из рассмотрения следующей простой ситуации (которая также приводит к возникновению понятий отношения). Предположим, что для некоторой конечной машины мы имеем множество программ $P$, конечное множество значений данных $D$ и множество результатов $R$. Если мы выберем конкретное значение из $D$, то оно может использоваться в некоторых программах из $P$, и для каждой программы из $P$ существует совокупность значений из $D$, которые в ней используются. Таким образом, мы имеем соответствие между значениями данных и программами, и, следовательно, существуют элементы в $D \times P$, представляющие интерес. Аналогично, если мы сведем рассмотрение к $p \in P$, то $p$ связывает соответствующие значения данных из $D$ с результатами из $R$. Можно рассмотреть данные, приводящие $p$ к остановке, или результаты, которые не могут быть получены из $p$. Следовательно, мы приходим к подмножеству $D \times R$. (При переработке данных от $D$ к $R$ возникают некоторые ассоциации, которые могут оказаться полезными для запоминания терминологии.)

Перейдем теперь к формальным рассмотрениям.

## § 1. Основные понятия

*n-местным отношением* $R$ на множествах $A_1, \ldots, A_n$ называется подмножество прямого произведения

$$A_1 \times \ldots \times A_n.$$

Другими словами, элементы $x_1, \ldots, x_n$ (где $x_1 \in A_1, \ldots$) связаны отношением $R$ тогда и только тогда, когда $(x_1, x_2, \ldots, x_n) \in R$ ($(x_1, x_2, \ldots, x_n)$ — упорядоченный набор из $n$ элементов).

Наиболее часто встречаются отношения при $n = 2$; в этом случае они называются *бинарными* отношениями. Следовательно, бинарное отношение между множествами $A$ и $B$ является просто подмножеством $A \times B$. Если эти множества эквивалентны (скажем, равны $A$), то будем говорить, что подмножество $A^2$ определяет отношение на $A$.

Отношения не являются чем-то новым. Можно построить отношения, которые, несомненно, будут знакомы читателю. Рассмотрим следующие примеры.

Пример 1.1. Пусть

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}.$$

Тогда $R = \{(x, y)\colon x, y \in A, x$ — делитель $y$ и $x \leqslant 5\}$ может быть записано в явном виде:

$$\begin{aligned} R = \{&(1, 1), (1, 2), (1, 3), (1, 4), (1, 5),\\ &(1, 6), (1, 7), (1, 8), (1, 9), (1, 10),\\ &(2, 2), (2, 4), (2, 6), (2, 8), (2, 10),\\ &(3, 3), (3, 6), (3, 9),\\ &(4, 4), (4, 8),\\ &(5, 5), (5, 10)\}. \ //\end{aligned}$$

Пример 1.2 (шахматы). Как и выше, пусть

$$F = \{a, b, c, d, e, f, g, h\}, \quad R = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

и пусть $S = F \times R$.

Таким образом, $S$ — множество всех клеток, обозначаемых парами $(x, y)$, где $x \in F$, $y \in R$. Определим бинарное отношение $C$ (для ладьи!) на $S$ так, что $(s, t) \in C$ тогда и только тогда, когда $s$ и $t$ — элементы $S$ и ладья может пройти от $s$ к $t$ одним ходом на пустой доске. (Напомним, что ладья может изменять либо горизонтальную координату, либо вертикальную, но не обе координаты

одновременно.) Поэтому $C \subseteq S \times S$ и

$$C = \{((f_s, r_s), (f_t, r_t)): (f_s = f_t \text{ и } r_s \neq r_t) \text{ или } (f_s \neq f_t \text{ и } r_s = r_t)\}. \ //$$

На первый взгляд определение $C$ может выглядеть сложно, но внимательное исследование показывает, что значение отношения находится непосредственно и вся необходимая информация содержится в определении.

В общем случае ряд различных отношений на множестве $A$ зависит от $|A|$. Бóльшая часть этих отношений не представляет особого интереса. Ниже приведены три отношения, которые полезны при рассмотрении множеств.

Определение. Для любого множества $A$ определим *тождественное* отношение $I_A$ и *универсальное* отношение $U_A$ следующим образом:

$$I = \{(a, a): a \in A\}, \quad U = \{(a, b): a \in A, b \in A\}.$$

Таким образом, $U_A = A^2$. Так как $\varnothing \subseteq A^2$, то $\varnothing$ является отношением на $A$ и называется *пустым* отношением. //

Пусть отношение $R$ определено в соответствии с изображением на рис. 2.1. Необходимо сконцентрировать наше внимание на том, что́ происходит на концах $R$, т. е.

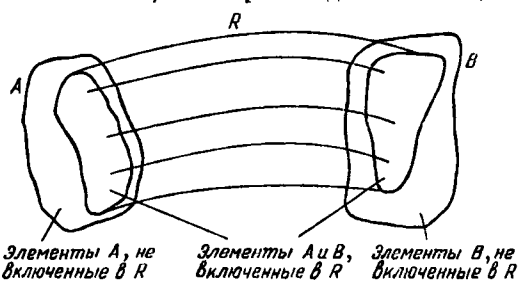



Рис. 2.1

рассмотреть элементы $A$ и $B$, которые принадлежат $R$. Они являются элементами подмножеств $A$ и $B$ соответственно и, как следовало ожидать, имеют специальные названия.

Определение. Свяжем с каждым бинарным отношением $R$ между $A$ и $B$ два множества — *область определения* $D(R)$ и *область значений* $\mathcal{R}(R)$. Они определяются следующим образом:

$$D(R) = \{x: (x, y) \in R\}, \quad \mathcal{R}(R) = \{y: (x, y) \in R\}. \ //$$

Пример 1.3. Пусть отношение $R$ такое же, как и в примере 1.1. Тогда

$$\mathscr{D}(R) = \{1, 2, 3, 4, 5\},\ \mathscr{R}(R) = A.\ //$$

Пример 1.4. Предположим, что мы имеем некоторую программу. Она читает два числа из множества $A = \{1, 2, 3, 4, 5\}$, обозначаемых $x$ и $y$, и, если $x < y$, печатает число $z$ (также из $A$) такое, что $x \leqslant z < y$. В любом случае программа останавливается после считывания всех чисел из $A$.

Задача определяет отношение $P$, $P \subseteq A \times A$ такое, что

$$P = \{((x, y), z):\ x < y,\ x \leqslant z < y\}.$$

Не все входные данные приводят к выдаче результата. Поэтому область определения $P$ не совпадает с $A^2$. Ясно, что

$$\begin{aligned} P = \{&((1,2), 1), ((1,3), 1), ((1,3), 2),\\ &((1,4), 1), ((1,4), 2), ((1,4), 3),\\ &((1,5), 1), ((1,5), 2), ((1,5), 3), ((1,5), 4),\\ &((2,3), 2), ((2,4), 2), ((2,4), 3),\\ &((2,5), 2), ((2,5), 3), ((2,5), 4),\\ &((3,4), 3), ((3,5), 3), ((3,5), 4),\\ &((4,5), 4)\}; \end{aligned}$$

$$\begin{aligned} \mathscr{D}(P) = \{&(1,2), (1,3), (1,4), (1,5),\\ &(2,3), (2,4), (2,5),\\ &(3,4), (3,5),\\ &(4,5)\}; \end{aligned}$$

$$\mathscr{R}(P) = \{1, 2, 3, 4\}.\ //$$

Хотя каждое отношение является множеством и может быть обозначено прописной буквой, существует также практика обозначения отношений строчными греческими буквами, например $\rho$, $\sigma$ и $\tau$. Часто используют следующие обозначения:

а) $(a, b) \in \rho$, т. е. $(a, b)$ находится в $\rho$;

б) $a \rho b$, $a$ связано с $b$ отношением $\rho$;

в) $b \in \rho(a)$.

Первое из обозначений является естественным (оно следует из определений теории множеств). Второе делается более разумным, если мы рассмотрим отношение порядка (которое будет определено несколько ниже), где

$$R \subseteq N^2 \text{ и } R = \{(x, y):\ x < y\}.$$

Здесь вместо $6R7$ мы можем написать $6 < 7$; следовательно, мы разрешаем запись любого отношения с использованием рассмотренных выше символов, если результирующая последовательность символов будет однозначно определенной. Третья форма записи является новой и будет преобразована к более привычным обозначениям в гл. 3. Из заданного бинарного отношения можно вывести ряд других отношений, большинство из которых будут обратными отношениями.

Определение. Пусть $R$ — бинарное отношение. Определим обратное отношение $R^{-1}$ следующим образом:

$$R^{-1} = \{(x, y):\ (y, x) \in R\}.$$

Таким образом, $R^{-1}$ связывает те же пары элементов, что и $R$, но «в другом порядке». //

Следовательно, если $R \subseteq A \times B$, то

$$R^{-1} \subseteq B \times A,\ \mathscr{D}(R^{-1}) = \mathscr{R}(R) \text{ и } \mathscr{R}(R^{-1}) = \mathscr{D}(R).$$

Чтобы избежать использования большого количества скобок, будем также использовать обозначения $\mathscr{D}_R$ и $\mathscr{R}_R$ вместо $\mathscr{D}(R)$ и $\mathscr{R}(R)$ соответственно.

Упражнение 2.1.

1. Пусть $X = \{2, 4, 6, 8\}$ и $\rho = \{(x, y):\ x, y \in X$ и $x < y\}$. Выписать все элементы $\rho$ и $\rho^{-1}$.

2. Пусть $\mathscr{E} = \mathscr{P}(\{a, b, c\})$. Найти все элементы отношений $\subset$ и $\subseteq$ на $\mathscr{E}$.

3. Пусть $\mathscr{E} = Z^2$ и $\rho = \{(x, y):\ x < y\}$. Описать $\rho'$ — дополнение $\rho$ — без использования отрицания отношения «меньше».

4. Пусть $\sigma = \{(x, y):\ x \subset y\}$. Может ли $\sigma'$ быть описано тем же способом, что и $\rho'$ в 2.1, 3? Ответ проверьте.

5. На улице есть 30 домов, пронумерованных обычным способом: нечетные номера с одной стороны, а четные с другой. Пусть $h_n$ обозначает жителя, живущего в доме с номером $n$. Описать при помощи символов отношение $N$ на множестве жителей такое, что $h_i$ находится в отношении $N$ к $h_j$, если они являются соседями.

Как будет выглядеть $N$, если улица является тупиком?

6. Вернемся к множеству $S$ клеток шахматной доски. Отношение $K$ связывает клетки, которые определяются ходом коня (т. е. если конь может перейти с $x$ на $y$ за один шаг). Определить $K$ при помощи символов.

7. Пусть $G$ — отношение на $S$ такое, что $xGy$ тогда и только тогда, когда $x$ есть начальная позиция (белой) пешки, а $y$ есть клетка, где первый ход игры заканчивается. Описать $G$, $\mathscr{D}(G)$ и $\mathscr{R}(G)$.

## § 2. Графические представления

При решении задачи на первом этапе часто полезно начертить «рисунок» для того, чтобы более ясно увидеть компоненты задачи. Особенно это полезно для описания отношений, так как записанные в виде множества упорядоченных пар отношения нелегко расшифровываются.

Отношения — это множества, обладающие определенной структурой; их элементы имеют несколько компонент, и поэтому, в принципе, мы можем использовать диаграммы Венна для их изображения. Хотя этим методом и можно воспользоваться, особенно при описании некоторых больших множеств чисел, существуют методы, которые более эффективны в общих ситуациях (включающих, в частности, бинарные отношения на небольших множествах). В этом параграфе мы кратко рассмотрим некоторые из них. Для описания этих методов используем множество

$$X = \{a, b, c, d\}$$

и отношения $I_X$, $U_X$ и $R$, где

$$R = \{(a, b), (a, c), (b, d), (c, e), (e, b)\}.$$

Вначале рассмотрим метод, относящийся к традиционной аналитической геометрии. Начертим пару взаимно перпендикулярных осей ($OX$ — горизонтальная ось, а $OY$ — вертикальная ось) и на каждой отметим точки, представляющие элементы множества $X$ (рис. 2.2, *а*). Теперь в правом верхнем координатном углу отметим точки с координатами $(x, y)$, у которых $x \in X$ и $y \in Y$. Множества, соответствующие $I_X$, $U_X$ и $R$, изображены на рис. 2.2, *b*, *c* и *d*.

Основной недостаток этого метода заключается в том, что при увеличении $|X|$ трудно увидеть элементы в области и установить соответствие с точками, обозначающими отношения. Чтобы преодолеть этот недостаток, можно опустить точки и соединить стрелкой $x \in \mathscr{D}$ и $y \in \mathscr{R}$, когда $(x, y)$ принадлежит отношению (рис. 2.3). Диаграмма, представляющая $U_X$, получилась довольно запутанной, но это естественно, поскольку число элементов в

$U_x$ увеличилось. С другой стороны, отношения $I_x$ и $R$ представлены наглядно, и легко увидеть их области определения и значений. Диаграмма для $U_x$ наиболее неудобна в месте пересечения осей. Теперь, когда не используются координаты в областях определения и значений для

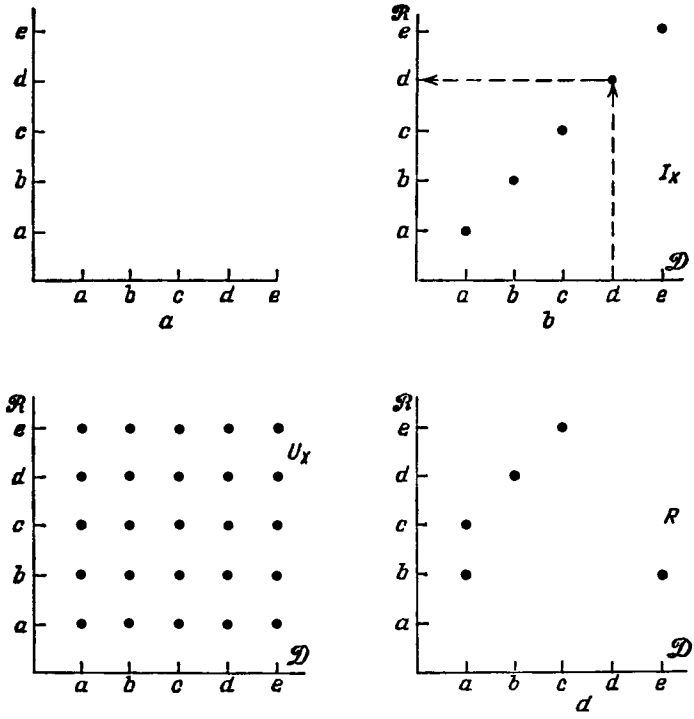



Рис. 2.2

расстановки элементов, отношения (как в первом методе) можно начертить параллельными. Поэтому, используя параллельные вертикальные линии и двигаясь слева направо (линия слева является областью определения), мы получаем диаграмму, изображенную на рис. 2.4. Здесь стрелки не требуются, так как мы знаем, что отношения идут от области определения к области значений. Это приводит к двум возможностям: мы можем или заменить стрелки прямыми линиями, или заменить две линии, изображающие области определения и значений, простой совокупностью точек. (Например, точка *c* в об-

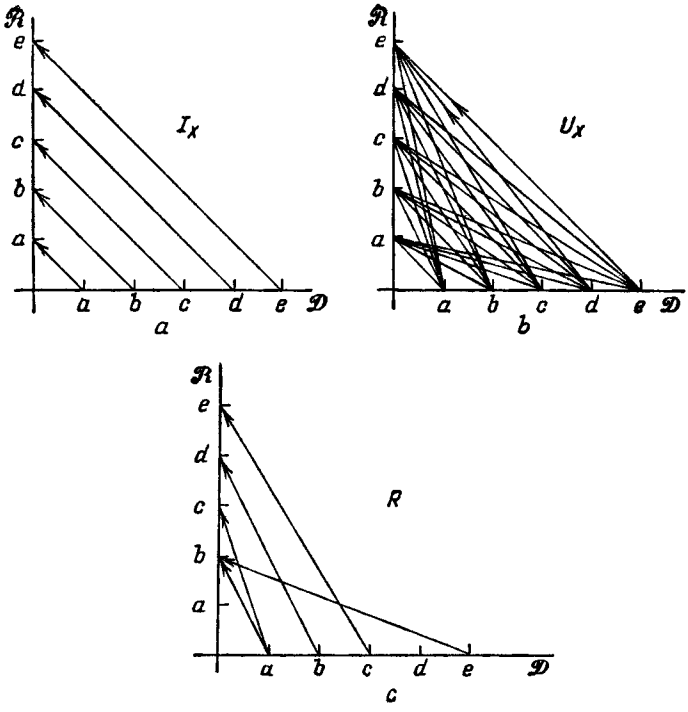

$\mathcal{R}$
e
d
c
b
a
$I_X$
a b c d e $\mathcal{D}$
a
$U_X$
b
R
c

Рис. 2.3

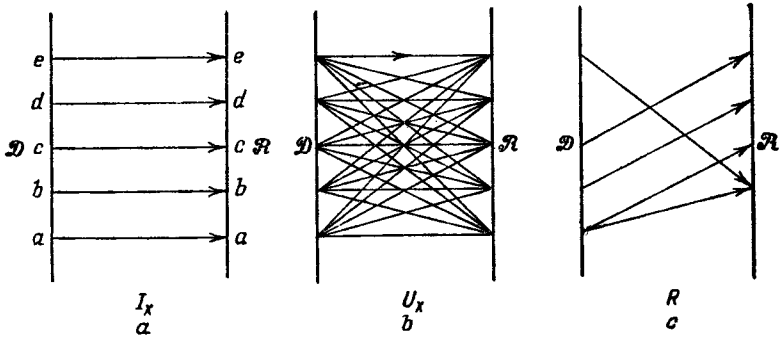

e
d
$\mathcal{D}$ c
b
a
$\mathcal{R}$
$I_X$
a
$U_X$
b
R
c

Рис. 2.4

ласти определения является той же самой, что и точка, представляющая $c$ в области значений.) Это показано на диаграмме, изображенной на рис. 2.5.

Итак, обозначены наиболее важные методы графического изображения бинарных отношений. Они будут ис-

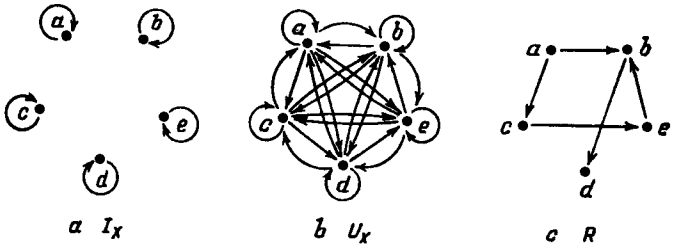



Рис. 2.5

пользоваться в оставшейся части книги. Обсуждение графических методов, связанных с соотношениями, мы продолжим в гл. 7.

Упражнение 2.2.

1. Начертить диаграмму, представляющую отношение $\rho$ из упражнения 2.1, 1.

2. Начертить диаграмму, представляющую отношение $N$ (см. упражнение 2.1, 5) на улице, имеющей десять домов. Как изменится диаграмма, если улица является тупиком?

## § 3. Свойства отношений

Очевидно, что общие отношения, будучи только подмножествами произведения множеств, не особенно интересны, поскольку о них можно сказать очень мало. Однако, когда отношения удовлетворяют некоторым дополнительным условиям, относительно них можно сделать более содержательные утверждения. В этом параграфе мы рассмотрим некоторые из основных свойств, которыми могут быть наделены отношения. Говорят, что свойство имеет место, если только выполнено соответствующее условие.

Определение. Пусть $\rho$ — отношение на множестве $A$. Тогда:

а) $\rho$ *рефлексивно*, если $x\rho x$ для любого $x \in A$;

б) $\rho$ *симметрично*, если $x\rho y$ влечет $y\rho x$;

в) $\rho$ *транзитивно*, если $x\rho y$ и $y\rho z$ влечет $x\rho z$;

г) $\rho$ *антисимметрично*, если $x\rho y$ и $y\rho x$ влекут $x = y$.

Терминология, введенная здесь, вероятно, является новой для читателя, однако он, по-видимому, знаком с этими понятиями. Поясним их на следующих примерах.

Пример 3.1. Пусть

$\rho = \{(x, y)$: $x, y \in \mathbf{N}$ и $x$ — делитель $y\}$,

$\sigma = \{(x, y)$: $x, y \in \mathbf{N}$ и $x \leqslant y\}$,

$\tau = \{(x, y)$: $x, y \in \mathbf{N} \backslash \{1\}$ и $x$ и $y$ имеют общий делитель$\}$.

Тогда $\rho$:

а) рефлексивно, так как $x/x = 1$ для всех $x \in \mathbf{N}$;

б) несимметрично, поскольку 2 — делитель 4, но 4 не является делителем 2;

в) транзитивно, так как если $y/x \in \mathbf{N}$ и $z/y \in \mathbf{N}$, то

$$z/x = (y/x) * (z/y) \in \mathbf{N};$$

г) антисимметрично, так как если $x/y \in \mathbf{N}$ и $y/x \in \mathbf{N}$, то $x = y$.

Аналогично $\sigma$:

а) рефлексивно, так как $x \leqslant x$ для всех $x \in \mathbf{N}$;

б) несимметрично, так как $2 \leqslant 3$, но $3 \not\leqslant 2$;

в) транзитивно;

г) антисимметрично, так как если $x \leqslant y$ и $y \leqslant x$, то $x = y$.

Наконец, $\tau$ рефлексивно и симметрично, но не транзитивно или антисимметрично. //

Пример 3.2. Пусть $P$ — множество всех людей, а $A$ и $S$ определяются следующим образом:

$$A = \{(x, y)\text{: } x, y \in P \text{ и } x \text{ — предок } y\},$$

$S = \{(x, y) \in P$ и $x$ и $y$ имеют одних и тех же родителей$\}$.

Очевидно, что $A$ транзитивно, а $S$ рефлексивно, симметрично и транзитивно. //

Заметим, что свойства симметричности и антисимметричности не являются взаимоисключающими. Для любого множества $X$ отношение $I_X$ является симметричным и антисимметричным. (Проверьте!) Мы можем также иметь отношения, которые не являются ни симметричными, ни антисимметричными.

Пример 3.3. Пусть опять $P$ есть множество всех людей. Определим отношение $B$ такое, что $xBy$ тогда и только тогда, когда $x$ является братом $y$. В семье, состоящей из двух братьев $p$ и $q$ и сестры $r$, мы имеем ситуацию, изображенную на рис. 2.6. Отношение $B$ не симметрично, так как $pBr$, но не $rBp$. Это отношение также

не является антисимметричным, так как $pBq$ и $qBp$, хотя $p$ и $q$ различны.

В более общей ситуации мы можем интерпретировать рассмотренные выше характеристики отношения путем построения диаграммы:

а) отношение рефлексивно тогда и только тогда, когда для каждого узла (точки) на диаграмме существует стрелка, которая начинается и заканчивается на этом узле;

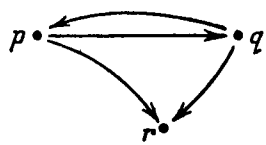


Рис. 2.6

б) отношение симметрично тогда и только тогда, когда для каждой стрелки, соединяющей два узла, существует также стрелка, соединяющая эти узлы в обратном направлении;

в) отношение транзитивно тогда и только тогда, когда для каждой пары узлов $x$ и $y$, связанных последовательностью стрелок от $x$ к $a_1$, от $a_1$ к $a_2$, ..., от $a_{n-1}$ к $a_n$, от $a_n$ к $y$, существует также стрелка от $x$ к $y$;

г) отношение антисимметрично тогда и только тогда, когда не существует двух различных узлов, связанных парой стрелок.

Существует много других свойств отношений, которые можно было бы рассмотреть. Однако рассмотренные выше свойства являются наиболее важными и будут часто использоваться в дальнейшем.

Упражнение 2.3.

1. Являются ли следующие отношения рефлексивными, симметричными, транзитивными или анситимметричными:

а) отношение на {1, 2, 3, 4, 5} определяется как

$$\{(a, b): a - b \text{ четное}\};$$

б) отношение на {1, 2, 3, 4, 5} определяется как

$$\{(a, b): a + b \text{ четное}\};$$

в) отношение на $P$ (множестве всех людей) определяется как

$$\{(a, b): a \text{ и } b \text{ имеют общего предка}\}?$$

2. Следующее утверждение ошибочно. Симметричное и транзитивное отношение на $S$ является также рефлексивным, так как $aRb$ и $bRa$ влекут $aRa$. Внимательно

изучив определения, найти ошибку. Построить отношение на {1, 2, 3}, которое является симметричным и транзитивным, но не рефлексивным.

3. Пусть $\rho$ — отношение между $A$ и $B$, $a \in A$. Тогда $\rho(a)$ определено как множество $\{b\colon a\rho b\}$ и является подмножеством $B$. Пусть на $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ определены следующие отношения:

$$\rho = \{(a, b)\colon a < b\},$$
$$\sigma = \{(a, b)\colon b - 1 < a < b + 2\},$$
$$\tau = \{(a, b)\colon a^2 \leqslant b\}.$$

Какие это множества:

а) $\rho(0)$; б) $\sigma(0)$; в) $\tau(0)$;
г) $\rho(1)$; д) $\sigma(-1)$; е) $\tau(-1)$?

## § 4. Разбиения и отношения эквивалентности

Во многих вычислительных задачах берутся большие множества и разбиваются таким образом, чтобы все интересующие нас ситуации можно было исследовать на нескольких правильно выбранных примерах. Например, один из путей получения качественной оценки характеристик языка программирования — это посмотреть конкретные программы, написанные на этом языке. Однако каждый интересный язык, включая такие языки высокого уровня, как Паскаль и Фортран, порождает бесконечно много программ, и, следовательно, мы должны выбирать программы так, чтобы они правильно отражали достоинства и недостатки языка. Чтобы быть более конкретными, давайте в дальнейшем предполагать, что язык имеет три основные управляющие структуры и четыре метода доступа и более у него нет никаких особых свойств. Мы могли бы в качестве примера взять семь программ, каждая из которых включает только одну характеристику языка (хотя, вообще говоря, каждая программа может использовать более чем одну характеристику языка). Исследование этих програм тогда могло бы покрыть бо́льшую часть свойств языка. Математически это можно определить следующим образом.

Определение. Пусть $A$ — непустое множество и $\{A_i\}$ — совокупность подмножеств ($i = 1, \ldots, n$, $n \in \mathbf{N}$) таких, что

$$\bigcup_{i=1}^{n} A_i = A.$$

Совокупность этих подмножеств называется *покрытием* $A$. //

Пример 4.1.

$\{A, B\}$ — покрытие $A \cup B$,

$\{A, A \cup B, B, C\}$ — покрытие $A \cup B \cup C$. //

Используя понятие покрытия, можно обеспечить, чтобы ни одно из свойств не было пропущено, так как каждый элемент включен по крайней мере в одно из подмножеств покрытия. Однако в общем случае могут встречаться случаи дублирования. Если в дальнейшем потребовать, чтобы элементы покрытия попарно не пересекались, то дублирования не будет. Отсюда возникает понятие разбиения.

Определение. *Разбиением* непустого множества $A$ называется совокупность подмножеств $\mathscr{P}(A)$ таких, что объединение всех элементов $\mathscr{P}(A)$ совпадает с $A$ и все элементы $\mathscr{P}(A)$ взаимно не пересекаются, т. е. $A$ разбито таким образом, что каждый элемент $A$ содержится только в одном подмножестве разбиения. //

Пример 4.2.

$\{A, A'\}$ — разбиение $\mathscr{E}$,

$\{A \cap B, A \cap B', A' \cap B, A' \cap B'\}$ — разбиение $\mathscr{E}$,

$\{A \backslash B, A \cap B, B \backslash A\}$ — разбиение $A \cup B$. //

Разбиение определяется однозначно, и части разбиения индуцируют особый род отношения, называемого отношением эквивалентности. Эти отношения ведут себя подобно отношению «=» между числами или множествами. Выделяя основные свойства равенства, мы приходим к следующему определению.

Определение. Бинарное отношение на множестве называют *отношением эквивалентности*, если оно рефлексивно, симметрично и транзитивно. //

Пример 4.3. На множестве всех треугольников отношение, определяемое как $\{(x, y)$: $x$ и $y$ имеют одинаковую площадь$\}$, является тривиальным отношением эквивалентности. Более интересно следующее отношение, определенное на множестве всех программ: $\{(a, b)$: $a$ и $b$ вычисляют одну и ту же функцию на определенной машине$\}$. Это отношение является отношением эквивалентности. //

Напомним, что мы рассматриваем более простые способы создания больших множеств, разбивая их на мел-

кие части, чем если бы в качестве этих частей брали элементы множества. В настоящий момент у нас уже имеется математический аппарат, однако недостает подходящих простых обозначений. Сейчас это будет сделано, после чего приведем несколько известных примеров.

Определение. Пусть $\rho$ — отношение эквивалентности на множестве $A$. Определим *класс эквивалентности* $[x]$ для $x \in A$:

$$[x] = \{y\colon x \rho y\}.$$

Таким образом, $[x]$ есть множество всех элементов $A$, которые $\rho$-эквивалентны $x$. В случаях, когда рассматривается только одно отношение эквивалентности, мы можем также использовать обозначение «$\equiv$» (эквивалентно), поэтому

$$[x] = \{y\colon x \equiv y\}.$$

В отдельных специальных случаях для обозначения эквивалентности иногда используют символ «$\sim$». Теперь вместо проверки всего множества мы можем любым способом выбрать представителей (по одному от каждого из классов эквивалентности), что упрощает вычисления.

Следующий пример иллюстрирует вышесказанное.

Пример 4.4. Пусть $s$ — фиксированный элемент N; определим отношение $\rho_s$ на Z:

$$\rho_s = \{(x, y)\colon x - y = ns, \text{ где } n \in \mathrm{Z}\}.$$

Рассмотрим случай $s = 10$. Тогда

$$[1] = \{11, 21, -9, 10\,976\,631, \ldots\},$$
$$[1066] = \{66, 226, -24, \ldots\}$$

и т. д.

В действительности существуют только десять различных классов эквивалентности. Целые 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 принадлежат различным классам. Поэтому мы можем использовать их в качестве представителей этих классов. //

В вычислениях отношения эквивалентности представляют особый интерес, поскольку они ассоциируются с различными алгоритмами, дающими один и тот же результат, или приводящими к одной и той же обработке данных, или же представляющими одну и ту же информацию о различных эквивалентных структурах данных. Примером таких очевидных ситуаций является борьба с трудностями, которые возникают при исследовании раз-

решимости. Такие факторы, однако, не вызывают проблем, если множества образуются путем хорошо сконструированных процессов с конечной базой. И все же типичным является случай, когда при описании даже «хорошего поведения» требуется большое число деталей, в связи с чем они должны быть здесь рассмотрены. Несмотря на это, мы будем возвращаться к подобным вопросам в § 6 и в гл. 8 и 9.

По-видимому, наиболее известное отношение эквивалентности знакомо читателю, хотя он мог и не знать, что оно — отношение эквивалентности. Это отношение связано с дробями. Рассмотрим множество $\mathbf{Z}\times\mathbf{N}$. Пару $(a, b)$ мы можем рассматривать как дробь $a/b$. Эти два способа обозначения элементов $\mathbf{Z}\times\mathbf{N}$ являются различными, но они «изоморфны». Мы будем их рассматривать далее в § 1 гл. 5. Отметим, что можно переходить от одной формы обозначения к другой.

До сих пор все было хорошо, однако существуют различные элементы в $\mathbf{Z}\times\mathbf{N}$, которые желательно рассматривать как одни и те же, хотя записываются они поразному. Чтобы преодолеть эту трудность, определим отношение эквивалентности на $\mathbf{Z}\times\mathbf{N}$ следующим образом:

$(a, b)\equiv(c, d)$ тогда и только тогда, когда $a*d=b*c$.

Множество всех классов эквивалентности, определяемых этим отношением на $\mathbf{Z}\times\mathbf{N}$, называют *рациональными числами* и обозначают символом Q. Обычно выбирают тех представителей классов, у которых самые малые $a$ и $b$.

Следует упомянуть, что предполагается существование действительных чисел (множество действительных чисел обычно обозначают через R). Эти числа можно представить в форме

$$\ldots 0\ d_n \ldots d_2\ d_1\ d_0\ \delta_1\ \delta_2 \ldots \delta_m \ldots,$$

где каждое $d_i$ и $\delta_j$ принадлежат множеству

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} = \mathbf{Z}_{10}, \quad d_n \neq 0,$$

исключая случай $n=0$. В частности, допускается бесконечная непериодическая десятичная дробь, хотя нули перед $d_n$ обычно опускают. (Отрицательные числа представляют ненулевыми положительными числами со знаком минус.)

Заметим, что индекс отношения эквивалентности $\rho$ на множестве $A$ — это количество частей $A$, индуцируемых $\rho$ (число $\rho$-эквивалентных классов).

Упражнение 2.4.

1. Доказать, что любое отношение эквивалентности порождает такое разбиение, что для любых $x, y \in A$ или $[x] = [y]$, или $[x] \cap [y] = \varnothing$.

2. Пусть $A$ — конечное множество. Какие отношения эквивалентности дают наибольшее и наименьшее число эквивалентных классов?

3. Если $\{A_1, A_2, \ldots, A_n\}$ — разбиение $A$ и $A$ конечное, показать, что

$$|A| = \sum_{i=1}^{n} |A_i|.$$

## § 5. Отношения порядка

Поскольку из понятия равенства (скажем, между числами) возникает математическое понятие эквивалентности, некоторые неравенства могут также использоваться как модели для более широкого класса отношений.

Частичным порядком на множестве $A$ назовем отношение, которое рефлексивно, антисимметрично и транзитивно. *Порядок* (называемый также *отношением порядка*) — это обобщение отношения $\leqslant$ на **N**. Поэтому можно легко проверить требуемые три свойства. Заметим, что мы могли бы в качестве определения взять отношение $<$. Тогда отношение порядка было бы только транзитивно. Поэтому свойство транзитивности является наиболее важным для отношения порядка.

Определив отношение $\leqslant$, можно определить отношение $<$ следующим образом:

$$a < b \Leftrightarrow a \leqslant b \text{ и } a \neq b.$$

Аналогично, если задано $<$, то

$$a \leqslant b \Leftrightarrow a = b \text{ или } a < b.$$

Пример 5.1. Пусть задано произвольное множество $A$. Тогда отношение $\subseteq$ на $\mathcal{P}(A)$ есть тривиальное отношение порядка. ($X \subseteq X$ для всех $X$; если $X \subseteq Y$ и $Y \subseteq X$, то $X = Y$; $X \subseteq Y$ и $Y \subseteq Z \Rightarrow X \subseteq Z$.) //

Отношение порядка $\rho$ на $A$ называется *полным*, если для любых $x, y \in A$ или $x \rho y$, или $y \rho x$ (или же выполняются оба).

Пример 5.2. Очевидно, что порядок на подмножествах данного множества не является полным. Естественный порядок чисел на действительной оси **R** является полным.

По мере достижения прогресса в изучении математики становится ясным, что математика не набор разрозненных идей, а совокупность связанных между собой концептуальных понятий, которые используются во многих непохожих друг на друга ситуациях. Отсюда следует, что, если основной принцип установлен и исследован, он фактически единым образом решает проблемы для всех этих различных случаев. Следующий пример является простой иллюстрацией этой идеи.

Пример 5.3. На основе порядка, определенного на N, мы можем формально получить обычные отношения порядка на множествах чисел Z, Q и R. (Как уже упоминалось, исследование N будет проведено в § 3 гл. 3.)

Вначале рассмотрим Z. Чтобы облегчить рассуждения, разобьем Z следующим образом:

$$Z = N \cup \{0\} \cup A.$$

Поэтому $A = \{-x: x \in N\}$. Определим отношение (которое будем называть полным отношением порядка) на Z рассмотрением всевозможных элементов $x$ и $y$ из разбиения $\{N \cup \{0\} \cup A\}$.

Если $x = y$, то $x \leqslant y$ и $y \leqslant x$. Пусть $x \neq y$. Тогда:

а) если $x$, $y \in N$, то порядок в Z тот же самый, что и в N;

б) если $x$, $y \in A$, то

$x \leqslant y$ тогда и только тогда, когда $-y \leqslant -x$ в N

(т. е. $-5 \leqslant -4$, так как $4 \leqslant 5$);

в) если $x = 0$ и $y \in N$, то $x \leqslant y$;

г) если $x \in A$ и $y = 0$, то $x \leqslant y$;

д) если $x \in A$ и $y \in N$, то $x \leqslant y$ или в противном случае $y \leqslant x$.

На основе порядка на Z и обычных арифметических операций с целыми числами мы можем определить порядок на Q:

$a/b \leqslant c/d$ тогда и только тогда, когда $a * d \leqslant b * c$.

Проверку этого утверждения оставляем в качестве упражнения. Наконец, определим отношение порядка на множестве действительных чисел R. Рассмотрим десятичные представления двух действительных положительных чисел:

$$D = \ldots 0d_n \ldots d_2 d_1 d_0 \delta_1 \delta_2 \ldots,$$

$$C = \ldots 0c_m \ldots c_2 c_1 c_0 \gamma_1 \gamma_2 \ldots.$$

Если $d_i = c_i$ и $\delta_i = \gamma_i$ для всех $i$, то $D = C$ и, следовательно, $D \leqslant C$ и $C \leqslant D$. В противном случае:

а) если $d_n \neq 0$, $c_m \neq 0$ и $n \neq m$, то $D \leqslant C$, если $n < m$, и $C \leqslant D$, если $m < n$;

б) если $n = m$ и $d_i \neq c_i$, но $d_j = c_j$ для всех $j$ таких, что $i < j \leqslant n$, то из $d_i < c_i$ следует, что $D \leqslant C$, и, обратно, если $c_i < d_i$, то $C \leqslant D$;

в) если $m = n$ и $d_i = c_i$ для всех $i$, но $\delta_k \neq \gamma_k$ для некоторых $k$ и $\delta_j = \gamma_j$ для всех $j$ таких, что $0 < j < k$, тогда $C \leqslant D$, если $\gamma_k < \delta_k$, и $D \leqslant C$, если $\delta_k < \gamma_k$.

Читатель может проверить это самостоятельно. Отрицательные числа могут быть исследованы так же, как в **Z**. //

Множество $X$ вместе с отношением порядка $\leqslant$ называется *частично упорядоченным множеством* (обозначается $(X, \leqslant)$). Тогда любой элемент $u \in (X, \leqslant)$ такой, что $x \leqslant u$ и $y \leqslant u$, называется *верхней границей* $x$ и $y$. Аналогично, если $1 \in (X, \leqslant)$, $1 \leqslant x$ и $1 \leqslant y$, то 1 является *нижней границей* $x$ и $y$. Множество всех верхних границ $x$ и $y$ является подмножеством $X$ и упорядочено отношением $\leqslant$. Если существует единственный наименьший элемент этого множества, т. е. если существует $\mu \in (X, \leqslant)$ такое, что $x \leqslant \mu$, $y \leqslant \mu$ и $\mu \leqslant u$ для любой верхней границы $u$, то $\mu$ называют *верхней гранью* (sup) $x$ и $y$. Аналогично, если существует единственная наибольшая нижняя граница $x$ и $y$, ее называют *нижней гранью* (inf) $x$ и $y$. Мы отложим изучение верхней и нижней граней до § 5 гл. 5, где будут изучаться решетки.

Наконец, заметим, что использование естественного порядка на **R** определяет новые множества. Их называют *интервалами*:

$$[a, b] = \{x: x \in \mathbf{R}, a \leqslant x \leqslant b\}$$

есть *замкнутый интервал* (*отрезок*) от $a$ до $b$;

$$]a, b[ = \{x: x \in \mathbf{R}, a < x < b\}$$

есть *открытый интервал* от $a$ до $b$. В каждом случае числа $a$ и $b$ называются концевыми точками.

Замкнутый интервал включает в себя концевые точки, а открытый нет. Удобно также определить полуоткрытые интервалы:

$$[a, b[ = \{x: x \in \mathbf{R}, a \leqslant x < b\},$$
$$]a, b] = \{x: x \in \mathbf{R}, a < x \leqslant b\}.$$

Для удобства будем использовать следующие обозначения:

$$]-\infty, a] = \{x: x \leqslant a\},$$
$$]-\infty, a[ = \{x: x < a\},$$
$$[a, \infty[ = \{x: a \leqslant x\},$$
$$]a, \infty[ = \{x: a < x\},$$
$$]-\infty, \infty[ = \mathbf{R}.$$

Хотя интервалы и множества чисел в общем-то не являются центральной частью нашего рассмотрения, мы увидим, что их удобно использовать время от времени.

Упражнение 2.5.

1. Пусть $A$ — произвольное множество и $\rho$ — отношение на множестве $\mathscr{P}(A)\times\mathscr{P}(A)$, определенное следующим образом:

$(P, Q)\rho(X, Y)$ тогда и только тогда, когда

$$(P\Delta Q)\subseteq(X\Delta Y).$$

Является ли $\rho$ отношением порядка?

2. Пусть $A$ — произвольное множество и $\sigma$ — отношение на $\mathscr{P}(A)\times\mathscr{P}(A)$, определенное следующим образом:

$(P, Q)\sigma(X, Y)$ тогда и только тогда, когда

$$P\subseteq X \text{ и } Q\subseteq Y.$$

Является ли $\sigma$ отношением порядка? Если да, то является ли этот порядок полным?

3. Пусть $\tau$ и $\pi$ — отношения на $\mathbf{N}^2$, определяемые соотношениями:

$(a, b)\tau(c, d)$ тогда и только тогда, когда $a\leqslant c$ и $b\leqslant d$;

$(a, b)\pi(c, d)$ тогда и только тогда, когда $a\leqslant c$ и $b\geqslant d$.

Являются ли $\tau$ и $\pi$ отношениями порядка?

4. Пусть $\pi$ определено на положительных элементах $\mathbf{Q}$ следующим образом:

$(a/b)\pi(c, d)$ тогда и только тогда, когда $a*d\leqslant b*c$.

Показать, что $\pi$ является полным отношением порядка.

## § 6. Отношения на базах данных и структурах данных

Как уже установлено, все вокруг определяется отношениями. Достаточно лишь взять отношение $s$ на переменных $(x_1, \ldots, x_n)$ так, чтобы можно было построить

множество

$$\{(x_1, x_2, \ldots, x_n): s(x_1, x_2, \ldots, x_n) \text{ истинно}\}.$$

Пусть задан набор $(x_1, \ldots, x_n)$. Отношение $s(x_1, \ldots, x_n)$ можно *разрешить*, т. е. выяснить, $s(x_1, \ldots, x_n)$ истинно или ложно. Конечно, $s$ не обязательно будет представлено «хорошей» формулой. Нетрудно показать, что вместо отношения $s$, определяющего множество наборов длины $n$, любое множество таких наборов также определяет отношение (и содержательные свойства $s$). Эти два подхода эквивалентны.

Определение. При обработке данных наборы из $n$ элементов называют *записями*; элементы этих наборов называют *полями*. Записи, определяющие отношение, обычно содержатся в файле. Если потребовать, чтобы несколько файлов содержали совокупность записей, удовлетворяющих некоторым отношениям, то мы получим (относительную) *базу данных*.

Замечание. Для случая обработки данных мы сейчас употребили термин «поле». В гл. 5 мы будем употреблять этот термин в математическом смысле, однако в данном случае это не приводит к недоразумению.

Таким образом, это дает нам первый реальный пример отношений, которые в большей мере связаны с вычислениями, в частности, с прикладными задачами. Тем не менее краткое обсуждение некоторых простейших свойств баз данных не только обеспечивает основу дальнейшего математического исследования отношений, но и проясняет некоторые факторы, понимание которых необходимо для эффективного управления системами баз данных.

Современная теория баз данных включает в себя изучение так называемых нормальных форм, однако обоснование некоторых из них очевидно лишь в простых случаях. Мы рассмотрим только три формы для следующих задач:

— вставить новый набор из $n$ элементов;

— удалить набор из $n$ элементов;

— модифицировать набор, содержащий $n$ элементов.

Начнем с простейшей нормальной формы.

Определение. *Файлы в первой нормальной форме* (1NF), или — более просто — *нормализованные файлы*, имеют записи фиксированной длины, состоящие из элементов, взятых из множеств, чьи элементы далее не могут быть разбиты, и в каждый момент времени этот

файл может быть представлен как массив значений $M \times N$. Каждая запись, будучи набором из $n$ элементов, может быть записана как строка массива. //

Пример 6.1. Рассмотрим отношение FAM1 (см. выше), в котором мы собрали вместе родителей и детей. Каждая запись содержит в указанном порядке фамилию и имена отца, матери и детей. Следовательно, мы имеем записи

(Смит, Джой, Джойс, (Сэлли, Бен)) ∈ FAM1,
(Браун, Фред, Лиза, (Люси)) ∈ FAM1.

Теперь, если мы обозначим через $F$ и $M$ множества отцов и матерей, то из определения следует, что

Джой(Смит) ∈ $F$, Джойс(Смит) ∈ $M$,
Фред(Браун) ∈ $F$, Лиза(Браун) ∈ $M$.

Таким образом, Люси является членом семьи Браун, но Сэлли и Бен не являются детьми семьи Смит. Так как в этой семье более одного ребенка, то соответствующая запись больше, и, следовательно, нарушены условия первой нормальной формы.

Из FAM1 мы можем получить отношение FAM2, построив его из $S$, $F$, $M$ и $C$, где $S$ — множество фамилий, а $C$ — множество детей, путем конструирования записей:

(Смит, Джой, Джойс, Сэлли),
(Смит, Джой, Джойс, Бен),
(Браун, Фред, Лиза, Люси).

Отношение FAM2 находится в 1NF и может быть представлено при помощи табл. 2.1.

Таблица 2.1

| Фамилия | Отец | Мать | Ребенок |
|---|---|---|---|
| Смит | Джой | Джойс | Сэлли |
| Смит | Джой | Джойс | Бен |
| Браун | Фред | Лиза | Люси |

Однако не совсем ясно, что будет, если, например, супруги Джонс не имеют детей? Если мы хотим иметь в файле запись о них, то следует пересмотреть структуру файла. Это означает, что все следует начать сначала. Введем следующую терминологию.

Определение. При использовании таблицы для изображения отношения (файла с $n$-мерными наборами/записями, записываемыми в виде строк) столбцы называются *атрибутами.* //

Следовательно, ФАМИЛИЯ, ОТЕЦ, МАТЬ и РЕБЕНОК являются атрибутами различных полей в FAM2. Для получения доступа к записям в файле используются так называемые ключи. Более точно это может быть определено в терминах атрибутов.

Определение. Атрибут или (упорядоченное) множество атрибутов, чьи значения однозначно определяют запись в файле, называются *ключом* этого файла. (Заметим, что в файле может быть много различных ключей.) //

Каждый ключ отношения/файла FAM2 должен включать атрибут РЕБЕНОК.

Перейдем к другим примерам.

Пример 6.2. Каждый владелец компьютера должен покупать к нему запасные части. Поэтому мы можем рассмотреть файл, структура которого показана в табл. 2.2.

Таблица 2.2

| КОМПАНИЯ | ОТДЕЛЕНИЕ | МЕНЕДЖЕР |
|---|---|---|
| ACE | LONDON | SMITH |
| IBL | LONDON | JONES |
| DATAMETZ | BIRMINGHAM | JONES |
| PRINTACO | MANCHESTER | BROWN |
| WOOLIES | BIRMINGHAM | BROWN |
| RTX | LONDON | SMITH |
| OXONDATA | OXFORD | WILSON |

Атрибут КОМПАНИЯ является ключом в SUP1; вся другая информация в файле доступна при посредстве ключа. Таким образом, например, можно извлечь атрибут ОТДЕЛЕНИЕ при помощи ключа WOOLIES или же МЕНЕДЖЕР из RTX. //

Определение. Если запись локализована с помощью некоторого ключа, то поле, выделяемое из этой записи, называется *проекцией.* В данном контексте проекцией является «из». Будем также говорить, что эти атрибуты *зависят от ключа.* //

На рис. 2.7 представлен графический пример зависимостей в SUP1.

Пример 6.3. Модифицируем файл SUP1 с целью включения туда информации об имеющихся на складе запасных частях и об их количествах, которые отдельный

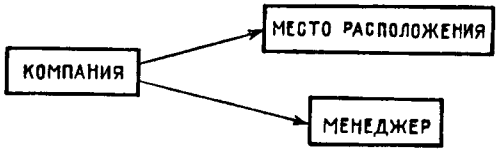



Рис. 2.7

поставщик желает продать в отдельной сделке. Включим в файл также код поставки, из которого мы можем выяснить скорость и частоту поставок. Во избежание излишней детализации введены номер компании и номер запасной части (табл. 2.3). //

Таблица 2.3

| КОМПАНИЯ | МЕСТО РАСПОЛОЖЕНИЯ | ПОСТАВЩИК | ЗАП. ЧАСТЬ | КОЛИЧЕСТВО |
|---|---|---|---|---|
| 1 | LONDON | 2 | 1 | 10 |
| 1 | LONDON | 2 | 2 | 1 |
| 1 | LONDON | 2 | 3 | 10 |
| 2 | LONDON | 2 | 2 | 2 |
| 2 | LONDON | 2 | 4 | 5 |
| 3 | B'HAM | 4 | 2 | 2 |
| 3 | B'HAM | 4 | 3 | 10 |
| 3 | B'HAM | 4 | 4 | 4 |
| 5 | B'HAM | 4 | 1 | 10 |
| 5 | B'HAM | 4 | 3 | 10 |
| 6 | LONDON | 2 | 1 | 10 |

Из табл. 2.3 выясним, какие преобразования мы можем делать с SUP2, а какие нет:

а) вставка. Например, мы не можем добавить в файл запись, указывающую, что компания 4 (PRINTACO) находится в Манчестере, без указания деталей, которые она может поставлять;

б) удаление. Если компания 6 (RTX) прекратила поставки запчастей 1, тогда мы обязаны удалить все записи, относящиеся к этой компании и имеющие в поле ЗАП. ЧАСТЬ код 1;

в) модификация. Если код поставщика для Лондона изменился, например, из-за транспорта, то соответствующее поле должно быть изменено в каждой записи, где есть код LONDON в поле МЕСТО РАСПОЛОЖЕНИЯ.

Что можно сделать для того, чтобы уменьшить или отодвинуть эти проблемы? С практической точки зрения мы должны выделить информацию в SUP2 так, чтобы по возможности избежать повторений. Таким образом,

Таблица 2.4

SUP3

| КОМПАНИЯ | МЕСТО РАСПОЛОЖЕНИЯ | ПОСТАВЩИК |
|---|---|---|
| 1 | LONDON | 2 |
| 2 | LONDON | 2 |
| 3 | B'HAM | 4 |
| 5 | B'HAM | 4 |
| 6 | LONDON | 2 |

ЗАП. ЧАСТЬ

| КОМПАНИЯ | ЗАП. ЧАСТЬ | КОЛИЧЕСТВО |
|---|---|---|
| 1 | 1 | 10 |
| 1 | 2 | 1 |
| 1 | 3 | 10 |
| 2 | 2 | 2 |
| 2 | 4 | 5 |
| 3 | 2 | 2 |
| 3 | 3 | 10 |
| 3 | 4 | 4 |
| 5 | 1 | 10 |
| 5 | 3 | 10 |
| 6 | 1 | 10 |

Таблица 2.5

| КОМПАНИЯ | МЕСТО РАСПОЛОЖЕНИЯ | ПОСТАВЩИК |
|---|---|---|
| 1 | LONDON | 7 |
| 2 | LONDON | 7 |
| 3 | B'HAM | 4 |
| 4 | M'CHESTER | 3 |
| 5 | B'HAM | 4 |
| 6 | LONDON | 7 |

ЗАП. ЧАСТЬ

| КОМПАНИЯ | ЗАП. ЧАСТЬ | КОЛИЧЕСТВО |
|---|---|---|
| 1 | 1 | 10 |
| 1 | 2 | 1 |
| 1 | 3 | 10 |
| 2 | 2 | 2 |
| 2 | 4 | 5 |
| 3 | 2 | 2 |
| 3 | 3 | 10 |
| 3 | 4 | 4 |
| 5 | 1 | 10 |
| 5 | 3 | 10 |

мы получаем возможность вставки/удаления части записи в SUP2. Возможное и, на наш взгляд, разумное разделение дается в SUP3 (табл. 2.4). Тогда остаток информации в SUP2 может содержаться в поле ЗАП.ЧАСТЬ. Используя эту конфигурацию, можно, например:

а) включить в SUP3 запись, означающую, что компания 4 находится в Манчестере (код поставщика 3);

б) удалить ссылку на компанию 6 как на поставщика запчасти 1, но оставить соответствующий код в SUP3;

в) изменить код поставщика для LONDON на 7 путем замены только трех входов, соответствующих компаниям с кодом 1, а не всех шести.

Результаты этих изменений приведены в табл. 2.5. Это уже значительно лучше, однако все же может быть

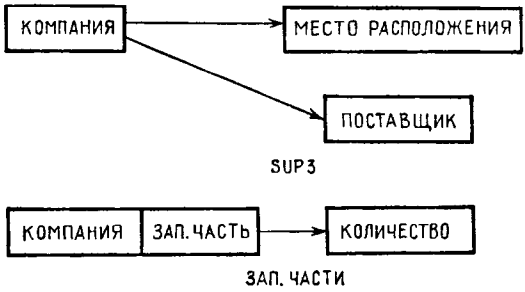



Рис. 2.8

еще усовершенствовано. Чтобы увидеть, в каком направлении продолжать исследования, отделим ключи и подчиненные части (рис. 2.8). Заметим, что ЗАП. ЧАСТЬ требует объединенного ключа.

Все не-ключи непосредственно связаны с ключом. Это дает нам следующее свойство нормальной формы. //

Определение. Файл имеет *вторую нормальную форму* (2NF), если он имеет форму 1NF и неключевые атрибуты полностью независимы от ключа. //

Пример 6.3 (продолжение). Файл SUP3 все еще является достаточно сложным в том смысле, что для данной записи ПОСТАВЩИК может быть установлен при помощи исследования поля КОМПАНИЯ или же поля МЕСТО РАСПОЛОЖЕНИЯ. Это является причиной того, что в требовании а) код поставщика для Манчестера должен быть вставлен перед записью кода 4 компании, а требование б), возможно, потребует изменить более одной записи для модификации единственного поля данных, относящегося к коду поставщика. На практике мы можем убрать эту проблему проектированием SUP3 в SUP4 и DEL (табл. 2.6). (Заметим, что коды поставщика, изменяемые таким образом, препятствуют той возможности, что любые другие записи в файле вызывают противоречивую информацию. В SUP3 можно иметь запись вида «ПОСТАВЩИК КОМПАНИИ 6-2» и «ПО-

СТАВЩИК КОМПАНИИ 1-7» на некотором этапе модификации SUP2, несмотря на тот факт, что обе компании находятся в Лондоне.)

Зависимость отношений в SUP4 и DEL изображена на рис. 2.9. //

Нетранзитивность отношения зависимости является внутренним свойством, из которого возникает понятие третьей нормальной формы.

Таблица 2.6

| КОМПАНИЯ | МЕСТО РАСПОЛОЖЕНИЯ |
|---|---|
| 1 | LONDON |
| 2 | LONDON |
| 3 | B'HAM |
| 4 | M'CHESTER |
| 5 | B'HAM |
| 6 | LONDON |

| МЕСТО РАСПОЛОЖЕНИЯ | ПОСТАВЩИК |
|---|---|
| LONDON | 7 |
| B'HAM | 4 |
| M'CHESTER | 3 |

Определение. Файл находится в *третьей нормальной форме* (3NF), если он является файлом 2NF, и каждый атрибут, не являющийся ключом, нетранзитивным образом зависит от ключа. //

Возможен и другой путь — каждый атрибут, не являющийся ключом, зависит только от ключа и ни от чего другого.

Как было отмечено ранее, существует много других «нормальных» форм, но мы не ставим изучение файлов

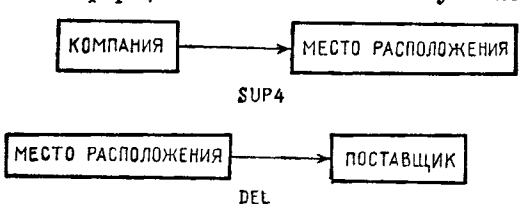



Рис. 2.9

своей целью в дальнейшем. Достаточно лишь иметь в виду, что информация в файлах является одной из реализаций математического понятия отношения.

Практическое использование отношений SUP4 и DEL требует явной связи атрибута МЕСТО РАСПОЛОЖЕНИЯ файла SUP4 с атрибутом МЕСТО РАСПОЛОЖЕНИЯ файла DEL. Это — отношение эквивалентности

(между компонентами различных файлов, имеющих одно и то же имя). Подобные отношения эквивалентности могут быть использованы для определения внутренних связей и других структурных данных. В качестве иллюстрации рассмотрим рис. 2.10. Диаграмма на рис. 2.10, *a*

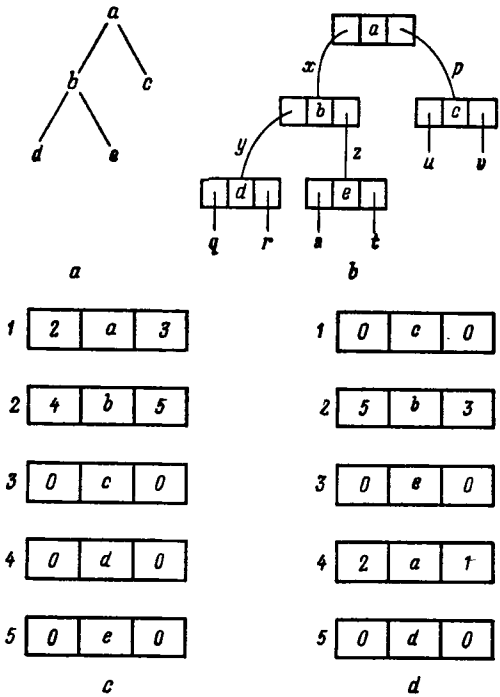



Рис. 2.10

изображает дерево, диаграмма на рис. 2.10, *b* подобна диаграмме структурных данных, а диаграммы на рис. 2.10, *c* и *d* описывают их возможные применения. Отметим, что отношения эквивалентности различны, но результирующие структурные связи сохраняются. С математической точки зрения это является разбиением на классы эквивалентности. Следовательно, мы можем определить произвольное представление этого дерева как элемент множества $T = D/E$, где

$$D = \{a = (x, a, p),\ b = (y, b, z),\ c = (u, c, v),$$
$$d = (q, d, r),\ e = (s, e, t)\},$$
$$E = \{(x, b),\ (y, d),\ (p, c),\ (z, e)\}.$$

Эти вопросы будут обсуждаться в § 4 гл. 3.

## § 7. Составные отношения

Подобно тому как мы устанавливали внутренние связи в файлах, для выделения некоторых данных из информации (например, ПОСТАВЩИК и МЕСТО РАСПОЛОЖЕНИЯ 3 посредством файлов DEL и SUP4 в примере 6.3) часто приходится связывать бинарные отношения друг с другом. Руководствуясь предыдущими рассуждениями, можно определить это понятие следующим образом.

Определение. Пусть заданы множества $A$, $B$ и $C$ и отношения $\sigma$ между $A$ и $B$ и $\rho$ между $B$ и $C$. Определим отношение между $A$ и $C$ следующим образом: оно действует из $A$ в $B$ посредством $\sigma$, а затем из $B$ в $C$ посредством $\rho$. Такое отношение называют *составным* и обозначают $\rho \circ \sigma$, т. е.

$$(\rho \circ \sigma)(a) = \rho(\sigma(a)). \quad //$$

Следовательно, $(x, y) \in (\rho \circ \sigma)$, если существует $z \in B$ такое, что $(x, z) \in \sigma$ и $(z, y) \in \rho$. Отсюда следует, что $\mathscr{D}_{\rho\circ\sigma} = \sigma^{-1}\mathscr{D}_\rho$. Чтобы проиллюстрировать ситуацию, рассмотрим рис. 2.11. Области определения и значений $\sigma$ и

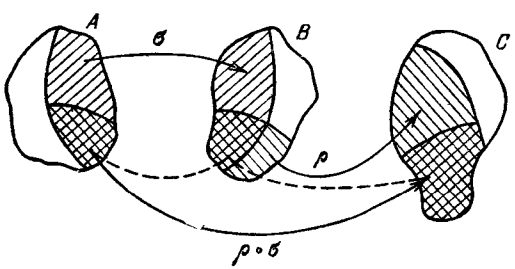



Рис. 2.11

$\rho$ заштрихованы в разных направлениях. Следовательно, сегменты с двойной штриховкой на $A$, $B$ и $C$ представляют собой $\mathscr{D}_{\rho\circ\sigma}$, $\mathscr{D}_\rho \cap \mathscr{R}_\sigma$ и $\mathscr{R}_{\rho\circ\sigma}$ соответственно.

Замечание. Из записи отношений $\sigma$ и $\rho$ следует, что они применяются справа налево. Следовательно, $(\rho \circ \sigma)(a)$ означает, что вначале берется $a$ и преобразуется посредством $\sigma$, а затем преобразуется посредством $\rho$. В алгебре это иногда записывают в виде $a\sigma\rho$. Следует обращать внимание при чтении других математических книг на то, какой порядок выполнения отношений принят в той книге.

Пример 7.1. Пусть $\sigma$ и $\rho$ — отношения на N такие, что

$$\sigma = \{(x, x+1): x \in \mathrm{N}\}, \quad \rho = \{(x^2, x): x \in \mathrm{N}\}.$$

Тогда

$$\mathscr{D}_\rho = \{x^2: x \in \mathrm{N}\}, \quad \mathscr{D}_\sigma = \{x: x, x+1 \in \mathrm{N}\} = \mathrm{N},$$

$$\mathscr{D}_{\rho\circ\sigma} = \sigma^{-1}\mathscr{D}_\rho =$$

$$= \{x: x \in \mathrm{N} \text{ и } x+1 = y^2, \text{ где } y \in \mathrm{N}\} = \{3, 8, 15, 24, \ldots\}$$

(рис. 2.12). //

В случае, когда мы рассматриваем отношение на множестве, оно может быть скомбинировано само с

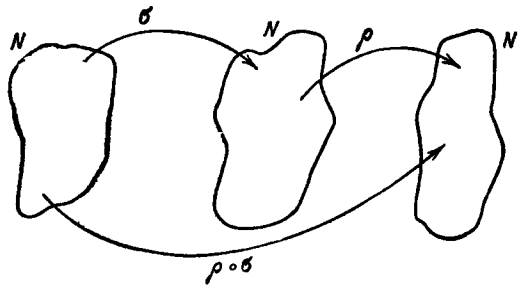



Рис. 2.12

собой. Например, используя отношения из примера 7.1, имеем

$$\sigma \circ \sigma = \{(x, x+2): x \in \mathrm{N}\} \text{ и } \rho \circ \rho = \{(x^4, x): x \in \mathrm{N}\}.$$

Эти отношения можно также обозначать соответственно $\sigma^2$ и $\rho^2$. В общем-то эти обозначения не совсем законны для множеств, однако их легко можно обосновать, поскольку если $(x, y) \in \sigma \circ \sigma$, то $((x, z), (z, y)) \in \sigma \times \sigma$ при некотором $z$; никакого недоразумения при этом не возникает, поскольку известна структура получаемого результата.

Используя это обозначение, мы можем определить $\sigma^n$ для любого $n \in \mathrm{N}$, $n > 1$, следующим образом:

$$\sigma^n = \{(x, y): x\sigma z \text{ и } z\sigma^{n-1}y \text{ для некоторого } z\}.$$

Если мы вновь возьмем отношения $\sigma$ и $\rho$ из примера 7.1, то получим

$$\sigma^n = \{(x, x+n), x \in \mathrm{N}\} \text{ и } \rho^n = \{(x^{2^n}, x): x \in \mathrm{N}\}.$$

Хотелось бы рассмотреть вопрос о том, насколько в этих

случаях применима аналогия с умножением. Пусть $A$ — множество, а $R$ — отношение на $A$. Тогда отношения $I_A \circ R$, $R$ и $R \circ I_A$ эквивалентны; поэтому $I_A$ является тождественным отношением на $A$, которое ведет себя подобно числу 1 по отношению к умножению чисел. Чтобы дополнить аналогию, желательно было бы иметь возможность писать $R^{-1} \circ R = I_A = R \circ R^{-1}$. Однако в общем случае этого делать нельзя. Для того чтобы иметь такую возможность, необходимо наложить дополнительные условия, которые мы рассмотрим в следующей главе.

Рассмотрим теперь несколько упражнений. Они изображают ситуации, которые «легко описать» в математических терминах, однако следует заботиться о том, чтобы в отношение включались только те пары, которые ему принадлежат. Мы обнаружим, что иногда полезно воспользоваться диаграммами.

Упражнение 2.6.

1. Пусть $R$ и $S$ определены на $P$, где $P$ — множество всех людей, следующим образом:

$$R = \{(x, y)\colon x, y \in P \text{ и } x \text{ является отцом } y\},$$

$$S = \{(x, y)\colon x, y \in P \text{ и } x \text{ — дочь } y\}.$$

Описать явно следующие отношения:

а) $R^2$; б) $S^2$; в) $R \circ S$; г) $S \circ R$;
д) $S \circ R^{-1}$; е) $R^{-1} \circ S$; ж) $R^{-1} \circ S^{-1}$;
з) $S^{-1} \circ R$; и) $S^{-1} \circ S^{-1}$; к) $S^{-1} \circ R^{-1}$.

## § 8. Замыкание отношений

Понятие замыкания является фундаментальным математическим понятием и используется в большинстве разделов математики. Чтобы проиллюстрировать это понятие, рассмотрим следующий пример.

Возьмем объект $x_0$ и процесс $p$, который порождает множество и определяет последовательность $x_1, x_2, \ldots$ $\ldots, x_n, \ldots$ такую, что

$$x_1 \in p(x_0),$$
$$x_2 \in p(x_1),$$
$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$
$$x_n \in p(x_{n-1}),$$
$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad .$$

Множество, содержащее все элементы всех последовательностей, которые могут быть выведены при помощи $p$, и начинающиеся с $x_0$, называется *замыканием процесса* $p$ относительно $x_0$. Поэтому «ответ» будет содержаться в $p^n(x_0)$ при некотором $n$. Однако мы не знаем заранее значение $n$. Более того, если мы возьмем произвольный элемент $y$ из этого замыкания и выполним процесс $p$, начиная с $y$, то не получим ничего нового. Результат уже содержится в замыкании. Множество не может быть расширено таким путем (оно замкнуто).

Пример 8.1. Возьмем квадрат $S$, размеченный, как это показано на рис. 2.13, и определим процесс $r$ следующим образом. Из заданного положения $S$ процесс $r$ порождает множество всех положений, получаемых в результате поворота по часовой стрелке на прямой угол.

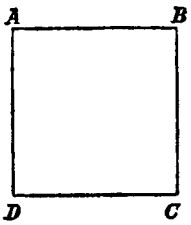



Рис. 2.13

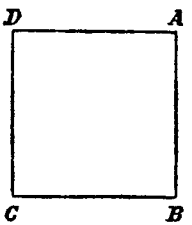



Рис. 2.14

Таким образом, $r(S)$ дает конфигурацию, изображенную на рис. 2.14. После применения $r$ четыре раза, мы вернемся к положению, с которого начали, и, следовательно, замыкание в данном случае есть множество из четырех позиций.

$$\left\{\begin{matrix}A & B\\ D & C\end{matrix}, \begin{matrix}D & A\\ C & B\end{matrix}, \begin{matrix}C & D\\ B & A\end{matrix}, \begin{matrix}B & C\\ A & D\end{matrix}\right\}. \text{//}$$

Рассмотрим теперь, что произойдет, если процесс определить при помощи отношения. (В действительности это всегда возможно, потому что мы можем определить подходящее отношение при помощи множества $\{(x, y): y \in p(x)$, где $p$ — изучаемый процесс$\}$.) Для построения замыкания отношения $A$ достаточно иметь составные отношения $A$, $A^2$, ..., $A^n$, ..., которые затем комбинируются обычным теоретико-множественным путем.

Определение. *Транзитивным замыканием* (или просто *замыканием*) отношения $A$ на множестве называется бесконечное объединение

$$\bigcup_{n=1}^{\infty} A^n = A \cup A^2 \cup A^3 \cup \ldots //$$

Транзитивность замыкания отношения следует, очевидно, из его определения, однако слово «транзитивное» часто включают, чтобы подчеркнуть различие между этой и подобной ей операцией, которая вскоре будет определена. Транзитивное замыкание отношения $A$ обозначают $A^+$.

Пример 8.2.

1. Пусть $R$ — отношение на $\mathbf{N}$ такое, что $R = \{(x, y): y = x + 1\}$. Тогда $R^+ = \{(x, y): x < y\}$.
2. Пусть $\sigma$ — отношение на $\mathbf{Q}$ такое, что $\sigma = \{(x, y): x < y\}$. Тогда $\sigma^+ = \sigma$.
3. Пусть $\rho$ — отношение на $\mathbf{Q}$ такое, что $\rho = \{(x, y): x * y = 1\}$. Тогда

$$\rho^+\{(x, x): x \neq 0\} \cup \rho.$$

4. Пусть $L$ — множество станций Лондонского метро и $a$, $b$ и $c$ — последовательные станции. Если отношение $N$ на $L$ определено как $N = \{(x, y): x$ является следующей за $y$ станцией$\}$, то $(a, b)$ и $(b, c) \in N$ и $(a, a)$, $(b, b)$, $(c, c)$ и $(a, c) \in N^2$. Следовательно $N^+ = U_L = L \times L$. //

Из этих примеров легко видеть, что замыкание отношения в общем случае не является рефлексивным. Однако иногда удобно сделать его таким. Это можно легко сделать. Вначале мы примем удобное допущение, что тождественное отношение на $X$, $I = \{(x, x): x \in X\}$ является нулевой степенью произвольного отношения на $X$. Таким образом, $A^0 = I$ для любого $A$.

Определение. *Рефлексивным замыканием* $A^*$ отношения $A$ называют множество

$$A^* = \bigcup_{n=0}^{\infty} A^n,$$

Замыкания отношений связаны между собой очевидным соотношением

$$A^* = A^+ \cup I. //$$

Пример 8.3. Используя отношения, определенные в предыдущих примерах, получаем

$$R^* = \{(x, y): x \leqslant y\}, \quad \sigma^* = \{(x, y): x \leqslant y\},$$
$$\rho^* = \rho^+ \cup \{(0, 0)\}, \quad N^* = N^+. \ //$$

Практические методы получения замыканий отношений будут обсуждаться в гл. 6, 7.

Упражнение 2.7.

1. Используя отношения $R$ и $S$ упражнения 2.6, описать замыкания следующих отношений:

а) $R^+$; б) $S^+$; в) $R^*$; г) $S^*$;

д) $(S \circ S^{-1})^+$; е) $(R^{-1} \circ R)^*$; ж) $(S^2 \circ R^2)^+$.

ГЛАВА 3

# ФУНКЦИИ

Понятие функции может быть уже известным читателю, однако в этой главе мы будем рассматривать функции как множество бинарных отношений. Собственно, будет дано определение функции и близкого к ней понятия отображения и изучены различные свойства, которыми они обладают. Затем функции будут использованы для того, чтобы формализовать процесс вычислений и дать определение мощности множества. В заключение будут рассмотрены конкретные функции, представляющие особый интерес, и функции, которые удобно определить с помощью операторов.

## § 1. Функции и отображения

Определение. Бинарное отношение $\rho$ между множествами $A$ и $B$ является *функцией*, если из $a\rho b$ и $a\rho c$ следует, что $b = c$; поэтому для любого $x \in A$ существует одно $y \in B$ такое, что $x\rho y$. Можно дать определение функции следующим образом:

$$\rho(x) = \varnothing \quad \text{или} \quad \rho(x) = \{y\}. \ //$$

Следует помнить, что если $\rho(x)$ существует (т. е. $\rho(x) \neq \varnothing$), то этот элемент единствен. В случае, когда $\rho(x) \neq \varnothing$, обычно опускают скобки при обозначении множества и записывают

$$y = \rho(x).$$

Функции обычно обозначают строчными латинскими буквами $f$, $g$, $h$, ... или в специальных случаях особыми сочетаниями, например sin, log, $F_n$, ... Если $f$ — функция между множествами $A$ и $B$, то этот факт может быть записан как $f\colon A \to B$. В дальнейшем, если $x \in A$ и $xfy$, мы будем обозначать это соотношение следующим образом:

$$f\colon x \mapsto y.$$

Это обозначение часто используют для того, чтобы описать правило, определяющее функцию (если оно существует).

Пример 1.1. Функция $f\colon A \to A$, где $A = \{-1, 0, 1\}$, определяется соотношением $f\colon x \mapsto x^3$. //

Даже когда $f$ не является отображением (см. ниже), мы часто будем использовать фразы типа «$f$ отображает $x$ в $x^3$». Понятия области определения и области значений в данном случае содержательны, поскольку функция является отношением, однако следует заметить, что обратная функция может не существовать.

Пример 1.2. На множестве $\{-1, 0, 1\}$ отношение $f\colon x \mapsto x^2$ является функцией, но обратной функции не существует, поскольку $f^{-1}(1) = \{-1, 1\}$. //

Определение. Функция $f\colon A \to B$ является *отображением*, если ее область определения совпадает с $A$, т. е. $\mathscr{D}_f = A$. Функции, не являющиеся отображениями, называют частичными. Отображение на множество называют *трансформацией* (*преобразованием*).

Замечание. Часто терминология отличается от принятой в этой книге, особенно в американских книгах. Термины «функция» и «отображение» иногда используют как синонимы, а отображение в том смысле, как мы его определили, называют полной функцией.

Конечно, каждая функция может рассматриваться как отображение на своей области определения; иногда это полезно, когда строятся сложные функции. Как мы увидим в последующих параграфах, исследовать свойства функций легче в тех случаях, если на функцию наложить некоторые условия.

Функция $f\colon A \to \mathbf{R}$ называется функцией, принимающей действительные значения, а функция, область определения которой совпадает с $\mathbf{R}$, называется вещественной.

Рассмотрим ограничения, которые помогут нам понять, что происходит с отдельными элементами в результате применения к ним функций. Будем заниматься классификацией функций, определенных на множествах, и предполагать, что мы имеем мало информации об этих множествах. Далее будут рассмотрены функции на множествах, где определены такие операции, как сложение.

Определение. Функция $f\colon A \to B$ называется *сюръективной* (*на*), если $\mathscr{R}_f = B$. Это означает, что для данного $b \in B$ имеем $f^{-1}(b) \neq \varnothing$. Функция $f\colon A \to B$ яв-

ляется *инъективной*, если из $a_1$, $a_2 \in A$ и $f(a_1) = f(a_2)$ следует, что $a_1 = a_2$. //

Итак, если $f$: $A \to B$ и $f$ сюръективна, то для любого $b \in B$ имеем $f^{-1}(b) \in \mathscr{P}(A) \setminus \varnothing$. Это может быть проинтерпретировано следующим образом: каждая точка из $B$ является «острым концом» по крайней мере одной $f$-стрелы, выходящей из $A$. Проиллюстрировать эту ситуацию достаточно трудно (исключая тривиальные случаи).

С другой стороны, наглядную характеристику инъективности легко дать в виде ограничения или запрета.

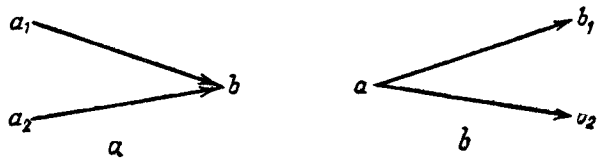



Рис. 3.1

Функция $f$ *не* инъективна в случае, изображенном на рис. 3.1, *а*.

Для сравнения на рис. 3.1, *b* изображено зеркальное отражение, которое отличает функцию от бинарного отношения.

Если $f$: $A \to B$ инъективна и $a \in \mathscr{D}_f$, то $a = f^{-1}(f(a))$. Далее, если $b \in \mathscr{R}_f$, т. е. $b \in \mathscr{D}_{f^{-1}}$, и в данном случае $f^{-1}$ — функция, то $b = f(f^{-1}(b))$.

Следовательно, если мы определим функцию $I_X$: $X \to X$ как *тождественное отображение* на $X$, т. е. $I_X$: $x \mapsto x$ для всех $x \in X$, тогда, если $f$ инъективна, то

$$f^{-1} \circ f = I_{\mathscr{D}_f} \quad \text{и} \quad f \circ f^{-1} = I_{\mathscr{R}_f}.$$

Используя функцию $f$ из примера 1.2, мы видим, что $f^{-1}(f(1)) \neq 1$. Это означает, что первое из этих тождеств в общем случае неверно, однако, как мы увидим в § 2, вычисления становятся гораздо легче в случаях, когда оба тождества имеют место. Прежде чем продолжить изложение, обсудим терминологию, объединяющую введенные выше свойства. Функция $f$ *биективна*, если она сюръективна и инъективна. Биективное отображение называют *биекцией*. Следовательно, используя биекцию $f$ между $A$ и $B$, можно брать элементы из $A$ и переходить в $B$ посредством $f$, осуществляя некоторые вычисления. Переход назад к $A$ осуществляется посредством $f^{-1}$.

Термины инъекции и сюръекции также применяются (для описания инъективного и сюръективного отображений), однако мы редко будем их использовать.

Упражнение 3.1.

1. Какие из указанных ниже отношений на множестве $\{-10, -9, \ldots, 0, 1, \ldots, 9, 10\}$ являются функциями? Дать противоречащие примеры в случаях, когда отношение не является функцией. (В 3.1, 1г) отношение порядка $\leqslant$ является отношением, индуцированным Z. В 3.1, 1а)—к) $|x|$ определяется следующим образом: $|x| = x$, если $x \geqslant 0$, и $|x| = -x$, если $x < 0$.)

а) $\rho_1 = \{(x, y): x = y^2\}$; б) $\rho_2 = \{(x, y): x^2 = y\}$;
в) $\rho_3 = \{(x, y): x = -y\}$; г) $\rho_4 = \{(x, y): x \leqslant y\}$;
д) $\rho_5 = \{(x, y): x * y = 6\}$;
е) $\rho_6 = \{(x, y): x^3 = y\}$; ж) $\rho_7 = \{(x, y): x = y^3\}$;
з) $\rho_8 = \{(x, y): x = |y|\}$; и) $\rho_9 = \{(x, y): |x| = |y|\}$;
к) $\rho_{10} = \{(x, y): y * |y| = x * |x|\}$.

2. Построить функцию $f: A \to A$, где $A = \{0, 1\}$, не имеющую обратной.

3. Какие из следующих функций являются отображениями:

а) $f$ на $\mathbf{R}$ определяется следующим образом: $\{(x, x^4): x \in \mathbf{R}\}$;
б) $f$ на $\mathbf{R}$ определяется следующим образом: $\{(x^3, x): x \in \mathbf{R}\}$;
в) $f$ на $\mathbf{R}$ определяется следующим образом: $\{(x, x^2): x \in \mathbf{R}\}$;
г) $f$ на $\mathbf{R}$, $f: x \mapsto \sin(x)$;
д) $f$ на $\mathbf{R}$, $f: x \mapsto 1/x$;
е) $f$ на $\mathbf{Q}$, $f: x \mapsto \arcsin x$;
ж) $f: A \to \mathscr{P}(A)$ определяется следующим образом: $f: x \mapsto \{x\}$;
з) $f: \mathscr{P}(A) \to A$ определяется следующим образом: $f = \{(x, y): y \in x \cap \{a\}\}$, где $a$ — фиксированный элемент из $A$?

4. Пусть $f: A \to B$ и $g: B \to C$ — отношения. Что является областью определения $g \circ f$:

а) когда $f$ и $g$ — функции;
б) когда $f$ — функция, $g$ — отображение;
в) когда $f$ — отображение, а $g$ — функция;
г) когда $f$ и $g$ — отображения?

5. Доказать, что если функция $f$ инъективна, то существует $f^{-1}$.

6. Если функция $f$ сюръективна, следует ли отсюда, что $f^{-1}$ — отображение?

7. Построить пример, показывающий, что функция на $A = \{-1, 0, 1\}$, определенная как $f\colon x \mapsto x^2$, такова, что $f^{-1} \circ f \neq I_A$.

8. Пусть $f\colon A \to B$ и $g\colon B \to C$ — функции. Доказать, что:

а) если $f$ и $g$ инъективны, то $g \circ f$ инъективна;

б) если $f$ и $g$ сюръективны, то $g \circ f$ также сюръективна.

9. Пусть $f\colon A \to B$ и $g\colon B \to C$ — функции и $g$ сюръективна. Достаточно ли этого, чтобы обеспечить сюръективность $g \circ f$?

## § 2. Обратные функции и отображения

Используя результаты, полученные в предыдущем параграфе, исследуем сложные функции. Пусть дана функция $f\colon A \to B$; в этом случае $f^{-1}$ является функцией тогда и только тогда, когда $f$ инъективна, а отображением тогда и только тогда, когда $f$ биективна. В большинстве рассматриваемых нами случаев $f$ — биекция; тогда $f^{-1}$ — также биекция, а функции $f^{-1} \circ f$ и $f \circ f^{-1}$ являются тождественными отображениями.

Рассмотрим функции $f\colon A \to B$ и $g\colon B \to C$. Тогда:

а) если $f$ и $g$ инъективны, то существует $g \circ f$;

б) если $f$ и $g$ сюръективны, то также существует $g \circ f$ (см. упражнение 3.1, 8).

Обратным отношением к $g \circ f$ является $f^{-1} \circ g^{-1}$. Порядок должен быть обратным, как указано на рис. 3.2.

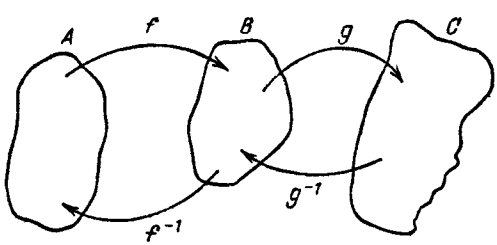



Рис. 3.2

Заметим, что если $g$ — отображение, т. е. $\mathscr{D}_g = B$, то $\mathscr{R}_f \subseteq \mathscr{D}_g$ и, следовательно, $\mathscr{D}_{g \circ f} = \mathscr{R}_g$. Аналогично, если $\mathscr{R}_f \supseteq \mathscr{D}_g$, то $\mathscr{R}_{g \circ f} = \mathscr{R}_g$. Если $f$ и $g$ инъективны, то существует $g \circ f$; следовательно, $f^{-1} \circ g^{-1}$ — функция. Суммируя вышесказанное, имеем: из $\mathscr{R}_f = \mathscr{D}_g$ следует, что

$g \circ f$: $\mathscr{D}_f \to \mathscr{R}_g$ — отображение; если $f$ и $g$ также инъективны, то $f^{-1} \circ g^{-1}$: $\mathscr{R}_g \to \mathscr{D}_f$ — биекция. Очевидно, что эти критерии выполняются, если $f$ и $g$ — биекции.

Упражнение 3.2.

1. Пусть $f$: $A \to B$ и $g$: $C \to B$; показать следующее:

а) если $f$ сюръективна и $g$ — отображение, то

$$\mathscr{R}_{g^{-1} \circ f} = C;$$

б) если $f$ и $g$ — биекции, то $(g^{-1} \circ f)^{-1} = f^{-1} \circ g$;

в) если $\mathscr{R}_g \subseteq \mathscr{R}_f$, то $(f \circ f^{-1} \circ g)C = \mathscr{R}_g$.

## § 3. Мощность множеств и счетность

Мы почти подошли к тому моменту, когда появляется возможность использовать понятие биекции для формализации понятия мощности и процесса вычислений. Вычисление важно не только само по себе, но также и потому, что функция является *вычислимой* тогда и только тогда, когда связанное с ней множество счетно. Вначале дадим определение множества N. Чтобы прояснить наши намерения, заметим, что любое число (1, 2 и т. д.) может быть использовано двумя различными способами: как существительное или как прилагательное, дающее номер другого существительного. Мы будем рассматривать числа как существительные.

В качестве предварительного определения N положим

$$\mathbf{N} = \{1\} \cup \{n + 1\colon n \in \mathbf{N}\}.$$

Из этого рекурсивного определения следует, что $1 \in \mathbf{N}$, и если к произвольному элементу из N прибавить 1, то полученный результат также принадлежит N. Следовательно, N содержит 1, $1 + 1 (= 2)$, $2 + 1 (= 3)$, $3 + 1 (= 4)$ и т. д.

К сожалению, это определение неприемлемо по причинам, которые будут рассмотрены ниже. Тем не менее здесь есть несколько важных моментов. Например, так как N (по крайней мере интуитивно) бесконечно, то мы должны иметь механизм, при помощи которого можно конструировать последующие элементы из конечного множества,— другими словами, никогда не сможем написать точное представление N. Мы также должны придумать имя числу, которое называем «один», и аналогично для «два» (сокращение $1 + 1$), «три» и т. д. Конечно,

мы могли бы выбрать любые имена или символы для этой цели, однако было бы неправильно использовать неудобные обозначения. Перейдем теперь к недостаткам данного определения. Проверка его обнаруживает, что оно содержит два новых символа: «1» и «+»; остальные символы известны из построения множеств. Символ «1» можно объяснить вышеуказанным способом. Однако символ «+» означает операцию *на* **N** и, следовательно, не может быть использован для определения **N**. (Операции будут определены в § 6.)

Чтобы выйти из этого затруднения, вернемся к основам теории множеств. Напомним, что нам нужно было построить число, которое на 1 больше максимального из всех предшествующих чисел. Легко получить аналогичный процесс для множеств (называемый построением сверхмножества с количеством элементов на 1 больше, чем в данном множестве).

Пример 3.1. Пусть $A = \{x, y, z\}$ и $B = \{x, y, z, A\}$. Тогда $A \subseteq B$ и $A \in B$, поэтому $B \setminus A = \{A\}$ и имеет только один элемент. //

Эта конструкция может быть перенесена на произвольное множество. Начиная от множества $X$, мы можем определить последующее множество (обозначается $X^*$): $X^* = X \cup \{X\}$. Чтобы использовать этот процесс для построения **N**, требуется некоторое начальное множество. Выберем в качестве такого множества $\{\varnothing\}$. Оно имеет один элемент. (Многие авторы начинают с $\varnothing$. Это порождает множество $\{0\} \cup \mathbf{N}$. Мы не считаем 0 натуральным числом, и в этом причина такого выбора начального элемента. Не существует универсального условия по отношению к 0 и **N**. Всегда следует проверять условные обозначения, принятые в других книгах, при обращении к ним.) Из $\{\varnothing\}$ создадим последовательности $\{\varnothing\}$: $\{\varnothing\}^* = \{\varnothing, \{\varnothing\}\}$, $\{\varnothing\}^{**} = \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}$ и т. д. Это приводит к прогрессии, которая является более привлекательной, чем 1, 1 + 1, 1 + 1 + 1, ..., по крайней мере ее конструкция является строго определенной. Чтобы навести порядок в вышесказанном, выберем временно имена для этих множеств.

Переименуем $\{\varnothing\}$ как 1, $1^*$ как 2, $2^*$ как 3 и т. д. Тогда

$$1 = \{\varnothing\},$$
$$2 = \{\varnothing, 1\},$$
$$3 = \{\varnothing, 1, 2\}, \ldots$$

Поэтому, например, множество 3 имеет три элемента. Чтобы избежать неточности, давайте снова изменим обозначения и определим множества

$$\mathbf{N}_m = m^{\bullet} \setminus \{\varnothing\} = \{1, 2, \ldots, m\},$$

$$\mathbf{N} = \mathbf{N}_1 \cup (\mathbf{N}_{m\oplus}\colon m \in \mathbf{N}).$$

Тогда из определения следует, что $|\mathbf{N}_m| = m$ (число $m$) и что если $a, b \in \mathbf{N}$, то $a \leqslant b$ тогда и только тогда, когда $\mathbf{N}_a \subseteq \mathbf{N}_b$. Поэтому наша вера в упорядоченность $\mathbf{N}$ формально обоснована. Итак, множества $\mathbf{N}$ и $\mathbf{N}_m$ (для каждого $m \in \mathbf{N}$) определены и могут быть использованы в дальнейшем. Введем некоторые понятия.

Определение. Два множества *биективны* (обозначается $X \sim Y$), если между ними существует биекция. Непустое множество *конечно*, если оно биективно некоторому $\mathbf{N}_m$ ($m \in \mathbf{N}$). Если $X \sim \mathbf{N}_m$, то мощность множества (обозначается $|X|$) равна $m$. (Числа в данном случае используются как прилагательные. Например, если $P$ — множество всех людей и $X \subseteq P$ таково, что $X \sim \mathbf{N}_m$, то $X$ есть множество из $m$ людей.) Напомним, что пустое множество $\varnothing$ биективно только по отношению к себе, является конечным и имеет мощность 0, т. е. $|\varnothing| = 0$.

Говорят, что множество *счетно*, если оно биективно $\mathbf{N}$. Символ $\aleph_0$ (алеф-нуль) часто используют для обозначения мощности $\mathbf{N}$. Множество называется *счетным*, если оно конечно или счетно, и может быть сосчитано с использованием биекции $f\colon \mathbf{N} \to X$, если $X$ счетно, или биекция $f\colon \mathbf{N}_m \to X$, если $|X| = m$, или $f\colon \varnothing \to X$; $i$-й элемент $X$ является образом $i$ отображения $f$. //

Перед тем как установить несколько полезных результатов, отметим одно существенное свойство множеств и биекций: отношение $\sigma$, определенное на множестве $S$ посредством

$$\sigma = \{(X, Y)\colon X \sim Y\},$$

является отношением эквивалентности, а подмножества $S$, входящие в классы эквивалентности, состоят из множеств, имеющих одинаковую мощность. Следовательно, чтобы продемонстрировать тот факт, что два множества имеют один и тот же размер, требуется построить биективное отображение между ними.

Пример 3.2. Покажем, что $|\mathbf{N}| = |\mathbf{Z}|$. Отображение

$$\psi\colon n \mapsto \begin{cases} \dfrac{1-n}{2}, & \text{если } n \text{ нечетно,} \\ n/2 & \text{в противном случае} \end{cases}$$

является биекцией между $\mathbf{N}$ и $\mathbf{Z}$. //

В предыдущем примере греческая буква $\psi$ использовалась для обозначения биекции. Использование греческих букв $\varphi$, $\psi$, $\chi$, ... для обозначения произвольных биекций является общепринятым в текстах по логике и будет здесь использоваться в этом контексте (среди других). Однако, чтобы не было путаницы с пустым множеством $\varnothing$, мы будем избегать использования этих букв в этом параграфе.

Пример 3.3. Покажем, что $|\mathbf{N}| = |\mathbf{Q}|$. Это требует несколько более сложных рассуждений. Вначале рассмотрим счетное количество копий $\mathbf{N}$, каждая из них соответствует своему номеру $n \in \mathbf{N}$. Мы можем записать это множество как $\mathbf{N} \times \mathbf{N}$ и упорядочить его элементы, как указано на рис. 3.3. Такое упорядочивание является

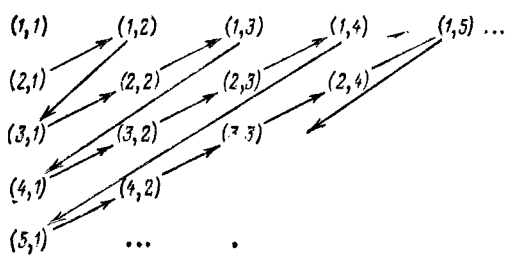



Рис. 3.3

биекцией $\mathbf{N} \times \mathbf{N} \to \mathbf{N}$, задаваемой отношением $(x, y) \mapsto \frac{(x+y-1)(x+y-2)}{2} + y$. Каждый положительный элемент $\mathbf{Q}$ может быть связан с дробью $(p, q)$, где $p$ и $q$ взаимно простые, и связан с элементом $(p, q)$ множества $\mathbf{N} \times \mathbf{N}$ естественным образом. Поэтому, записывая

$$T = \{x\colon x \in \mathbf{Q},\ x > 0\},$$

получаем

$$|T| \leqslant |\mathbf{N} \times \mathbf{N}| = |\mathbf{N}|.$$

(Использование отношения «$\leqslant$» между этими бесконечными числами не обосновано. Соотношение $|A| \leqslant |B|$

следует читать как «$A$ биективно подмножеству $B$». Доказательство неочевидного факта, что это отношение является отношением порядка, лежит за пределами этой книги.) С другой стороны, каждое $p \in \mathbf{N}$ может быть представлено как $p/1$ и, следовательно, связано с парой $(p, 1)$. Поэтому $|\mathbf{N}| \leqslant |\mathbf{T}| \leqslant |\mathbf{N}|$, откуда следует, что $|\mathbf{T}| = |\mathbf{N}|$. Для достижения нашей цели существенно сказать следующее. Возьмем диагонально упорядоченное множество $\mathbf{N} \times \mathbf{N}$ (см. рис. 3.3) и выбросим из него пары, имеющие нетривиальный общий множитель. Это дает метод нумерации элементов $T$, однако трудно дать формулу, которая связывала бы элементы $T$ с элементами $\mathbf{N}$. Теперь мы должны повторить наше рассуждение применительно ко всем рациональным числам. Это можно сделать несколькими способами. Выберем для наглядности следующий. Расширяя уже полученное соответствие между $T$ и $\mathbf{N} \times \mathbf{N}$ до оператора между $\mathbf{Q}$ и $\mathbf{Z} \times \mathbf{N}$, получаем

$$|\mathbf{N}| = |\mathbf{Z}| \leqslant |\mathbf{Q}| \leqslant |\mathbf{Z} \times \mathbf{N}| = |\mathbf{N} \times \mathbf{N}| = |\mathbf{N}|,$$

что дает требуемый результат. Построение биекции между $\mathbf{Z} \times \mathbf{N}$ и $\mathbf{N} \times \mathbf{N}$ оставляем в качестве упражнения. //

Предыдущий пример несколько длинен, однако при его рассмотрении возникло несколько важных моментов, которые мы сейчас отметим.

1. Если $S$ конечно и $\chi\colon S \to S$ — инъективное отображение, тогда $\chi$ биективно.

Доказательство. Пусть выполнены условия утверждения. Если $S = \varnothing$, то требуемый результат тривиален. Если $S \neq \varnothing$, тогда существует биекция $\psi\colon \mathbf{N}_m \to S$ для некоторого $m \in \mathbf{N}$ и отображение $\psi^{-1} \circ \chi \circ \psi$ инъективно: $\mathbf{N}_m \to \mathbf{N}_m$ и, следовательно, является биекцией. (Доказательство этого факта оставляем в качестве упражнения.) Основная идея заключается в переупорядочивании $m$ объектов и известна под названием «принцип раскладывания по гнездам». Даны $m$ гнезд, каждое в своем ящике. Любая схема переселения, при которой в одном ящике может быть не более одного гнезда, должна использовать все $m$ ящиков, т. е. $\psi^{-1} \circ \chi \circ \psi$ является перестановкой $\mathbf{N}_m$ (рис. 3.4). Однако $\psi$ — биекция; следовательно, $\chi \circ \psi$ и $\chi$ также биекции. (Обратно, если $\chi\colon S \to S$ не сюръективно, то $S$ должно быть бесконечным.)

2. Множество $\mathbf{N}$ бесконечно, так как отображение на $\mathbf{N}$, определенное как $n \mapsto n+1$, инъективно, но не биективно (нет элемента, отображающегося в 1). Следовательно, обращая предыдущий результат, получаем, что $\mathbf{N}$ не может быть конечным.

Подмножество $A$ из $\mathbf{R}$ *ограничено сверху* (*снизу*) если существует *верхняя* (*нижняя*) *граница*. $A$ *ограничено*, если оно ограничено сверху и снизу.

3. Ограниченное подмножество из $\mathbf{N}$ конечно.

Доказательство. Каждое подмножество из $\mathbf{N}$ ограничено снизу нулем. Пусть $A \subseteq \mathbf{N}$ ограничено сверху некоторым $m \in \mathbf{N}$. Определим отображение $\chi$: $A \to \mathbf{N}$

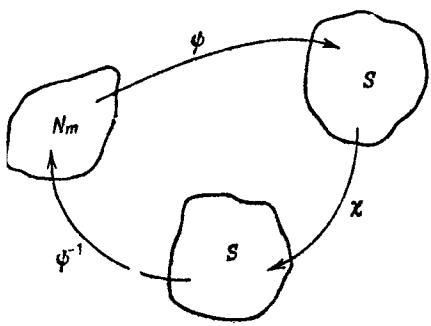



Рис. 3.4

так, что если $A = \{a_1, a_2, \ldots, a_i, \ldots\}$ и $a_1 < a_2 < a_3 < \ldots$ $\ldots < m$ (такой порядок возможен, так как $A \subseteq \mathbf{N}$), то $\chi(a_i) = i$.

Следствием этого является соотношение $\chi(a_i) \leqslant a_i$, и $\chi$, очевидно, инъективно. Оно также должно быть биекцией, т. е. $\chi$: $A \to \mathbf{N}_n$ для некоторого $n \leqslant m$. Если это не так, тогда существует $a_p \in A$ такое, что $\chi(a_p) > m$ и, таким образом, $a_p \geqslant \chi(a_p) > m$. Однако $A$ ограничено $m$; поэтому мы пришли к противоречию. Следовательно, $\chi$ — биекция на $\mathbf{N}_n$ и $A$ конечно.

4. Каждое подмножество конечного множества конечно.

Доказательство. Пусть $A \subseteq B$ и $B$ конечно. Если $B = \varnothing$, то $A = \varnothing$, и утверждение доказано. В противном случае $B \sim \mathbf{N}_m$ для некоторого $m \in \mathbf{N}$. Тогда существует биекция $\chi$: $B \to \mathbf{N}_m$. Применение $\chi$ к $A$ дает подмножество из $\mathbf{N}_m$ и поэтому $\chi(A)$ ограничено. Из случая 3 следует конечность $\chi(A)$. Поэтому, так как $\chi(A)$ биективно с $A$, то $A$ конечно.

5. Прямым следствием случая 4 является тот факт, что любое множество, имеющее бесконечное подмножество, само бесконечно.

Доказывая некоторые неочевидные факты о размерностях множеств Z, Q, N, N × N, ..., разумно задаться вопросом: существуют ли множества, мощность которых больше мощности N? Ответ на этот вопрос утвердительный. Действительно, из данного произвольного множества мы можем (используя диагональную процедуру Кантора; см. ниже) создать множество, мощность которого строго больше мощности N.

Мы не будем рассматривать общий случай, а ограничимся рассмотрением удивительного примера, который носит фундаментальный характер и соответствует целям нашего изложения.

Пример 3.4. Покажем, что

$$|[0, 1[| > |\mathbf{N}|, \quad [0, 1[ = \{x: x \in \mathbf{R}, 0 \leqslant x < 1\}.$$

Доказательство (А. Кантор). Каждое число между 0 и 1 может быть записано в виде бесконечной десятичной дроби $0.d_{n1}d_{n2}d_{n3}\ldots$ Предположим, что эти числа могут быть перенумерованы и что $n$-е число имеет значение, данное выше. Будем конструировать следующее число: возьмем $n$-ю цифру десятичной формы, равную $n$-му числу. Это дает нам число $0.d_{11}d_{22}d_{33}\ldots$ Построим новое число: $0.\delta_{11}\delta_{22}\delta_{33}\ldots$, где каждая цифра $\delta_n$ отличается от соответствующей цифры $d_n$. Тогда это построенное число будет отличаться от каждого числа из первоначального перечня, а именно от $n$-го числа оно будет отличаться $n$-й цифрой. Следовательно, мощность $[0, 1[$ строго больше, чем мощность N, и счетной биекции не существует.

Принцип построения числа $0.\delta_{11}\delta_{22}\delta_{33}\ldots$ в предыдущем доказательстве является наиболее существенным моментом, хотя проницательный читатель может заметить, что существуют задачи, где десятичное представление чисел не единственно. Это встречается тогда, когда представление заканчивается бесконечной последовательностью 0 или 9. (Например, 0.3999... и 0.4000...) Чтобы построенное число $0.\delta_{11}\delta_{22}\delta_{33}\ldots$ отличалось от уже выписанного списка чисел, мы можем оговорить, что $\delta_n$ должны отличаться от 0, $d_n$ и 9. Соответствующие проблемы возникают и в других подобных конструкциях, однако, чтобы не отвлекать внимание от основных идей, в большинстве случаев мы будем их игнорировать.

На самом деле можно показать, что $[0, 1[ \sim \mathbf{R}$. Очевидно, что $\mathbf{R}$ — важное множество. Поэтому его мощность обозначают специальным символом $\aleph_1$ (алеф-один). Рассмотрим теперь способы, с помощью которых можно объединять множества и отношения между мощностями отдельных множеств и мощностью результирующего множества. Первый из них довольно очевиден.

**Теорема.** *Если* $A \sim B$ *и* $C \sim D$, *то* $(A \times C) \sim (B \times D)$.

**Доказательство.** Пусть $\chi\colon A \to B$ и $\psi\colon C \to D$ — биекции. Тогда $(a, c) \mapsto (\chi(a), \chi(c))$ является биекцией между $A \times C$ и $B \times D$. //

**Теорема.** *Если* $Z$ — *конечное множество и* $\{X, Y\}$ — *разбиение* $Z$, *тогда* $|Z| = |X| + |Y|$.

**Доказательство.** Так как $Z$ конечно, то $Z \sim \mathbf{N}_m$ для некоторого $m \in \mathbf{N}$ и существует биекция $\chi\colon Z \to \mathbf{N}_m$ (рис. 3.5). Более того, так как $X \subseteq Z$ и $Y \subseteq Z$, то $\chi(X) \subseteq \mathbf{N}_m$ и $\chi(Y) \subseteq \mathbf{N}_m$. Пусть $\psi_1$ является биекцией из $\chi(X)$ в $\mathbf{N}_{p_1}$ для некоторого $p_1 \leqslant m$, где $(\psi_1 \circ \chi)(X) = \mathbf{N}_{p_1}$. Аналогично пусть $\psi_2$ — биекция из $\chi(Y)$ в $\mathbf{N}_{p_2}$ для некоторого $p_2 \leqslant m$, где $(\psi_2 \circ \chi)(Y) = \mathbf{N}_{p_2}$. Тогда если $\sigma\colon \mathbf{N} \to \mathbf{N}$ определяется как $\sigma\colon x \mapsto x + p_1$, то $\sigma$ является биекцией между $\mathbf{N}_{p_2}$ и $\mathbf{N}_{p_1+p_2} \setminus \mathbf{N}_{p_1}$.

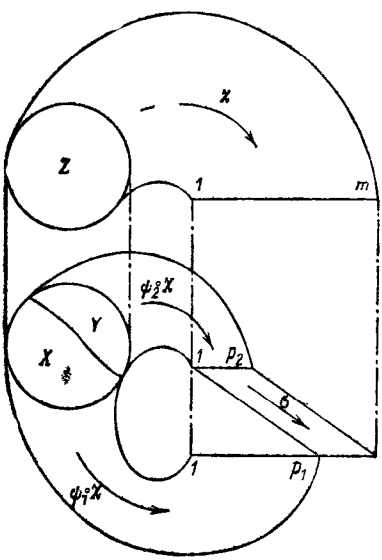


Рис. 3.5

Следовательно, отображение

$$z \mapsto \begin{cases} \psi_1 \circ \chi(z), & \text{если } z \in X, \\ \sigma \circ \psi_2 \circ \chi(z), & \text{если } z \in Y, \end{cases}$$

является инъекцией и биективно (так как $Z$ конечно) между $Z$ и $\mathbf{N}_{p_1+p_2}$. Таким образом, $Z \sim \mathbf{N}_{p_1+p_2}$. Следовательно, $m = p_1 + p_2$ и $|Z| = |X| + |Y|$.

**Пример 3.5.** Чтобы проиллюстрировать применение предыдущей теоремы, рассмотрим случай, когда

$$Z = \{a, b, c, d, e, f\},$$

$$X = \{b, e\},\ Y = \{a, c, d, f\},$$

Тогда биекция между $Z$ и $\mathbf{N}_6$ задается следующим образом:

$$\chi = \{(a, 5), (b, 1), (c, 6), (d, 3), (e, 4), (f, 2)\},$$

и поэтому

$$\chi(X) = \{1, 4\}, \chi(Y) = \{2, 3, 5, 6\}.$$

Подходящими отображениями $\psi_i$ являются

$$\psi_1 = \{(1, 2), (4, 1)\}, \psi_2 = \{(3, 1), (5, 2), (6, 3), (2, 4)\}.$$

Поэтому если $\sigma$ определяется как $x \mapsto x + 2$, то

$$\psi_1 \circ \chi(X) = \{1, 2\},$$
$$\psi_2 \circ \chi(Y) = \{1, 2, 3, 4\},$$
$$\sigma \circ \psi_2 \circ \chi(Y) = \{3, 4, 5, 6\}.$$

Комбинирование $\psi_1 \circ \chi$ и $\sigma \circ \psi_2 \circ \chi$ дает нужный результат. //

Предыдущий пример не является характерным в том смысле, что он требует очень тщательных манипуляций с биекциями. Обычно в этом нет необходимости, так как почти всегда можно сослаться на хорошо известные результаты.

Закончим обсуждение следующим основным результатом. Если $A$ и $B$ — конечные множества, то $A \times B$ конечно и $|A \times B| = |A| * |B|$. Этот результат не только интересен сам по себе, но и дает способ для введения *доказательства по индукции.*

Доказательство по индукции использует два основных понятия, содержащихся в определении N:

(I) Существует некоторый начальный элемент (в N это 1).

(II) Для заданного утверждения, соответствующего некоторому элементу, существует метод, позволяющий перейти к следующему (в случае N это — создание следующего числа за наибольшим до сих пор числом, включенным в N).

Более конкретно, если для некоторого $n_0 \in \mathbf{N}$ (обычно $n_0 = 1$, но не обязательно) мы можем доказать, что утверждение $P(n_0)$ справедливо и для любого $n \in \mathbf{N}$ $(n \geqslant n_0)$ справедливость $P(n)$ влечет справедливость $P(n+1)$ (здесь $n+1$ — следующий за $n$ элемент), то заключаем, что $P(n)$ справедливо для всех $m \geqslant n_0$. Шаг (I) является *основанием индукции*, шаг (II) — *шагом индукции.*

Весь процесс, по существу, является прямым доказательством

$$P(n_0) \Rightarrow \ldots \Rightarrow P(m)$$

и, следовательно, осуществляет непосредственную проверку промежуточных результатов.

Рассмотрим следующий пример.

Пример 3.6. Если $A$ и $B$ конечны, то

$$|A \times B| = |A| * |B|.$$

Доказательство. Поскольку $A$ и $B$ конечны, то $A \sim \mathbf{N}_m$ с биекцией $\tau$: $A \to \mathbf{N}_m$ и $B \sim \mathbf{N}_n$ для некоторых $m, n \in \mathbf{N}$. Будем использовать индукцию по $n$ — размерности $B$. Заметим, что от размерности $A$ требуется конечность, так как мы предполагаем знакомство только с умножением конечных величин.

*Основание индукции.* Если $B = \varnothing$, то $A \times B = \varnothing$, и поэтому имеем тривиальное равенство

$$|A \times B| = 0 = |A| * 0 = |A| * |B|.$$

Если $B = \{b\}$, то отображение $A \to A \times B$ такое, что $a \mapsto (a, b)$, очевидно, биективно, и поэтому

$$|A \times B| = |A| = |A| * 1 = |A| * |B|.$$

*Шаг индукции.* Предположим, что

$$|A \times B_k| = |A| * |B_k|,$$

где $B_k \subseteq B$ и $|B_k| = k \in \mathbf{N}$. Тогда

$$|A| * |B_k| = m * k \in \mathbf{N}$$

и существует биекция $\psi$: $A \times B_k \to \mathbf{N}_{m*k}$. Если $k < n$, то можно взять подмножество $B$, которое имеет $j = k + 1$ элементов. Пусть $B_j = B_k \cup \{x\}$, где $B_k$ — множество из $k$ элементов, и пусть отображение $\chi$, $\chi$: $A \times B_j \to \mathbf{N}$ определяется следующим образом:

$$\chi\colon (a, x) \mapsto \tau(a) + m*k,$$

$$\chi\colon A \times \{x\} \to \{m * k + 1, \ldots, m * k + m\},$$

$$\chi\colon (a, b) \mapsto \psi(a, b), \text{ если } b \in B_k.$$

Очевидно, что $\chi$ является биекцией на $\mathbf{N}_{m*k+m}$ и

$$m * k + m = m * (k + 1) = m * j.$$

Поэтому $|A \times B_j| = m * j = |A| * |B_j|$. Следовательно, тождество справедливо для всех подмножеств $B$, содержащихся в $B$, и поэтому $|A \times B| = |A| * |B|$. //

Упражнение 3.3.

1. Доказать, что отношение между множествами

$$\{(A, B): A \sim B\}$$

является отношением эквивалентности.

2. Построить биекцию между множествами $\mathbf{Z} \times \mathbf{N}$ и $\mathbf{N} \times \mathbf{N}$.

3. Доказать, что если $A$ и $B$ — множества и $A \cup B$ конечно, то

$$|A \cup B| + |A \cap B| = |A| + |B|.$$

4. Доказать (от противного), что произвольная инъекция $\mathbf{N}_m \to \mathbf{N}_m$ ($m \in \mathbf{N}$) является биекцией.

5. Доказать (без использования равенства $|\mathscr{P}(A)| = 2^{|A|}$), что для любого конечного множества $A$, такого что $|A| \geqslant 2$, справедливо соотношение $|\mathscr{P}(A)| > |A|$.

6. Показать, что результат задачи 5 имеет место и для бесконечных множеств.

7. Пусть множество $\{0, 1\}^{\mathbf{N}}$ представляет собой последовательность $a_1, a_2, a_3, \ldots, a_n, \ldots$, где $a_i \in \{0, 1\}$. Доказать, что $|\{0, 1\}^{\mathbf{N}}| > |\mathbf{N}|$.

8. Доказать (построением подходящих биекций и рассуждений по индукции), что если $A_1, \ldots, A_n$ — конечные множества, то

$$|A_1 \times \ldots \times A_n| = |A_1| * \ldots * |A_n|.$$

## § 4. Некоторые специальные классы функций

В этом параграфе мы несколько отойдем от основной темы обсуждения для того, чтобы кратко рассмотреть следующие четыре важных класса функций: подстановки, последовательности, функционалы и отображения, сохраняющие эквивалентность. Эти функции часто используются; особо отметим их приложения к теории графов, к трассировке вычислений, к определению языков программирования и переводу, к машинной графике.

Определение. *Подстановкой* множества $A$ называется биекция на $A$. //

*Подстановки* конечных множеств представляют особый интерес в вычислениях. Когда $A$ конечно, мы в состоянии вычислить число различных подстановок $A$.

Пусть $|A| = n \in \mathbf{N}$. Обозначим через ${}_nP_n$ число таких подстановок. Значение ${}_nP_n$ легко вычислить. Можно рассматривать задачу построения биекции на $A$ как задачу

заполнения ящиков, пронумерованных от 1 до $n$ (рис. 3.6), объектами $a_1, \ldots, a_n$. Порядок, в котором заполняются ящики, несуществен (любой другой порядок можно получить перемешиванием ящиков). Поэтому будем заполнять их слева направо. Первый ящик может быть заполнен $n$ способами, так как мы имеем свободный выбор из всего множества $A$. Убирая выбранный


Рис. 3.6

элемент из $A$, получим множество из $n-1$ элементов. Следовательно, второй ящик может быть заполнен $n-1$ способами, третий ящик — $n-2$ способами и т. д. Продолжая этот процесс, получим, что $(n-1)$ й ящик может быть заполнен двумя способами, а ящик с номером $n$ — единственным оставшимся элементом из $A$. Следовательно, число различных подстановок из $A$ равно

$$n*(n-1)*(n-2)*\ldots*3*2*1.$$

Это произведение называется *факториалом* $n$ (обозначается $n!$). Следовательно, ${}_nP_n = n!$.

Так как $A \sim \mathbf{N}_n$, то можно свести наше рассмотрение к $\mathbf{N}_n$. Любая подстановка на $\mathbf{N}_n$ должна определять образ каждого элемента в $\mathbf{N}_n$ (который, конечно, должен быть единственным и отличным от других). Пусть $\psi$ — подстановка на $\mathbf{N}_n$. Тогда $\psi$ можно определить как множество из $n$ пар следующим образом:

$$\psi = \{(1, x_1), (2, x_2), \ldots, (n, x_n)\},$$

где

$$\{x_1, \ldots, x_n\} = \mathbf{N}_n.$$

Не обязательно, конечно, должно быть $x_1 = 1$ и т. д. Можно также представить $\psi$ следующим образом:

$$\psi = \begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ x_1 & x_2 & x_3 & \ldots & x_n \end{pmatrix}.$$

Пример 4.1. Пусть $\sigma$ — подстановка на $\mathbf{N}_6$:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 6 & 3 & 1 & 4 & 2 \end{pmatrix}.$$

Тогда $\sigma(1) = 5$, $\sigma(3) = 3$ и т. д. //

Достоинством этого обозначения является простота, с которой могут быть вычислены сложные подстановки. Предположим, что $\psi$ — подстановка на $\mathbf{N}_n$, определенная выше, а $\chi$ — другая подстановка на том же самом множестве. Тогда подстановка $\chi$ может быть записана как совокупность пар в порядке, определяемом $x_1, x_2, \ldots, x_n$. Если две последовательности записать одну над другой (первая применяемая подстановка должна быть записана первой), то верхняя и нижняя строки дадут результирующую подстановку.

Пример 4.2. Пусть $\sigma$ — подстановка из примера 4.1 и

$$\rho = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 6 & 1 & 4 & 5 \end{pmatrix}.$$

Можно переписать $\rho$ в виде

$$\rho = \begin{pmatrix} 5 & 6 & 3 & 1 & 4 & 2 \\ 4 & 5 & 6 & 3 & 1 & 2 \end{pmatrix}.$$

Поэтому $\rho \circ \sigma$ может быть вычислено следующим образом:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 6 & 3 & 1 & 4 & 2 \end{pmatrix}$$

$$\rho = \begin{pmatrix} 5 & 6 & 3 & 1 & 4 & 2 \\ 4 & 5 & 6 & 3 & 1 & 2 \end{pmatrix} \quad \text{одинаковые}$$

$$\rho \circ \sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 3 & 1 & 2 \end{pmatrix}$$

Следовательно, например,

$$\rho \circ \sigma(2)\,(= \rho(\sigma(2)) = \rho(6)) = 5 \text{ и т. д. } //$$

Отсюда следует, что представление обратной (конечной) подстановки получается перестановкой строк, представляющих исходную подстановку. Хотя такое представление полезно в вычислениях, оно требует много лишнего места, особенно в тех случаях, когда многие элементы не меняются в процессе подстановки. Существует более простое обозначение, которое может употребляться непосредственно для некоторых простых подстановок и косвенно для всех конечных.

Определение. Пусть $A = \{a_1, \ldots, a_n\}$. Подстановку $\rho$ называют *циклом* (*циклической подстановкой*), если

$$\rho = \begin{pmatrix} a_1 & a_2 \ldots a_{n-1} & a_n \\ a_2 & a_3 \ldots a_n & a_1 \end{pmatrix},$$

Предположим, что $A \subseteq B$ и $B$ конечно. Расширяя $\rho$ на все $B$, можно определить подстановку $\sigma$ так, что

$$\sigma\colon x \mapsto \begin{cases} \rho(x), & \text{если } x \in A, \\ x, & \text{если } x \in B \setminus A. \end{cases}$$

В этом случае $\sigma$ ведет себя подобно $\rho$ во всех случаях, когда элементы $B$ не остаются на месте. Применение $\sigma$ к $A$ передвигает элементы по кругу циклическим образом, и, если известна область $A$, мы можем обозначить подстановку как $(a_1, a_2, \ldots, a_n)$. Эта подстановка называется *циклом длины n.* //

Пример 4.3. Рассмотрим опять подстановку

$$\rho = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 6 & 1 & 4 & 5 \end{pmatrix}.$$

Подстановка является циклом длины 5 и может быть записана как

$$(1, 3, 6, 5, 4). \text{ //}$$

Не все подстановки являются циклами. Например, подстановка $\sigma$ в примере 4.1 не является циклом. Напомним, что $\sigma$ имела вид

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 6 & 3 & 1 & 4 & 2 \end{pmatrix}.$$

Поэтому $\sigma(1) = 5$, $\sigma(5) = 4$, $\sigma(4) = 1$, откуда следует, что $\sigma$ содержит цикл $(1, 5, 4)$. Начиная с 2, получаем другой цикл $-(2, 6)$. Таким образом, имеем $\sigma = (1, 5, 4) \circ (2, 6)$ и $\sigma = (2, 6) \circ (1, 5, 4)$.

В действительности любая конечная подстановка может быть представлена как произведение циклов, при этом циклы могут располагаться в любом порядке. Из построения следует, что один элемент не может встретиться более чем в одном цикле, т. е. циклы *не пересекаются.*

Теорема. *Каждая подстановка $\rho$ на конечном множестве $A$ выражается в виде произведения непересекающихся циклов.*

Доказательство. Поскольку $|A| = n \in \mathbf{N}$, то $A \sim \mathbf{N}_n$. Поэтому без потери общности мы можем ограничиться рассмотрением подстановки $\rho$ на $\mathbf{N}_n$.

В теореме утверждается, что $\rho = \sigma_1 \circ \sigma_2 \circ \ldots \circ \sigma_r$, где каждое $\sigma_i$ является циклом и циклы не пересекаются. Для доказательства теоремы построим требуемые циклы. Сначала найдем наименьший элемент $x_1 \in \mathbf{N}_n$ такой, что

$\rho(x_1) \neq x_1$ и $\rho(x) = x$ для всех $x$, $1 \leqslant x < x_1$. Если такого $x_1$ не существует, то $\rho = I$ (т. е. $\rho$ является тривиальным пустым произведением циклов). В противном случае вычислим $x_1$, $\rho(x_1)$, $\rho^2(x_1)$, $\rho^3(x_1)$ и т. д. Все эти элементы находятся в $\mathbf{N}_n$. Поэтому элементы в этой последовательности должны содержать повторения. Предположим, что $\rho^k(x_1)$ — первый такой элемент (который уже повторялся в последовательности). Покажем, что $\rho^k(x_1) = x_1$. Предположим, что это соотношение не выполняется. Тогда $\rho^l(x_1) = \rho^k(x_1)$ для некоторого $l$, $0 < l < k$. Следовательно,

$$\rho^{l-1}(x_1) = \rho^{-1} \cdot \rho^l(x_1) = \rho^{-1} \cdot \rho^k(x_1) = \rho^{k-1}(x_1) \text{ и т. д.}$$

Поэтому $\rho^{l-l}(x_1) = \rho^{k-l}(x_1)$, т. е. $\rho^{k-l}(x_1) = \rho^0(x_1) = x_1$, что противоречит минимальности $k$ (так как $k - l < k$). Таким образом, $\rho^k(x_1) = x_1$, и подстановка

$$\sigma_1 = (x_1, \rho(x_1), \rho^2(x_1), \ldots, \rho^{k-1}(x_1))$$

задает цикл внутри $\rho$.

Если все элементы $x \in \mathbf{N}_n$ такие, что $\rho(x) \neq x$ (будем называть такие элементы *нестационарными*), содержатся в $\sigma_1$, то $\rho = \sigma_1$ — единственный цикл (который, естественно, не пересекается). В противном случае найдем следующий наименьший элемент $x_2 \in \mathbf{N}_n$ такой, что $\rho(x_2) \neq x_2$ и $x_2$ не встречается в $\sigma_1$. Из $x_2$ строим множество различных степеней $\rho$:

$$\sigma_2 = (x_2, \rho(x_2), \rho^2(x_2), \ldots, \rho^m(x_2)).$$

Это цикл длины не менее 2, и он не пересекается с $\sigma_1$. (Доказательство оставляем в качестве упражнения.) Если все нестационарные элементы исчерпаны, то $\rho = \sigma_1 \cdot \sigma_2 = \sigma_2 \cdot \sigma_1$. Очевидно, что множество нестационарных элементов, не входящих в эти циклы, можно уменьшить, и в конце концов придем к $\varnothing$. Следовательно, $\rho = \sigma_1 \cdot \sigma_2 \cdot \sigma_3 \cdot \ldots \cdot \sigma_r$ для некоторого $r \in \mathbf{N}$. //

Рассмотрим теперь несколько иную ситуацию. Возьмем множества $A$: $|A| = n$ и $B \subseteq A$, $|B| = r \leqslant n$. Возникает вопрос: сколько биективных функций существует из $A$ в $B$? Или, что эквивалентно, сколько существует инъективных отображений из $B$ в $A$? Число перестановок (без повторений) из $n$ элементов по $r$ обозначается ${}_nP_r$ и вычисляется так же, как и ${}_nP_n$, за исключением того факта, что процесс прекращается после заполнения $r$ ящиков. Таким образом,

$$P_r = n \cdot (n-1) \cdot \ldots \cdot (n-r+1),$$

Легко видеть, что, продолжая процесс заполнения ящиков, оставшиеся $n-r$ элементов можно разместить по последним $n-r$ ящикам ${}_{n-r}P_{n-r}$ способами. Поэтому и

$$ {}_nP_r = \frac{{}_nP_n}{{}_{n-r}P_{n-r}} = \frac{n!}{(n-r)!}. $$

При вычислении ${}_nP_r$ мы находим число биективных функций из $A$ в $B$. Подсчитаем числа таких функций.

Определение. Пусть $A$ — конечное множество и $B \subseteq A$, $|A| = n \geqslant r = |B|$. Множество $B$ называется *сочетанием* (без повторений) из $n$ элементов по $r$. Число таких сочетаний обозначается через $C_n^r$. //

Вычисление $C_n^r$ производится следующим образом. Положим $|A| = n$. Возьмем произвольное подмножество $B \subseteq A$ такое, что $|B| = r$. Тогда $B$ является образом подстановки из $n$ элементов по $r$. Число инъективных функций на $A$, имеющих $B$ своим образом, есть ${}_nP_r$. Если $f$ является такой функцией и $g$ — другая такая функция, имеющая ту же самую область значений, то $g$ связано с $f$ соотношением $g = \varphi \circ f$, где $\varphi$ — подстановка на $B$. Функции $g$ и $f$ определяют одну и ту же комбинацию, и в действительности число функций, определяющих эту комбинацию, равно числу подстановок $\varphi$ на $B$. Следовательно,

$$ {}_nP_r = C_n^r * {}_rP_r, $$

откуда

$$ C_n^r = \frac{{}_nP_r}{{}_rP_r} = \frac{n!}{r!\,(n-r)!}. $$

Поскольку относительные дополнения единственны и $|A \setminus B| = n - r$, то отсюда следует, что $C_n^r = C_n^{n-r}$.

Вернемся теперь к математическим объектам, которые должны быть знакомы читателю, но которые, вероятно, не рассматривались как функции.

Определение. *Последовательностью* на множестве $S$ называют отображение $\mathbf{N} \to S$.

Если $\sigma$: $\mathbf{N} \to S$ — заданная последовательность и $\sigma(n) = s_n$, то обычно обозначают последовательность не $\sigma$, а $(s_n)$ или $(s_1, s_2, \ldots, s_n, \ldots)$. В этом случае $s_n$ называют $n$-м членом последовательности. //

Часто при изучении свойств последовательностей возникает понятие «расстояние» между соседними элементами последовательности (скажем, $s_n$ и $s_{n+1}$) и между элементами $s_n$ при $n \geqslant n_0$ (где $n_0$ — некоторый фиксированный элемент $\mathbf{N}$) и фиксированным элементом из $S$.

Мы вернемся к этим вопросам несколько позже, поскольку в настоящий момент у нас в общем случае нет понятия расстояния.

Особый интерес в задачах, связанных с трансляцией языков программирования, представляют функционалы. Существуют функции, которые определены не на множествах простых объектов (таких, как числа), а, например, на множестве функций.

Определение. Пусть даны множества $A$, $B$ и $C$. Обозначим через $[B \to C]$ множество всех функций из $B$ в $C$. Функция $f$: $A \to [B \to C]$ называется *функционалом*. Следовательно, $a \in \mathscr{D}_f \Rightarrow f(a)$ — функция, и $f(a)$: $B \to C$. Далее, $b \in \mathscr{D}_{f(a)} \Rightarrow f(a)(b) \in C$. //

На первый взгляд такая комбинация, как $f(a)(b)$, выглядит настолько необычной, что может казаться незаконной. Это в основном потому, что рассматриваемые функции использовались как объекты особого рода, в ряде случаев отличные от элементов, которые были в области определения и области значения функций. Конечно, множества функций могут рассматриваться так же, как и любые другие множества.

В развитых языках программирования имена целых переменных не отличаются от имен переменных функций и могут изучаться аналогичными способами. Хотя эти функции являются обычно довольно сложными, языки программирования редко дают примеры важности функционалов.

Пример 4.4. Пусть $P$ — множество программ, т. е. текстов программ (строк символов), которые должны быть обработаны компилятором. Аналогично пусть $I$ и $O$ — множества соответственно входных и выходных значений, которые доступны программе для ввода и вывода. Тогда компилятор (с соответствующего языка) является функционалом типа $P \to [I \to O]$; для данной $p \in P$ он пытается создать машинный код, который при выполнении будет читать $i \in I$ и выдавать $o \in O$. Из дальнейшего изложения будет видно, что определение функций (или функционалов) сложности компилятора длинное и сложное. Тем не менее эти понятия можно изучать в более простых ситуациях.

Пример 4.5. Пусть все данные принадлежат R. Тогда, если $f$: $a \mapsto [x \mapsto a + x]$, то

$$f(2)\text{: } x \mapsto 2 + x \text{ и } f(2)(3) = 5,$$

то время как $f(3)$: $x \mapsto 3 + x$ и $f(3)(3) = 6$ и т. д. //

Обращение с функционалами не вызывает трудностей при условии, что ссылка делается на основной функционал (т. е. $A \to B$ или $A \to [B \to C]$). Следовательно, в дальнейшем мы будем рассматривать их просто как функции, имеющие нетривиальные области значений, и будем обращаться с ними соответствующим образом.

В заключение параграфа определим функции, которые сохраняют некоторые структуры. Из дальнейшего будет видно, что в некоторых ситуациях желательно сохранить многие из алгебраических свойств, которыми множества могут обладать. Ограничимся вначале рассмотрением простейшего случая.

Определение. Пусть $X$ — множество, на котором задано отношение эквивалентности $\rho$. Тогда $X$ *разбивается отношением* $\rho$ *на* $\rho$*-эквивалентные классы*; множество классов обозначается как $X/\rho$.

Определение. Пусть $X$ и $Y$ — множества, а $\rho_X$ и $\rho_Y$ — отношения эквивалентности на них, и пусть $f\colon X \to Y$ — отображение. Обозначим через $\hat{f}$ *отношение*

$$\hat{f}\colon X/\rho_X \to Y/\rho_Y$$

такое, что

$$\hat{f} = \{([x], [f(x)])\colon x \in X\},$$

где $[x]$ — класс эквивалентности $x$. Если $\hat{f}$ — функция, то

$$x_1 \rho_X x_2 \Rightarrow \hat{f}([x_1]) = \hat{f}([x_2]),$$

и $f$ является *отображением, сохраняющим эквивалентность*. В этом случае говорят, что $f\colon X \to Y$ *индуцирует* отображение

$$\hat{f}\colon X/\rho_X \to Y/\rho_Y. \ //$$

Наглядный способ представления такого отображения дан на рис. 3.7.

Если рассмотреть отображение $f$, согласованное с отношением эквивалентности, то можно переходить от $x_1$ к $y_1$ или через $x_2$, используя соотношения $y_2 = f(x_2)$ и $x_2 \rho_X x_1$, или через $y_1$, используя соотношения $y_1 = f(x_1)$ и $y_2 \rho_Y y_1$.

Пример 4.6. Пусть $X = \{1, 2, 3\}$, $Y = \{1, 4, 9\}$, и пусть $\rho_X$ и $\rho_Y$ таковы, что

$$X/\rho_X = \{\{1\}, \{2, 3\}\}, \quad Y/\rho_Y = \{\{1\}, \{4, 9\}\},$$

и $f\colon X \to Y$ такое, что $x \mapsto x^2$. Тогда

$$\widehat{f}([1]) = [f(1)] = [1] = \{1\},$$
$$\widehat{f}([2]) = [4] = \{4, 9\},$$
$$\widehat{f}([3]) = [9] = \{4, 9\}.$$

В этом случае $\{2, 3\} \in X/\rho_X \Rightarrow 2\rho_X 3 \Rightarrow [2] = [3]$ и $\widehat{f}([2]) = \widehat{f}([3])$. Поэтому $\widehat{f}$ является функцией и $f$ сохраняет отношения эквивалентности. //

Пример 4.7. Пусть $X$, $Y$ и $f$ те же, что и раньше, и отношение эквивалентности $\sigma_X$ и $\sigma_Y$ индуцируют разбиения $\{\{1\}, \{2, 3\}\}$ и $\{\{1, 4\}, \{9\}\}$ соответственно. В этом случае индуцированные отношения дают

$$\widehat{f}([2]) = [f(2)] = [4] = \{1, 4\},$$
$$\widehat{f}([3]) = [f(3)] = [9] = \{9\}.$$

Так как $2\sigma_X 3$, то $[2] = [3]$ в $X/\sigma_X$, но $(4, 9) \notin \sigma_Y$, поскольку $[4] \neq [9]$ в $Y/\sigma_Y$. По сравнению с рис. 3.7 этот пример дает отношения, показанные на рис. 3.8. Так как

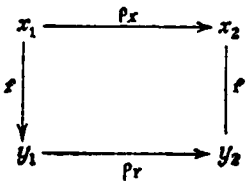



Рис. 3.7

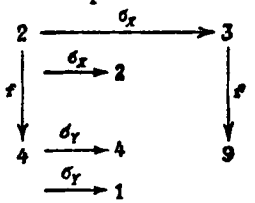



Рис. 3.8

нельзя соединить стороны прямоугольника во всех случаях, то отношения эквивалентности не сохраняются. //

В гл. 5 и далее будет показано, как эти диаграммы могут быть использованы для определения операций таким образом, чтобы соединить углы прямоугольника. После этого можно будет объединять диаграммы подобно строительным блокам.

## § 5. Аналитические свойства вещественных функций

Этот параграф содержит материал, использующий теорию множеств из гл. 1. Цель, которая при этом преследуется, состоит не в развитии техники вычислений, а в создании строгих утверждений типа:

«Предел $f(x)$ при $x$, стремящемся к 0, есть $y$»,

«Наклон графика $f$ в точке $a$ равен $b$»,

«$f$ имеет гладкий график» и т. п.

(Два последних понятия, очевидно, относятся к графике.) Мы дадим основные определения, которые используются при получении некоторых результатов. Этого достаточно для того, чтобы проиллюстрировать доказательства большинства теорем.

**5.1. Последовательности.** *Вещественной последовательностью* называется отображение $\mathbf{N}$ на $\mathbf{R}$. Последовательность записывают в виде $(a_n)$. Если при возрастании $n$ члены $a_n$ становятся «близкими» к некоторому фиксированному значению $a \in \mathbf{R}$, то говорят, что последовательность $(a_n)$ имеет предел $a$ или что $a_n$ стремится к $a$ при стремлении $n$ к бесконечности. Дадим строгое определение сказанному.

Определение. Если $(a_n)$ — вещественная последовательность и для любого $\varepsilon > 0$ существует $N_\varepsilon \in \mathbf{N}$ такое, что $N > N_\varepsilon \Rightarrow |a_n - a| < \varepsilon$, то говорят, что $(a_n)$ имеет *предел* $a$, и записывают это как $\lim\limits_{n\to\infty} a_n = a$ или $a_n \to a$ при $n \to \infty$. (Здесь $|x|$ обозначает модуль числа $x \in \mathbf{R}$.)

Если $(a_n)$ имеет предел, то говорят, что последовательность *сходится*. Если последовательность не имеет предела, то говорят, что она *расходится*.

Пример 5.1.

1. Последовательность $(a_n)$, где $a_n = 1/n$, имеет предел 0; для $\varepsilon > 0$ можно выбрать $N_\varepsilon$ — любое натуральное число, большее $1/\varepsilon$. Тогда

$$N > N_\varepsilon \Rightarrow |a_N - 0| = 1/N < 1/N_\varepsilon < \varepsilon;$$

следовательно,

$$\lim_{n\to\infty} \frac{1}{n} = 0.$$

2. Последовательность $(a_n)$, где $a_n = (-1)^n$, расходящаяся.

Предложение. *Если $(s_n)$ и $(t_n)$ — последовательности и $\lambda \in \mathbf{R}$, тогда $(s_n + t_n)$, $(s_n t_n)$ и $(\lambda s_n)$ также являются последовательностями, и если $\lim\limits_{n\to\infty} s_n = s$ и $\lim\limits_{n\to\infty} t_n = t$, то:*

а) $\lim\limits_{n\to\infty} (s_n + t_n) = s + t$;

б) $\lim\limits_{n\to\infty} (s_n t_n) = st$;

в) $\lim\limits_{n\to\infty}(\lambda s_n) = \lambda s$;

г) *если* $t \neq 0$, *то* $s_n/t_n \to s/t$ *при* $n \to \infty$.

Доказательство. Пусть $\varepsilon > 0$. Тогда существует $N_\varepsilon \in \mathbf{N}$ такое, что

$$|s_N - s| < \varepsilon/2 \text{ и } |t_N - t| < \varepsilon/2$$

при $N > N_\varepsilon$. Так как при $N > N_\varepsilon$

$$|s_N + t_N - (s+t)| = |s_N - s + t_N - t| \leqslant$$
$$\leqslant |s_N - s| + |t_N - t| < \varepsilon,$$

то $\lim\limits_{n\to\infty}(s_n + t_n) = s + t$. Аналогично для случая б)

$$|s_N t_N - st| = |s_N t_N - s_N t + s_N t - st| \leqslant$$
$$\leqslant |s_N t_N - s_N t| + |s_N t - st| \leqslant |s_N|\,|t_N - t| + |s_N - s|\,|t|.$$

Пусть задано $\varepsilon > 0$. Тогда существует $N_\varepsilon \in \mathbf{N}$ такое, что для $N > N_\varepsilon$ справедливы неравенства

$$|s_N - s| < \frac{1}{2}\frac{\varepsilon}{|t|+1},$$

$$|t_N - t| < \frac{1}{2}\frac{\varepsilon}{|s|+1},\quad |s_N| < |s| + 1.$$

Следовательно,

$$|s_N|\,|t_N - t| + |s_N - s|\,|t| \leqslant (|s| + 1)\,|t_N - t| +$$
$$+ |s_N - s|\,|t| < \frac{1}{2}\varepsilon + \frac{1}{2}\frac{\varepsilon}{|t|+1}|t| < \frac{1}{2}\varepsilon + \frac{1}{2}\varepsilon = \varepsilon,$$

откуда получаем $(s_n t_n) \to st$. Доказательство случаев в), г) предложения оставляем в качестве упражнения. //

Определение. Пусть $(a_n)$ — последовательность в $\mathbf{R}$. Последовательность $s_n = \sum\limits_{i=1}^{n} a_i$ определяет *ряд* $\sum a_n$. При этом $s_n$ называют *n-й частичной суммой ряда*. Если последовательность $(s_n)$ сходится, то говорят, что ряд *сходящийся*, и число $\lim\limits_{n\to\infty} s_n$ называют *суммой ряда*. Оно обозначается

$$\sum_{n=1}^{\infty} a_n. \;//$$

**5.2. Непрерывность.** Понятие непрерывности почти полностью игнорируется при изучении элементарных вычислений. В неформальной математике это понятие счи-

тается очевидным. Однако при начальном изучении математики достаточно трудно найти нужный путь. На самом деле определение непрерывности базируется на понятии предела. В этом параграфе через $I$ будем обозначать интервал действительной оси $\mathbf{R}$. Если $f\colon I \to \mathbf{R}$ и $f(x)$ становится «неограниченно близким» к некоторому числу $b$ при $x$, «приближающемся» к $a \in I$, то говорят, что предел $f(x)$ при $x$, стремящемся к $a$, есть $b$. Дадим строгое определение этого понятия.

Определение. Функция $f\colon I \to \mathbf{R}$ имеет *предел* $b$ в точке $a$, если для любого $\varepsilon > 0$ существует $\delta_\varepsilon > 0$ такое, что

$$0 < |x - a| < \delta_\varepsilon \Rightarrow |f(x) - b| < \varepsilon.$$

В этом случае будем писать

$$\lim_{x \to a} f(x) = b$$

или

$$f(x) \to b \quad \text{при} \quad x \to a. \ //$$

Заметим, что в определение не входит значение $f(x)$ в точке $a$.

Пример 5.2.

1. $\lim\limits_{x \to a} x = a$; достаточно выбрать $\delta_\varepsilon = \varepsilon$.
2. $\lim\limits_{x \to 2} x^2 = 4$, поскольку

$$|x - 2| < \delta_\varepsilon \Leftrightarrow -\delta_\varepsilon < x - 2 < \delta_\varepsilon \Leftrightarrow 4 - \delta_\varepsilon < x + 2 < 4 + \delta_\varepsilon.$$

Следовательно,

$$|x^2 - 4| = |x + 2|\,|x - 2| < \delta_\varepsilon |x + 2| < \delta_\varepsilon (4 + \delta_\varepsilon).$$

Если выбрать $\delta_\varepsilon = \min(1, \varepsilon/5)$, то

$$|x^2 - 4| < 5\delta_\varepsilon \leqslant \varepsilon. \ //$$

Легко показать эквивалентность следующих утверждений:

$$\lim_{x \to a} f(x) = b, \qquad \lim_{x \to a} (f(x) - b) = 0,$$

$$\lim_{h \to 0} f(a + h) = b, \quad \lim_{h \to 0} (f(a + h) - b) = 0.$$

Теперь мы готовы к изучению понятия непрерывности для вещественных функций. Грубо говоря, функция $f\colon I \to \mathbf{R}$ непрерывна в точке $a \in I$, если точки, «близкие» к $a$, отображаются в точки, «близкие» к $f(a)$. Бо-

лее строго это понятие может быть определено следующим образом.

Определение. Функция $f\colon I \to \mathbf{R}$ *непрерывна* в $a \in I$, если

$$\lim_{x \to a} f(x) = f(a).$$

Говорят, что $f(x)$ *непрерывна*, если она непрерывна в каждой точке своей области определения. //

Из определения видно, что в данном случае требуется, чтобы $f(x)$ была определена при $x = a$. Такое определение непрерывности соответствует интуитивному представлению. Поясним его на рисунках.

На рис. 3.9, *а* дан график непрерывной функции $f_1$: $[-2, 2] \to \mathbf{R}$, $f_1(x) = |x|$. На рис. 3.9, *б* представлен график функции $f_2$: $[0, 4] \to \mathbf{R}$, где

$$f_2(x) = \begin{cases} x & \text{при } 0 \leqslant x \leqslant 2, \\ x + 1 & \text{при } 2 < x \leqslant 4. \end{cases}$$

Функция $f_2$ непрерывна в каждой точке $[0, 4]$, *за исключением* точки $x = 2$, так как не существует интервала вида $2 - \delta < x < 2 + \delta$, для которого $|f_2(x) - f_2(2)| < 1$.

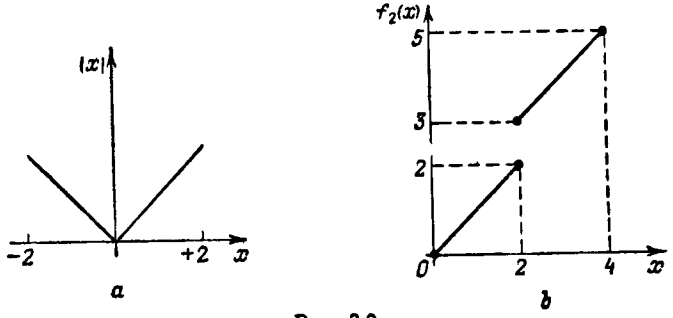



Рис. 3.9

В заключение этого раздела сформулируем без доказательства несколько утверждений. Их доказательство можно найти в большинстве книг по математическому анализу. Если $f(x) \to l$ и $g(x) \to m$ при $x \to a$, то

$$(f + g)(x) = f(x) + g(x) \to l + m,$$
$$(fg)(x) = f(x)g(x) \to lm,$$
$$(f/g)(x) = f(x)/g(x) \to l/m \text{ при условии } m \neq 0,$$
$$(\lambda f)(x) = \lambda f(x) \to \lambda l \text{ для всех } \lambda \in \mathbf{R}.$$

Отсюда следует, что если $f$ и $g$ непрерывны в точке $a$, то непрерывными являются и функции $\lambda f$, $f+g$, $fg$, $f/g$ при условии, что $a$ находится в области определения каждой из «новых» функций.

**5.3. Дифференцируемость.** Графическое представление функции $f\colon I \to \mathbf{R}$, обсуждаемое в п. 5.2, предполагало, что эта функция определяет некоторую другую функцию $f'\colon I \to \mathbf{R}$, где $f'(a)$ есть «наклон» графика $f$ в точке $a$. В общем случае $\mathscr{D}_{f'} \subseteq \mathscr{D}_f$, так как на самом деле его может не быть для всех точек $\mathscr{D}_f$. В этом случае $f'$ не существует. Определим строго функцию $f'$.

Определение. Функция $f\colon I \to \mathbf{R}$ дифференцируема в точке $a \in I$, если

$$\lim_{h\to 0} \frac{f(a+h)-f(a)}{h}$$

существует. Множество точек, где этот предел существует, устанавливает область определения $\mathscr{D}_{f'}$ *производной* $f'\colon I \to \mathbf{R}$ функции $f$, и

$$f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}.$$

Иногда производную $f'$ записывают как $df/dx$. Отношение $\frac{f(x+h)-f(x)}{h}$ часто записывают в виде $\delta f/\delta x$, где $\delta$ читают как «малое приращение». В этих обозначениях

$$\lim_{\delta x\to 0} \frac{\delta f}{\delta x} = \frac{df}{dx}. \quad //$$

Если $f$ дифференцируема в точке $a$, то она и непрерывна в $a$, так как

$$f(a+h)-f(a) = \left(\frac{f(a+h)-f(a)}{h}\right)h;$$

следовательно,

$$\lim_{h\to 0}(f(a+h)-f(a)) =$$

$$= \lim_{h\to 0} \frac{f(a+h)-f(a)}{h} \lim_{h\to 0} h = f'(a)\cdot 0 = 0.$$

Другими словами, $\lim_{h\to 0} f(a+h) = f(a)$, и, таким образом, $f$ непрерывна в $a$ по определению. Поэтому непрерывность функции является необходимым условием ее дифференцируемости, но не достаточным, как показывает следующий пример.

Пример 5.3. Функция $f(x)=|x|$ не дифференцируема в точке $x=0$, так как

$$\frac{f(0+h)-f(0)}{h}=\frac{|h|}{h},$$

где

$$\frac{|h|}{h}=\begin{cases}+1, & \text{если } h>0,\\ -1, & \text{если } h<0.\end{cases}$$

Следовательно, в любом интервале $]-h, h[$, где $h$ произвольное, функция $|h|/h$ принимает оба значения $\pm 1$, и поэтому предел при $h\to 0$ не существует. //

Пример 5.4.

1. Пусть $f\colon \mathbf{R}\to\mathbf{R}$ — постоянная функция, т. е. $f(x)=c$ для всех $x\in\mathbf{R}$. Тогда

$$\frac{f(x+h)-f(x)}{h}=\frac{c-c}{h}=0$$

и $\lim\limits_{h\to 0} 0=0$. Таким образом, $f'(x)=0$ для всех $x\in\mathbf{R}$. Обратно, если $f'(x)=0$ для всех $x\in\mathbf{R}$, тогда $f$ — постоянная.

2. Пусть $f(x)=x^2$ для всех $x\in\mathbf{R}$. Тогда

$$\frac{f(x+h)-f(x)}{h}=\frac{(x+h)^2-x^2}{h}=\frac{x^2+2xh+h^2-x^2}{h}=2x+h,$$

$$\lim_{h\to 0}(2x+h)=\lim_{h\to 0}2x+\lim_{h\to 0}h=2x.$$

Следовательно, $f'(x)=2x$ для всех $x\in\mathbf{R}$. //

Предложение. *Если $f$ дифференцируема в $x$ и $\lambda\in\mathbf{R}$, то $\lambda f$ дифференцируема в $x$ и*

$$(\lambda f)'(x)=\lambda f'(x).$$

Доказательство.

$$(\lambda f)'(x)=\lim_{h\to 0}\frac{\lambda f(x+h)-\lambda f(x)}{h}=\lambda\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}=$$

$$=\lambda f'(x).\ //$$

Следующие результаты оказываются полезными при дифференцировании функций, которые определены через другие функции.

Предложение. *Если $f$ и $g$ дифференцируемы в $x$, то*

а) *$f+g$ дифференцируема в $x$ и*

$$(f+g)'(x)=f'(x)+g'(x);$$

б) *$fg$ дифференцируема в $x$ и*

$$(fg)'(x)=(f'g)(x)+(fg')(x);$$

в) *$f/g$ дифференцируема в $x$ при $g(x)\neq 0$ и*

$$(f/g)'(x)=\frac{(f'g)(x)-(fg')(x)}{g^2(x)}.$$

Доказательство оставляем в качестве упражнения.

Эти формулы могут использоваться при доказательстве некоторых, возможно, знакомых простых результатов.

Пример 5.5.

1. Пусть $f$: $\mathbf{R}\to\mathbf{R}$, где $f(x)=1/x$. Тогда

$$f'(x)=-1/x^2 \quad \text{и} \quad \mathscr{D}_f=\mathscr{D}_{f'}=\mathbf{R}\setminus\{0\}.$$

2. Пусть $f$: $\mathbf{R}\to\mathbf{R}$ задано формулой $f(x)=x^n$ $(n\in\mathbf{N})$. Тогда $f'(x)=nx^{n-1}$. //

Предложение (правило дифференцирования сложной функции). *Если $f$ дифференцируема в $x$ и $g$ дифференцируема в $y=f(x)$, то $g\circ f$ дифференцируема в $x$ и*

$$(g\circ f)'(x)=(g(f(x)))'=g'(y)f'(x).$$

Доказательство. Пусть $w=g(y)=g(f(x))=g\circ f(x)$. Тогда $\frac{d(g\circ f)}{dx}=\lim\frac{\delta w}{\delta x}=\lim\frac{\delta w}{\delta y}\frac{\delta y}{\delta x}$ (при условии $\delta y\neq 0$) $=\lim\limits_{\delta y\to 0}\frac{\delta w}{\delta y}\lim\limits_{\delta x\to 0}\frac{\delta y}{\delta x}$ (см. п. 5.1). Однако $f$ дифференцируема в точке $x$, и поэтому

$$\lim_{\delta x\to 0}\frac{\delta y}{\delta x}=\frac{df}{dx};$$

аналогично

$$\lim_{\delta y\to 0}\frac{\delta w}{\delta y}=\frac{dg}{dy}.$$

Поэтому

$$\frac{d(g\circ f)}{dx}=\frac{dw}{dy}\frac{dy}{dx} \quad \text{и} \quad (g\circ f)'=g'(y)f'(x). \ //$$

Производную от $f'$ записывают в виде $f''$ или $d^2f/dx^2$ и называют второй производной функции $f$. Аналогично производная от $d^{n-1}f/dx^{n-1}$ $(n\geqslant 3)$ записывается как $f^{(n)}$ или $d^nf/dx^n$ и называется $n$-й производной функции $f$. Если $f'$ существует и непрерывна, то говорят, что $f$ принадлежит классу $C^1$; $f$ принадлежит классу $C^n$, если $f^{(n)}$ существует и непрерывна, и классу $C^\infty$, если $f^{(n)}$ существует для всех $n\in\mathbf{N}$.

**5.4. Интегрирование.** Пусть $f\colon [a, b] \to \mathbf{R}$, $n \in \mathbf{N}$, $h = (b-a)/n$ и $x_k = a + kh$ при $0 \leqslant k < n$. Тогда можно определить последовательность $(s_n(f))$:

$$(s_n(f)) = \sum_{k=0}^{n-1} f(x_k)\, h.$$

Если $(s_n(f))$ имеет предел, то будем говорить, что $f$ интегрируема на $[a, b]$, и обозначать

$$\lim_{n\to\infty} s_n(f) = \int_a^b f(x)\, dx.$$

Величину $\int_a^b f(x)\, dx$ называют *интегралом Римана* функции $f(x)$ на $[a, b]$.

Заштрихованная площадь на рис. 3.10 является графическим представлением $s_5(f)$ для непрерывной функции $f$ на $[a, b]$. Для неограниченно больших значений $n$ интуитивно можно ожидать, что заштрихованная площадь будет хорошо аппроксимировать площадь под графиком между $x = a$ и $x = b$ и ограниченным значением (если оно существует) этой площади.

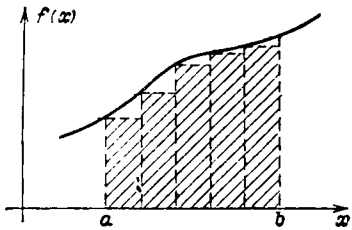


Рис. 3.10

Если $\mathscr{F}$ обозначает множество всех вещественных функций на $[a, b]$, то интегрирование может рассматриваться как функция $\mathscr{F} \to \mathbf{R}$, область определения которой есть

$$\left\{ f \in \mathscr{F}\colon \int_a^b f(x)\, dx \text{ существует} \right\}.$$

Некоторые важные свойства интеграла приводятся ниже.

Предложение.

а) *Если $f$ непрерывна на $[a, b]$, то она интегрируема на этом отрезке*;

б) *если $f$ интегрируема на $[a, b]$ и $x \in [a, b]$, тогда $f$ интегрируема на $[a, x]$ и $[x, b]$ и*

$$\int_a^b f(x)\, dx = \int_a^x f(x)\, dx + \int_x^b f(x)\, dx;$$

в) *если* $f$ *интегрируема на* $[a, b]$ *и* $\lambda \in \mathbf{R}$, *тогда* $\lambda f$ *интегрируема на* $[a, b]$ *и*

$$\int_a^b (\lambda f)(x)\,dx = \lambda \int_a^b f(x)\,dx;$$

г) *если* $f$ *и* $g$ *интегрируемы на* $[a, b]$, *то* $f+g$ *интегрируема на* $[a, b]$ *и*

$$\int_a^b (f+g)(x)\,dx = \int_a^b f(x)\,dx + \int_a^b g(x)\,dx.$$

Доказательство. В случае а) формальное доказательство давать не будем. Заметим, однако, что для непрерывной функции $f$ интуитивно ясно, что площадь под графиком $f$ является хорошо определенным понятием, и, следовательно, можно ожидать, что интеграл от $f$ существует. Доказательства б)—г) следуют из соответствующих свойств последовательностей. Рассмотрим, например, случай г). Если $f$ и $g$ интегрируемы на $[a, b]$, то последовательности

$$s_n(f) = \sum_{k=0}^{n-1} f(x_k)\,h \quad \text{и} \quad s_n(g) = \sum_{k=0}^{n-1} g(x_k)\,h$$

имеют пределы. Рассмотрим последовательность $s_n(f) + s_n(g)$. Тогда

$$\lim_{n\to\infty} (s_n(f) + s_n(g)) = \lim_{n\to\infty} s_n(f) + \lim_{n\to\infty} s_n(g). \;//$$

Чтобы вычислить интеграл, редко используют определение и вычисляют предел. Следующая теорема является основной. (Она устанавливает тот факт, что интегрирование и дифференцирование — взаимно обратные процессы.)

Теорема. *Пусть* $f$: $[a, b] \to \mathbf{R}$ *непрерывна. Определим функцию* $F$: $[a, b] \to \mathbf{R}$ *формулой*

$$F(t) = \int_a^t f(x)\,dx.$$

*Тогда* $F$ *дифференцируема на* $[a, b]$ *и* $F' = f$.

Доказательство. Будем лишь фиксировать основные моменты доказательства. Используя результаты

предыдущего предложения, имеем

$$\frac{F(t+h)-F(t)}{h}=\frac{\int\limits_a^{t+h} f(x)\,dx-\int\limits_a^t f(x)\,dx}{h}=$$

$$=\frac{\int\limits_a^t f(x)\,dx+\int\limits_t^{t+h} f(x)\,dx-\int\limits_a^t f(x)\,dx}{h}=\frac{1}{h}\int\limits_t^{t+h} f(x)\,dx.$$

Рассмотрим

$$\lim_{h\to 0}\frac{1}{h}\int\limits_t^{t+h} f(x)\,dx.$$

Из определения интеграла и его интерпретации как площади ясно, что для малых $h$ интеграл $\int\limits_t^{t+h} f(x)\,dx$ стремится к $f(t)h$ и

$$\lim_{h\to 0}\frac{1}{h}\int\limits_t^{t+h} f(x)\,dx=f(t).$$

Следовательно,

$$F'(t)=f(t) \quad \text{для всех} \quad t\in[a,\ b]. \ //$$

Пусть $\Phi$ — произвольная функция, для которой $\Phi'=f$. Тогда $F-\Phi$ является постоянной, так как

$$F'(t)=f(t)=\Phi'(t).$$

Следовательно,

$$(F-\Phi)'(t)=0 \quad \text{для всех} \quad t\in[a,\ b],$$

и из п. 5.3 заключаем, что $F-\Phi=\lambda$, $\lambda\in\mathbf{R}$. Таким образом,

$$\Phi(t)=\int\limits_a^t f(x)\,dx+\lambda.$$

Функцию $\Phi$ называют *неопределенным* интегралом от $f$ и обозначают $\int f(t)\,dt$. Неопределенный интеграл определен с точностью до постоянного слагаемого. Он определяет класс эквивалентности функций $[\Phi]$: $\Phi_1\sim\Phi_2$ тогда и только тогда, когда $\Phi_1$ и $\Phi_2$ — неопределенные интегралы от $f$.

Предложение. *Если* $\Phi$ — *неопределенный интеграл от* $f$, *то*

$$\int_a^b f(x)\,dx = \Phi(b) - \Phi(a).$$

Доказательство.

$$\Phi(b) - \Phi(a) = (F(b) + \lambda) - (F(a) + \lambda) = F(b) - F(a) =$$
$$= \int_a^b f(x)\,dx - \int_a^a f(x)\,dx = \int_a^b f(x)\,dx. \;//$$

Как и при исследовании дифференцирования, эти результаты могут быть использованы для вычисления интегралов, некоторые примеры которых даны ниже.

Пример 5.6.

1. Если $F(t) = \int_0^t x\,dx$, то $F'(t) = t$, и неопределенные интегралы от функции $f$: $x \mapsto x$ есть

$$\Phi(t) = t^2/2 + \lambda, \qquad \lambda \in \mathbf{R}.$$

Таким образом,

$$F(t) = \Phi(t) - \Phi(0) = t^2/2.$$

2. В более общем случае, если $f: x \mapsto x^n$ для $n \in \mathbf{Z}\backslash\{-1\}$ и $F(t) = \int_a^t x^n\,dx$, то $F'(t) = t^n$, и неопределенный интеграл есть

$$\Phi(t) = \frac{t^{n+1}}{n+1} + \mu, \mu \in \mathbf{R}.$$

Тогда

$$F(t) = \Phi(t) - \Phi(a) = \frac{t^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}.$$

Очевидно, что это соотношение неверно при $n = -1$. Этот случай будет рассмотрен в п. 5.5.

**5.5. Некоторые специальные функции.** Мы предполагаем, что читатель знаком с геометрическими определениями функций $\sin x$ и $\cos x$, из которых следует, что

$$\sin: \mathbf{R} \to [-1, 1],$$
$$\cos: \mathbf{R} \to [-1, 1],$$

где

$$\mathscr{D}_{\sin} = \mathscr{D}_{\cos} = \mathbf{R}, \quad \mathscr{R}_{\sin} = \mathscr{R}_{\cos} = [-1, 1].$$

Мы также предполагаем знакомство с периодическими свойствами этих функций. Некоторые другие элементарные свойства приведены в следующем предложении.

Предложение. *Для всех* $x, y \in \mathbf{R}$ *имеем*:

а) $\sin(x+y) = \sin(x)\cos(y) + \cos(x)\sin(y)$;

б) $\sin(x-y) = \sin(x)\cos(y) - \cos(x)\sin(y)$;

в) $\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$;

г) $\cos(x-y) = \cos(x)\cos(y) + \sin(x)\sin(y)$;

д) $\sin^2 x + \cos^2 x = 1$;

е) $\frac{d}{dx}(\sin(x)) = \cos(x), \frac{d}{dx}(\cos(x)) = -\sin(x)$. //

Эти результаты непосредственно следуют из определений. Их доказательство оставляем читателю в качестве упражнения. Мы не будем касаться обоснования этих понятий (можно принять их в качестве допущений).

В п. 5.4 был определен $\int x^n\,dx$ при $n \neq -1$. В действительности интеграл $\int_1^t \frac{1}{x}\,dx$ существует для всех $t > 0$ и равен $\ln t$. Функция ln является отображением $]0, \infty[ \to \mathbf{R}$ и обладает свойством

$$\ln(xy) = \ln x + \ln y \text{ для всех } x, y \in ]0, \infty[,$$

так как

$$\frac{d}{dx}\ln(xy) = \frac{y}{xy} = \frac{1}{x} = \frac{d}{dx}\ln x.$$

Из результатов п. 5.4 имеем $\ln(xy) - \ln x = \lambda$, где $x \in ]0, \infty[$ и $\lambda \in \mathbf{R}$. В частности, при $x = 1$ имеем $\ln y - \ln 1 = \lambda$ и $\ln 1 = 0$; поэтому $\ln y = \lambda$. Следовательно,

$$\ln(xy) = \ln x + \ln y.$$

Можно показать, что ln биективна и, следовательно, существует функция

$$\exp\colon \mathbf{R} \to ]0, \infty[$$

такая, что $\ln(\exp p) = p$ для всех $p \in \mathbf{R}$ и $\exp(\ln q) = q$ для всех $q \in ]0, \infty[$. Из свойств функции ln следует, что

$$\exp(x+y) = \exp x \exp y \quad \text{для всех} \quad x, y \in \mathbf{R},$$

$$\exp 0 = 1,$$

$$\frac{d}{dx}\exp x = \exp x.$$

Удобно обозначить $\ln x$ через $\log_e x$, а $\exp x$ через $e^x$.

Функцию $\log_e x$ называют *натуральным логарифмом* числа $x$, а функцию $x \mapsto e^x$ — *экспоненциальной* функцией.

Если при $a > 0$ функция $f\colon ]-a, a[ \to \mathbf{R}$ принадлежит $C^\infty$ и $x \in ]-a, a[$, то ряд

$$\sum_{k=0}^{\infty} f^k(0)\frac{x^k}{k!}$$

называют *рядом Маклорена* для $f$ в точке $x$. Для некоторых функций можно показать, что ряд Маклорена сходится к значению функции $f$ в точке $x$. Другими словами, для $f$ имеем

$$f(x) = \lim_{N\to\infty} \sum_{k=0}^{N} f^{(k)}(0)\frac{x^k}{k!}.$$

В частности, это справедливо для функций $\sin x$, $\cos x$, $e^x$, для которых

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots,$$

$$\cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots;$$

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

для всех $x \in \mathbf{R}$.

Упражнение 3.5.

1. Показать, что последовательности, определенные ниже, сходятся, и найти их пределы:

а) $s_n = 1/n^2$;

б) $s_n = 3n/(n+3)$;

в) $s_n = 1 + 1/2^n$.

2. Пусть $(s_n)$ и $(t_n)$ — последовательности, $\lim_{n\to\infty} s_n = s$ и $n\,|t_n| < |s_n|$ для всех $n \in \mathbf{N}$. Показать, что $\lim_{n\to\infty} t_n = 0$.

3. Доказать, что если $(s_n)$ и $(t_n)$ имеют пределы $s$ и $t$ соответственно, то последовательность $(p_n)$, где $p_n = \lambda s_n$, имеет предел $\lambda s$. Если $t \neq 0$, то $\lim_{n\to\infty} (s_n/t_n) = s/t$.

4. Найти производные следующих функций (определить области, в которых существует производная):

а) $f\colon \mathbf{R} \to \mathbf{R}$, где $f(x) = x^{1/2}$;
б) $f\colon \mathbf{R}\backslash\{0\} \to \mathbf{R}$, где $f(x) = 1/x$;
в) $f\colon \mathbf{R} \to \mathbf{R}$, где $f(x) = |x|^2$.

5. Показать, что если $f$ и $g$ дифференцируемы в точке $x$, то:

а) $f + g$ дифференцируема в точке $x$ и

$$(f+g)'(x) = f'(x) + g'(x);$$

б) $fg$ дифференцируема в точке $x$ и

$$(fg)'(x) = (f'g)(x) + (fg')(x);$$

в) $f/g$ дифференцируема в точке $x$ при $g(x) \neq 0$ и

$$(f/g)'(x) = \frac{(f'g)(x) - (fg')(x)}{g^2(x)}.$$

6. Показать, что если $a_k \in \mathbf{R}$ при $0 \leqslant k \leqslant N$ и $p\colon \mathbf{R} \to \mathbf{R}$ определено соотношением

$$p(x) = \sum_{k=0}^{N} a_k x^k, \text{ то } p'(x) = \sum_{k=1}^{N} a_k k x^{k-1}.$$

7. Определить производные следующих функций:
а) $f\colon \mathbf{R} \to \mathbf{R}$, где $f(x) = x \sin x/(1 + \cos x)$,
б) $g\colon \mathbf{R} \to \mathbf{R}$, где $g(x) = \sin x^2 + x \cos^2 x$.

8. Вычислить интегралы:

а) $\int_1^2 x^3\,dx$; б) $\int_{-1}^{1} x^{1/3}\,dx$; в) $\int_0^{\pi/2} \cos x\,dx$.

9. Найти неопределенные интегралы:

а) $\int xe^{x^2}\,dx$; б) $\int \left(\frac{x^3 + 2xe^x + 1}{x}\right) dx$; в) $\int \sin x \cos x\,dx$.

## § 6. Операции

Часто некоторые функции (такие, например, как сложение целых чисел) используются при введении более простых обозначений. Это можно использовать для описания основных идей, изложенных в предыдущих параграфах, что, в свою очередь, позволит нам сделать доказательства более короткими и в то же время точно выделить свойства, на основе которых делаются выводы. Более детальное исследование будет проведено в гл. 5—8.

Определение. *Операцией над множеством* $S$ называется функция $f\colon S^n \to S$, $n \in \mathbf{N}$. В этом определении есть два важных момента, которые заслуживают особого упоминания. Во-первых, раз операция является функцией, то результат применения операции *однозначно определен*. Поэтому данный упорядоченный набор из $n$ элементов $S$ функция $f$ переводит только в один элемент $S$. Во-вторых, поскольку область значений операции лежит в $S$, на которое операция действует, будем говорить, что операция *замкнута* на $S$.

Говорят, что операция $S^n \to S$ *имеет порядок* $n$. Ограничимся рассмотрением ситуаций, когда порядок равен 1 или 2. В этом случае операции называют *монадическими* (или *унарными*) и *диадическими* (или *бинарными*) соответственно. Элементы набора из $n$ элементов в области определения называют операндами. Операции обычно обозначают символами, называемыми операторами. В случае унарных операций обычно символ оператора ставят перед операндом. //

Наиболее простым примером является операция изменения знака на $\mathbf{R}$. В предположении, что операция сложения уже определена, $-x$ определяет операцию $x \mapsto y$: $x + y = 0$ ($x$ отображается в $y$: $x + y = 0$).

Определение. Бинарные операции обозначают одним из трех способов. В первом случае оператор ставится между операндами (*infix*), во втором — перед операндами (*prefix*) и в третьем — после операндов (*postfix*). //

Пример 6.1.

$a + b$ *infix*,
$+ab$ *prefix*,
$ab+$ *postfix*. //

Переход от одной формы к другой нетруден и лучше всего описывается в терминах ориентированных графов, которые будут обсуждаться в § 6 гл. 7.

В соответствии с большинством математических текстов, исключая некоторые работы по алгебре и формальной логике, мы будем использовать обозначение *infix*. Другие обозначения имеют то преимущество, что не требуют скобок при определении порядка вычислений сложных выражений, и это делает их особенно удобными для автоматической обработки. Читатель может проверить соответствие между следующими парами выражений, записанными в формах *infix* и *postfix* соответственно:

а) $a+b*c+(d+e*(f+g))$,
$abc*+defg+*++$;
б) $(a+b)*c+d+e*f+g$,
$ab+c*d+ef*+g+$;
в) $a+(b*(c+d)+e)*f+g$,
$abcd+*e+f*+g+$.

Пример 6.2. Рассмотрим алгебраическое выражение

$$a+b*c+(d+e*(f+g))$$

и его представление на рис. 3.11, которое называют деревом. Из свойств арифметических операций мы знаем, что значение этого выражения можно вычислить многими способами. Однако если двигаться слева направо и снизу вверх, то получаем

$$\alpha \leftarrow b*c, \quad \beta \leftarrow a+\alpha, \quad \gamma \leftarrow f+g,$$
$$\delta \leftarrow e*\gamma, \quad \pi \leftarrow d+\delta, \quad \rho \leftarrow \beta+\pi.$$

Здесь греческими буквами обозначаются промежуточные результаты, за исключением $\rho$ — искомого результата.

Вычисление значения этого выражения с помощью дерева производится очень просто, однако если работать непосредственно с исходным выражением, то это можно

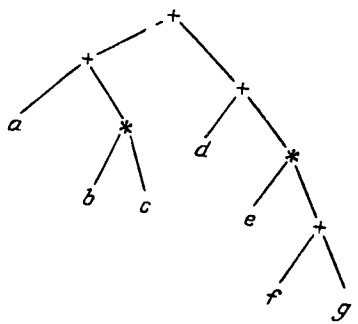
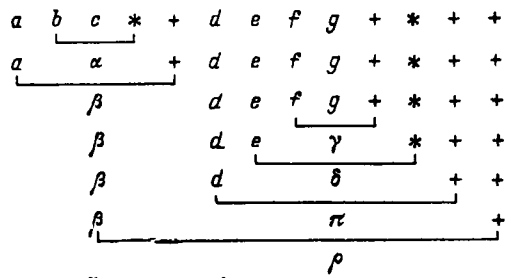


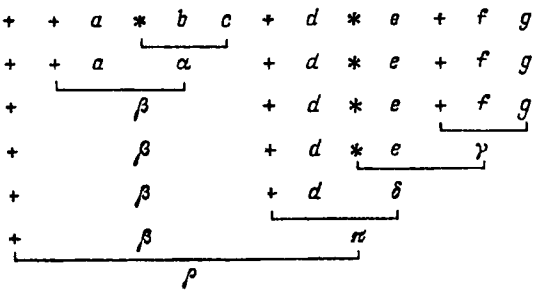

Рис. 3.11

сделать по-другому. Действительно, обычно (*infix*) выражение, как это показано в примере, нерегулярно потому, что некоторые подвыражения заключены в скобки, а некоторые нет. Особенно такая ситуация будет наблюдаться в том случае, если проинтегрировать информацию о различных символах на дереве (поскольку на самом деле его нет). Очевидно, что формы записи *prefix* и *postfix* этого выражения несут больше информации.

Вычисление значения выражения в форме *postfix* осуществляется следующим образом:



Аналогично в форме *prefix* вычисления осуществляются следующим образом:



«Переходы» по дереву показаны на рис. 3.12, *а* (форма *prefix*) на рис. 3.12, *b* (форма *postfix*) и на рис. 3.12, *с* (форма *infix*) со скобками:

$$((a+(b*c))+(d+(e*(f+g))))).$$

К этим вопросам мы вернемся позднее.

Конечно, мы уже знакомы со многими бинарными операциями, например с арифметическими операциями +, *, —, / и операциями над множествами — объединением (∪) и пересечением (∩).

Операции, определенные на конечных множествах, часто удобнее задавать при помощи таблиц.

Пример 6.3. Пусть операция ⊛ определена на множестве $\{a, b, c\}$ при помощи таблицы

| ⊛ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | $a$ | $a$ | $b$ |
| $b$ | $b$ | $a$ | $c$ |
| $c$ | $a$ | $b$ | $b$ |

Следовательно,

$$a \circledast b = a,$$
$$b \circledast b = a,$$
$$c \circledast b = b, \ldots \ //$$

Такие символы, как $\oplus$ и $\circledast$, будут использоваться для обозначения различных операций, которые будут вво-

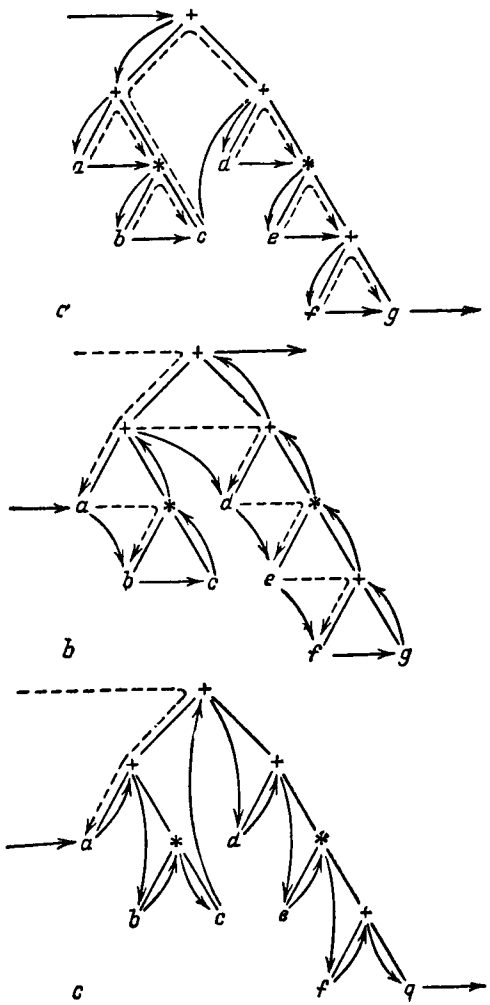



Рис. 3.12

диться в процессе изложения. Очевидно, что использование таблиц имеет важное значение, так как некоторые операции, с которыми приходится иметь дело в компьютерной математике, непригодны для словесного задания.

Обратим теперь внимание на свойства операций. Операции вместе со своими следствиями обеспечивают основу всех алгебраических вопросов математики, так как они определяют порядок работы с объектами.

Определение. Говорят, что бинарная операция $\circledast$ на множестве $A$ *коммутативна*, если

$$a \circledast b = b \circledast a \quad \text{для всех} \quad a, b \in A. \ //$$

Следовательно, обычная операция сложения на Z коммутативна, а вычитания — нет.

Определение. Говорят, что операция $\circledast$ на множестве $A$ *ассоциативна*, если

$$(a \circledast b) \circledast c = a \circledast (b \circledast c) \quad \text{для всех} \quad a, b, c \in A. \ //$$

Заметим, что в определении ассоциативности порядок операндов $a$, $b$ и $c$ сохранен (операция может быть некоммутативной!) и использованы круглые скобки, чтобы определить порядок вычислений.

Таким образом, выражение $(a \circledast b) \circledast c$ требует, чтобы сначала вычислялось $a \circledast b$ и результат этого (скажем, $x$) участвовал в операции с $c$, т. е. давал $x \circledast c$. Если операция ассоциативна, то порядок вычислений несуществен и, следовательно, скобки не требуются.

Пример 6.4. Над Z имеем

$$(1+2)+3 = 1+2+3 = 1+(2+3),$$

но

$$(1-2)-3 = -4 \quad \text{и} \quad 1-(2-3) = 2.$$

Таким образом, операция вычитания не ассоциативна. //

Коммутативность и ассоциативность являются двумя важными свойствами, которые могут быть определены для простых операций. Перед тем как описывать свойства, связывающие две операции, определим некоторые термины, относящиеся к специальным элементам множеств, к которым эти операции применяются.

Определение. Пусть $\circledast$ — бинарная операция на множестве $A$ и $l \in A$ такая, что

$$l \circledast a = a \quad \text{для всех} \quad a \in A.$$

Тогда $l$ называется *левой единицей* по отношению к $\circledast$

на $A$. Аналогично, если существует $r \in A$ такое, что

$$a \circledast r = a \quad \text{для всех} \quad a \in A,$$

то $r$ является *правой единицей* по отношению к $\circledast$. Далее, если существует элемент $e$, который является и левой, и правой единицей, т. е.

$$e \circledast a = a \circledast e = a \quad \text{для всех} \quad a \in A,$$

то $e$ называется (*двусторонней*) *единицей* по отношению к $\circledast$. //

Пример 6.5. Над R 0 является правой единицей по отношению к вычитанию и единицей по отношению к сложению, так как

$$a - 0 = a,$$

но

$$0 - a \neq a, \quad \text{если} \quad a \neq 0;$$
$$a + 0 = a \text{ и } 0 + a = a \text{ для всех } a. \; //$$

Определение. Пусть $\circledast$ — операция на $A$ с единицей $e$ и $x \circledast y = e$. Тогда говорят, что $x$ — *левый обратный* элемент к $y$, а $y$ — *правый обратный* элемент к $x$. Далее, если $x$ и $y$ такие, что

$$x \circledast y = e = y \circledast x,$$

то $y$ называется *обратным элементом* к $x$ по отношению к $\circledast$, и наоборот. //

Замечание. В некоторых работах левые (правые) обратные элементы относят к левой (правой) единице, однако, как мы скоро увидим, в большинстве случаев единицы являются двусторонними и, следовательно, не требуется делать никаких различий. Для решения уравнений необходимо существование и единственность единиц и обратных элементов. Менее общим свойством операций является идемпотентность, хотя оно используется в алгебре логики.

Определение. Пусть операция $\circledast$ на множестве $A$ и произвольный элемент $x \in A$ таковы, что $x \circledast x = x$. Тогда говорят, что $x$ *идемпотентен* по отношению к $\circledast$. //

Очевидно, что любое подмножество идемпотентно по отношению к операциям пересечения и объединения.

Определение. Пусть дано множество $A$, на котором определены две операции $\circledast$ и $\oplus$. Тогда, если

$$a \circledast (b \oplus c) = (a \circledast b) \oplus (a \circledast c) \quad \text{для всех} \quad a, b, c \in A,$$

то говорят, что $\circledast$ *дистрибутивна* по отношению к $\oplus$.

Если сказанное выше не совсем понятно, следует провести соответствие между этим тождеством и обычной арифметикой на R, например,

$$3*(1+2)=(3*1)+(3*2).$$

Может вызвать удивление, что в § 5 рассматривались только несколько специальных свойств, и можно прийти к выводу, что практически ничего нельзя вывести из того факта, что множество и связанные с ним операции обладают некоторыми из этих свойств. На самом деле (как будет видно из последующих глав) наиболее общеизвестная алгебра может быть построена из относительно небольшого набора основных правил. Сейчас мы продемонстрируем, как из элементарных предположений можно извлечь некоторые простые следствия; большинство примеров дано в виде упражнений.

Пример 6.6. Пусть $\circledast$ — операция на множестве $A$ и существует единица по отношению к $\circledast$. Тогда единичный элемент единствен.

Доказательство. Предположим, что $x$ и $y$ — единицы по отношению к $\circledast$, т. е.

$$x \circledast a = a \circledast x = a,$$

$$y \circledast a = a \circledast y = a \quad \text{для всех} \quad a \in A.$$

Тогда $x = x \circledast y$, так как $y$ — единица, и $x \circledast y = y$, поскольку $x$ — единица. Следовательно, $x = y$. //

Пример 6.7. Пусть $\circledast$ — ассоциативная операция на множестве $A$ и $e$ — единица по отношению к $\circledast$. Тогда если $x \in A$ и $x$ имеет обратный, то обратный элемент единствен по отношению к $\circledast$.

Доказательство. Допустим, что $x'$ и $x''$ — обратные элементы к $x$, так что

$$x \circledast x' = x' \circledast x = e \quad \text{и} \quad x \circledast x'' = x'' \circledast x = e.$$

Тогда

$$x' = x' \circledast e = x' \circledast (x \circledast x'') = (x' \circledast x) \circledast x'' =$$

$$= e \circledast x'' = x''. \ //$$

Упражнение 3.6.

1. Рассмотреть указанные ниже «определения» $\circledast$. Решить, правильно или нет каждое из них определяет бинарную операцию, и если так, то является ли операция коммутативной. Найти, если это возможно, единицу и обратный элемент к $x$. Предполагаются выполненными обычные свойства арифметики:

а) $x \circledast y = x - y$ на $\mathbf{N}$;

б) $x \circledast y = (x*y) - 1$ на $\mathbf{Z}$;

в) $x \circledast y = \max\{x, y\}$ на $\mathbf{N}$;

г) $x \circledast y = \sqrt{x^2 + y^2}$ на $\{x\colon 0 \leqslant x, x \in \mathbf{R}\}$;

д) $x \circledast y = x/y$ на $\{x\colon 0 < x,\ x \in \mathbf{R}\}$.

2. Определим операцию $\varphi$ на множестве $\{a, b, c\}$, как указано ниже. Проверить, что $\varphi$ ассоциативна и коммутативна и найти единичный элемент.

| $\varphi$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | $b$ | $c$ | $a$ |
| $c$ | $a$ | $b$ | $c$ |
| $b$ | $c$ | $a$ | $b$ |

3. Предполагая обычные свойства операций $+$, $-$, $*$ и $/$ на $\mathbf{R}$, доказать, что операция $\psi$, определенная на $[1, \infty[$ следующим образом:

$$a \psi b = \frac{(a*b) + 1}{a + b},$$

ассоциативна. Обосновать ответ.

Указание: не следует особо обращать внимание на область определения.

4. Пусть $\circledast$ — ассоциативная операция на множестве $A$ с единицей $e$ такая, что каждый элемент $a \in A$ обратим и обратный обозначается через $a'$. Показать, что

$$(a \circledast b)' = b' \circledast a'.$$

5. Показать, что если $\circledast$ — ассоциативная операция на множества $A$ с единицей $e$ такая, что $a \circledast a = e$ для каждого $a \in A$, то $\circledast$ коммутативна.

6. Пусть $\circledast$ — ассоциативная операция на множестве $A$ такая, что для любых $a$, $b \in A$, если $a \circledast b = b \circledast a$, то $a = b$. Показать, что каждый элемент $A$ идемпотентен по отношению к $\circledast$. Что можно сказать про $\circledast$, если операция имеет единицу?

ГЛАВА 4

# ОСНОВНЫЕ ПОНЯТИЯ АРИФМЕТИКИ

Итак, мы определили операции и описали некоторые их свойства. Теперь посмотрим, что можно сделать с совокупностью операций, заданных на множестве.

Множество с заданными на нем операциями называют алгебраической структурой. Некоторые из наиболее часто встречающихся алгебраических структур будут рассмотрены позднее. Прежде чем приступить к их рассмотрению, посмотрим на арифметику с неформальной точки зрения. В большинстве случаев мы будем опускать формальные определения, делая ударения на «следствия из правил», даже в тех случаях, когда это приводит к необычным способам использования известных символов, которые обычно используются для представления десятичных чисел.

## § 1. «Малая» конечная арифметика

Арифметику можно рассматривать как множество с двумя операциями, действующими подобно сложению и умножению. Ее можно изучать многими способами. Чтобы уяснить требования арифметической системы, примем конструктивное приближение и рассмотрим целые числа (0, 1, 2, ...) просто как символы. В дальнейшем будем рассматривать только конечную арифметику, в которой используется лишь конечное множество чисел; вначале это множество будет небольшим. Подразумевается, что если $A \sim \mathbf{N}_m$, то требуется $m$ различных символов, при этом никакие комбинации символов не разрешаются. Если используются только десятичные числа, то $m \leqslant 10$. Поскольку все множества данного размера биективны, то можно рассматривать только множества $\mathbf{N}_m$.

Для большей наглядности рассмотрим множество $\mathbf{N}_6$. Для этого необходимо построить таблицы умножения и сложения. Множество $\mathbf{N}_6$ достаточно велико для того, чтобы изучать свойства основной структуры. Можно по-

думать, что для этой цели более уместным является множество $N_2$, однако это не так. Начнем со сложения.

Операция сложения имеет единицу, которая обычно обозначается символом 0, однако $0 \notin N_6$. Поэтому будем использовать множество $Z_6 = \{0, 1, 2, 3, 4, 5\}$, которое более удобно. Очевидно, что $Z_6 \sim N_6$. Поэтому можно работать с $Z_6$, не теряя никаких свойств. Таким образом, к настоящему моменту мы имеем соответствующую табл. 4.1.

Таблица 4.1

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | | | | | |
| 2 | 2 | | | | | |
| 3 | 3 | | | | | |
| 4 | 4 | | | | | |
| 5 | 5 | | | | | |

Так как операция коммутативна, то таблица должна быть симметричной. Труднее обстоит дело с ассоциативностью. Если мы хотим, чтобы операция была ассоциативной, и требуем, как обычно, существования обратных элементов по сложению, то любой элемент должен входить ровно один раз в каждую строку и каждый столбец. Поясним это высказывание.

Если $a + b = a + c$, то

$$-a + (a + b) = -a + (a + c),$$
$$(-a + a) + b = (-a + a) + c,$$
$$0 + b = 0 + c,$$
$$b = c.$$

Рассмотрим теперь операцию, определенную в табл. 4.2. Из трех возможностей для операции сложения на $Z_6$ только $C$ удовлетворяет всем условиям, что выглядит несколько необычно. Операция $A$ не коммутативна, а в $B$ нарушен критерий «единственности результата». Как же построить соответствующую операцию, удовлетворяющую всем обсуждаемым выше свойствам? Из дальнейшего изложения будет видно, что наиболее трудно обеспечить выполнение свойства ассоциативности. В предложенной ниже процедуре мы используем ассо-

циативность как основной шаг построения, и, следовательно, это свойство будет выполняться автоматически.

*Шаг* 1. Число 0 является единицей для операции сложения. Поэтому получаем табл. 4.3.

Таблица 4.2

| $A$ | 0 | 1 | 2 | 3 | 4 | 5 | $B$ | 0 | 1 | 2 | 3 | 4 | 5 | $C$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 2 | 0 | 4 | 5 | 3 | 1 | 1 | 1 | 2 | 3 | 4 | 5 | 1 | 1 | 5 | 3 | 4 | 2 | 0 |
| 2 | 2 | 0 | 1 | 5 | 3 | 4 | 2 | 2 | 2 | 2 | 3 | 4 | 5 | 2 | 2 | 3 | 1 | 5 | 0 | 4 |
| 3 | 3 | 5 | 4 | 0 | 2 | 1 | 3 | 3 | 3 | 3 | 3 | 4 | 5 | 3 | 3 | 4 | 5 | 0 | 1 | 2 |
| 4 | 4 | 3 | 5 | 1 | 0 | 2 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 2 | 0 | 1 | 5 | 3 |
| 5 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 4 | 2 | 3 | 1 |

Таблица 4.3

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | | | | | |
| 2 | 2 | | | | | |
| 3 | 3 | | | | | |
| 4 | 4 | | | | | |
| 5 | 5 | | | | | |

*Шаг* 2. Определим следующую строку таблицы, удовлетворяющую условию «единственности результата». Чтобы подчеркнуть используемую технику, специально выберем результат, который отличается от привычного. Возьмем

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 0 | 5 | 2 | 4 |

Так как операция должна быть коммутативной, заполним соответствующий столбец табл. 4.4.

*Шаг* 3. Заполним другие клетки таблицы, используя ассоциативность. Проследим подробно за каждой деталью:

$$2+2=2+(1+4)=(2+1)+4=0+4=4,$$
$$2+3=2+(1+1)=(2+1)+1=0+1=1,$$
$$2+4=(2+1)+5=0+5=5,$$
$$2+5=(2+1)+3=0+3=3.$$

Здесь мы использовали соотношения $2+1=0$ и $0+x=x$. Далее

$$3+3=(1+1)+3=1+(1+3)=1+5=4 \text{ и т. д.}$$

Таким образом, на основе значений $1+x$ получаем таблицу для операции $+$ (табл. 4.5).

Таблица 4.4

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 3 | 0 | 5 | 2 | 4 |
| 2 | 2 | 0 | | | | |
| 3 | 3 | 5 | | | | |
| 4 | 4 | 2 | | | | |
| 5 | 5 | 4 | | | | |

Таблица 4.5

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 3 | 0 | 5 | 2 | 4 |
| 2 | 2 | 0 | 4 | 1 | 5 | 3 |
| 3 | 3 | 5 | 1 | 4 | 0 | 2 |
| 4 | 4 | 2 | 5 | 0 | 3 | 1 |
| 5 | 5 | 4 | 3 | 2 | 1 | 0 |

При выполнении процесса надо учитывать дополнительные ограничения на шаге 2. Значения в нулевой строке должны выбираться так, чтобы они «продолжали» все $Z_6$. Например, начиная с 1 (как мы делали), получаем

$$1+1=3, \quad 3+1=5, \quad 5+1=4, \quad 4+1=2,$$
$$2+1=0, \quad 0+1=1.$$

Следовательно, прибавляя только 1, можно получить все $Z_6$.

Перейдем теперь к умножению. Сначала заметим, что единица для операции умножения должна отличаться от нуля. В противном случае для любых $x$ и $y$ мы имели бы

$$x=0*x=x*0, \quad y=0+y=y+0,$$

поэтому

$$x*y=x*(0+y)=(x*0)+(x*y)=x+(x*y),$$

а значит, $x=0$. Поэтому 0 не является единицей для умножения.

На самом деле нам требуется число, которое будет порождать $Z_6$. Следовательно, мы могли бы определить аналогичным образом операцию умножения на основе частичной табл. 4.6. Однако в этом случае мы не должны настаивать на выполнении критерия «единственности результата». (В обычной арифметике не существует целого числа, которое при умножении на 2 давало бы 1! Поэтому в конечном множестве могут быть повторения.)

Вместо того чтобы повторять процедуру построения таблицы для умножения, вернемся к проблеме связи двух операций — дистрибутивности умножения относительно сложения. Эта проблема связана с ассоциативностью. Рассмотрим (уже построенную) операцию сложения.

Таблица 4.6

| * | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | | 0 | | | | |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | | 2 | | | | |
| 3 | | 3 | | | | |
| 4 | | 4 | | | | |
| 5 | | 5 | | | | |

Таблица 4.7

| * | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | 0 | 2 | 1 | 4 | 3 | 5 |
| 3 | 0 | 3 | 4 | 4 | 3 | 0 |
| 4 | 0 | 4 | 3 | 3 | 4 | 0 |
| 5 | 0 | 5 | 5 | 0 | 0 | 5 |

Заметим, что $1+1=3$. Поэтому из предположения дистрибутивности получаем, что

$$3*0=(1+1)*0=$$
$$=(1*0)+(1*0)=0+0=0,$$
$$3*2=(1*2)+(1*2)=2+2=4,$$
$$3*3=(1*3)+(1*3)=3+3=4,$$
$$3*4=(1*4)+(1*4)=4+4=3,$$
$$3*5=(1*5)+(1*5)=5+5=0.$$

Теперь $3+3=4$, $3+1=5$, $1+4=2$ и $1+2=0$. Действуя как и ранее, получаем следующую операцию (табл. 4.7).

Следовательно, начиная с почти произвольного выбора строки в таблице, не содержащей 1 по сложению, и накладывая ряд простых ограничений, мы приходим к приемлемой арифметической системе. Теперь достаточно установить, что полученная система не находится в противоречии с высказанными ранее соображениями, т. е. что $1+1$ действительно существует. Короче говоря, если в нормальной (бесконечной) арифметике $a+b=c$ и $c\in$ $\in \mathbf{Z}_6$, то хотелось бы, чтобы и в нашей арифметике ответ был $c$. Следовательно, мы пришли к выбору

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 5 | ? |

Недостающим элементом должен быть 0, поскольку $6\notin$ $\notin \mathbf{Z}_6$, и 0 является единственным элементом $\mathbf{Z}_6$, которо-

го нет в строке. В результате такого выбора получаем соответствующую табл. 4.8. Она определяет так называемую арифметику по модулю 6. (Эта арифметика работает точно так же, как и обычная целочисленная арифметика, за исключением того, что все целые числа заменяются на остатки от деления их на 6.)

Таблица 4.8

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 2 | 3 | 4 | 5 | 0 |
| 2 | 2 | 3 | 4 | 5 | 0 | 1 |
| 3 | 3 | 4 | 5 | 0 | 1 | 2 |
| 4 | 4 | 5 | 0 | 1 | 2 | 3 |
| 5 | 5 | 0 | 1 | 2 | 3 | 4 |

| * | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | 0 | 2 | 4 | 0 | 2 | 4 |
| 3 | 0 | 3 | 0 | 3 | 0 | 3 |
| 4 | 0 | 4 | 2 | 0 | 4 | 2 |
| 5 | 0 | 5 | 4 | 3 | 2 | 1 |

Упражнение 4.1.

1. По аналогии с «естественной» арифметикой, полученной для $Z_6$, построить аналогичную арифметику для $Z_{16}$, используя символы $\{0, 1, \ldots, 9, A, B, \ldots, F\}$.

2. Построить арифметику для $Z_6$, которая согласуется со строкой

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 2 | 2 | 3 | 4 | 0 | 5 | 1 |

3. Рассматривая $1 + 3$, показать, что следующая таблица приводит к противоречию:

| + | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 0 | 3 | 4 | 5 | 2 |

## § 2. «Большая» конечная арифметика

Мы уже построили арифметику для $Z_6$. Возникает вопрос: как можно расширить эту систему, чтобы иметь возможность считать после 5? Для этого достаточно иметь множество $n$-значных чисел (наборов из $n$ элементов из $Z_6$) с арифметикой над $Z_6$. Чтобы проиллюстрировать это, рассмотрим упорядоченную тройку из $Z_6$, т. е. элементы множества $Z_6 \times Z_6 \times Z_6$. Если $Z_6$ упорядочено обычным способом, т. е. $0 < 1 < 2 < 3 < 4 < 5$, тогда определим порядок на $Z_6 \times Z_6 \times Z_6$ по правилу

$$(a, b, c) < (x, y, z),$$

если $a < x$ или $a = x$ и $b < y$ или $a = x$, $b = y$ и $c < z$. В этом случае элементы $Z_6^3 = Z_6 \times Z_6 \times Z_6$ будут упорядочены следующим образом:

$$
\begin{array}{l}
(0, 0, 0), (0, 0, 1), \ldots, (0, 0, 5), \\
(0, 1, 0), \ldots, \quad\quad (0, 1, 5), \\
\cdots\cdots\cdots\cdots\cdots \\
(0, 5, 0), \ldots, \quad\quad (0, 5, 5), \\
(1, 0, 0), \ldots, \quad\quad (1, 0, 5), \\
\cdots\cdots\cdots\cdots\cdots \\
(5, 5, 0), \ldots, \quad\quad (5, 5, 5).
\end{array}
$$

Таким образом, существует $6^3 = 216$ различных троек. Поэтому нужно уметь производить арифметические вычисления над $Z_6^3$ в пределах от 0 до 215. В приведенном выше упорядочивании элементов из $Z_6^3$

$$
\begin{array}{l}
(0, 0, 5) \text{ непосредственно предшествует } (0, 1, 0). \\
(0, 1, 5) \text{ непосредственно предшествует } (0, 2, 0) \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
(0, 5, 5) \text{ непосредственно предшествует } (1, 0, 0).
\end{array}
$$

Следовательно, хотелось бы, чтобы выполнялось соотношение

$$(0, 0, 5) + 1 = (0, 1, 0).$$

Однако до сих пор не было представления 1 в $Z_6^3$. И хотя это соотношение выглядит естественным, мы должны заботиться о том, чтобы не использовать ни одного не определенного понятия. Для облегчения описания представим $Z_6^3$ как $A_2 \times A_1 \times A_0$ и рассмотрим сумму $(a_2, a_1, a_0)$ и $(b_2, b_1, b_0)$. Покомпонентное сложение дает $(a_2 + b_2, a_1 + b_1, a_0 + b_0)$, где сложение осуществляется в $Z_6$, и пока, как кажется, этого достаточно.

Пример 2.1. Рассмотрим соотношение

$$(0, 1, 3) + (4, 2, 1) = (4, 3, 4),$$

которое будет более наглядным, если записать его в виде

$$
\begin{array}{r}
0, 1, 3 \\
+ \\
\underline{4, 2, 1} \\
= 4, 3, 4
\end{array}
$$

Однако

$$\begin{array}{r} 1,\ 2,\ 5 \\ +\ 2,\ 3,\ 1 \\ \hline =\ 3,\ 5,\ 0 \end{array} \quad \text{и} \quad \begin{array}{r} 0,\ 0,\ 5 \\ +\ 0,\ 0,\ 1 \\ \hline =\ 0,\ 0,\ 0 \end{array} \begin{array}{l} (=1?) \\ (=0?). \end{array} //$$

В результате операции сложения множество $A_0$ переходит в себя. Однако для того чтобы сумма достаточно больших чисел (таких, как $5+1$) могла бы выйти за пределы $A_0$, нам необходимо производить некоторые действия в $A_1$ и также, возможно, в $A_2$. (Это иллюстрируется табл. 4.9).

Таблица 4.9

| $+_s$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 2 | 3 | 4 | 5 | 0 |
| 2 | 2 | 3 | 4 | 5 | 0 | 1 |
| 3 | 3 | 4 | 5 | 0 | 1 | 2 |
| 4 | 4 | 5 | 0 | 1 | 2 | 3 |
| 5 | 5 | 0 | 1 | 2 | 3 | 4 |

| $+_c$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 0 | 1 | 1 | 1 | 1 |
| 5 | 0 | 1 | 1 | 1 | 1 | 1 |

Возьмем любые два числа $a$ и $b$ из $\mathbf{Z}_6$. Тогда их сумма (в $\mathbf{Z}$) составляет

$$6 * (a +_c b) + (a +_s b).$$

Пример 2.2. 4 плюс 4 дает $6*(1)+(2)=8$ в $\mathbf{Z}$.

Таблица $+_s$ дает «обычную» сумму двух элементов из $\mathbf{Z}_6$, в то время как таблица $+_c$ показывает, когда необходим «переход» в следующее множество $\mathbf{Z}_6$, и содержит только нули и единицы. Значения в $+_c$ ограничены, потому что если

$$0 \leqslant x < n \quad \text{и} \quad 0 \leqslant y < n,$$

то

$$0 \leqslant x \leqslant x + y < x + n < n + n = 2n \quad (\text{и } n = 6 \text{ в } \mathbf{Z}_6). //$$

В действительности можно получить лучшую оценку, поскольку

$$0 \leqslant x \leqslant n - 1 \quad \text{и} \quad 0 \leqslant y \leqslant n - 1,$$

и, следовательно,

$$0 \leqslant x + y \leqslant 2n - 2 < 2n - 1.$$

Возвращаясь к суммированию $(a_2, a_1, a_0)$ и $(b_2, b_1, b_0)$

и обозначая ответ через $(d_2, d_1, d_0)$, получим

$$d_0 = a_0 +_* b_0,$$
$$x_0 = a_0 +_c b_0,$$
$$d_1 = a_1 +_* b_1 +_* x_0,$$
$$x_1 = \text{если } a_1 +_c b_1 = 1 \text{ тогда } 1,$$
$$\text{иначе } (a_1 +_* b_1) +_c x_0,$$
$$d_2 = a_2 +_* b_2 +_* x_1,$$
$$x_2 = \text{если } a_2 +_c b_2 = 1 \text{ тогда } 1,$$
$$\text{иначе } (a_2 +_* b_2) +_c x_1.$$

Поскольку $0 \leqslant a_i + b_i < 2n - 1$ и $x_i = 0$ или $x_i = 1$, то переносимый результат из $a_i + b_i + x_{i-1}$ никогда не может быть больше 1.

Заметим, что в наших определениях числа (0, 0, 0) и (0, 0, 1) в $\mathbf{Z}_6^3$ действуют, как 0 и 1 в новой арифметике. Кроме этого, если $x_2 = 5$, то результат сложения может оказаться слишком большим для $\mathbf{Z}_6^3$. В этом случае говорят, что произошло «переполнение». Этот случай мы обсудим более подробно в § 3 гл. 4, а до конца параграфа возможность перевыполнения будем игнорировать.

Аналогично можно использовать операции $*_p$ и $*_c$ (таблицы произведения и переноса), заданные в табл. 4.10 для того, чтобы производить умножение над $\mathbf{Z}_6^3$, однако мы не будем этим заниматься.

Таблица 4.10

| $*_p$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | 0 | 2 | 4 | 0 | 2 | 4 |
| 3 | 0 | 3 | 0 | 3 | 0 | 3 |
| 4 | 0 | 4 | 2 | 0 | 4 | 2 |
| 5 | 0 | 5 | 4 | 3 | 2 | 1 |

| $*_c$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 | 2 | 2 |
| 4 | 0 | 0 | 1 | 2 | 2 | 3 |
| 5 | 0 | 0 | 1 | 2 | 3 | 4 |

До сих пор мы рассматривали только символы, имеющие вид положительных чисел. Конечно, с символами 0, 1, 2, ... можно обращаться «естественным» образом, и, следовательно, их можно интерпретировать как неотрицательные числа. Арифметика над $\mathbf{Z}_6^3$ оперирует с числами от $0 = (0, 0, 0)$ до $215 = (5, 5, 5)$, которые были получены из последовательности (от 0 до 5) в $\mathbf{Z}_6$. Если

взять множество $\{-3, -2, -1, 0, 1, 2\}$ вместо $Z_6$, то получим систему, которая содержит отрицательные числа, но ведет себя странным образом.

Если мы возьмем два множества $Z_6$ и множество $\{-3, -2, -1, 0, 1, 2\}$, которое назовем $Z_6^-$, и образуем множество $Z_6^- \times Z_6 \times Z_6$, то можно построить арифметику с числами от $-108$ до 107. На самом деле арифметика является той же самой, за исключением того, что значения 3, 4 и 5 в $A_2$ сейчас интерпретируются как $-3$, $-2$, $-1$ соответственно. Поэтому, например, $(-2, 4, 2)$ вычисляется в **Z** как

$$(-2 * 36) + (4 * 6) + 2 = -46.$$

Биекция между двумя системами, определенная следующим образом:

$$3 \mapsto -3,$$
$$4 \mapsto -2,$$
$$5 \mapsto -1,$$
$$x \mapsto x \text{ в остальных случаях,}$$

может быть применена в любой момент при условии, что результат вычислений не имеет цифр 3, 4 или 5 в $A_2$. Мы выбираем обозначения из соображения удобства, не вводя ограничений на случаи, когда можно применять биекцию или обратное отображение. На этом этапе мы настойчиво советуем игнорировать любые очевидные противоречия, относящиеся к переполнению $A_2$. Этот случай будет подробно рассмотрен в последующих параграфах на более простом примере. Отметим, что новая система «переходит» от положительных чисел к отрицательным. Например,

$$(2, 2, 5) + (1, 0, 1) = \begin{cases} (3, 0, 0) & \text{в старой системе,} \\ (-3, 0, 0) & \text{в новой системе.} \end{cases}$$

(В **Z** это дает $107 + 1 = -108$!)

Вычисления, включающие в себя сложение и вычитание, в новой арифметике довольно просты и зависят от двух тождеств. Первое имеет вид

$$(5, 5, 5) + (0, 0, 1) = (0, 0, 0)$$

($(5, 5, 5)$ эквивалентно $(-1, 5, 5)$), а второе — вид

$$(a_2, a_1, a_0) + (b_2, b_1, b_0) = (5, 5, 5).$$

Они имеют место тогда и только тогда, когда

$$a_2 + b_2 = 5, \quad a_1 + b_1 = 5, \quad a_0 + b_0 = 5.$$

Таким образом, чтобы вычислить обратное по сложению число к $(a_2, a_1, a_0)$, мы должны сначала найти число $(b_2, b_1, b_0)$, которое называют *дополнением по* 5, и затем прибавить $1 = (0, 0, 1)$. Это даст дополнение до 6. Проиллюстрируем этот процесс на следующем примере.

Пример 2.3. Найдем обратные элементы к $(-3, 4, 1)$ и $(3, 4, 1)$.

| | | |
|---|---|---|
| Из | 3, 4, 1 | |
| получаем | 2, 1, 4 | (дополнение до 5) |
| | $+$ 1 | |
| | 2, 1, 5 | (дополнение до 6) |
| Проверим результат | 3, 4, 1 | |
| | $+$ | |
| | 2, 1, 5 | |
| | $= 0, 0, 0$ | |
| Поэтому | $-(-3, 4, 1) = (2, 1, 5)$. // | |

Таким образом, вычитание сводится к сложению с соответствующим дополнением.

Пример 2.4. Вычислим $(1, 3, 4) - (2, 1, 5)$.

| | | |
|---|---|---|
| Берем | 2, 1, 5 | |
| получаем | 3, 4, 0 | (дополнение до 5) |
| результат | 3, 4, 1 | (дополнение до 6) |
| прибавим (1, 3, 4) | (1, 3, 4) | |
| | $5, 1, 5 = (-1, 1, 5)$. | |

Проверяя вычисления над $\mathbf{Z}$, получаем $58 - 83 = -25$. // Конечно, причиной образования так называемых дополнений до 5 и 6 является тот факт, что мы проводим вычисления над $\mathbf{N}_6$ (или $\mathbf{Z}_6$).

В общем случае, если вычисления производятся в $\mathbf{N}_m$, мы должны использовать дополнение до $m-1$ и $m$ соответственно.

Надо подчеркнуть, что в вычислениях на ЭВМ мы обычно имеем дело с $\mathbf{Z}_m^n$ для некоторых фиксированных $m$ и $n$ и очень редко — с множеством $\mathbf{Z}$. Таким образом, совокупность имеющихся в нашем распоряжении чисел всегда ограничена, и, хотя границы могут быть очень большими, мы не должны забывать о том, что они существуют. Риск тем более велик по той причине, что в записи обычно опускают запятые и все нули, стоящие слева от первой ненулевой цифры за исключением числа

(0, 0, 0). Следовательно, (1, 3, 4) запишется как 134, (0, 0, 6) как 6 и (0, 0, 0) как 0.

Упражнение 4.2. Обозначим $\mathbf{Z}_m^- \times \mathbf{Z}_m^{n-1}$ через $\mathbf{Z}_m^{n-}$, где

$$\mathbf{Z}_m^- = \left\{-\frac{m}{2}, \ldots, 0, \ldots, \frac{m}{2} - 1\right\}, \quad \text{если } m \text{ четное,}$$

и

$$\mathbf{Z}_m^- = \left\{-\frac{m-1}{2}, \ldots, 0, \ldots, \frac{m-1}{2}\right\}, \text{ если } m \text{ печетное.}$$

Вычислить: а) $10 - 7$; б) $17 - 23$; в) $(-8) + (-21)$ в каждой из «естественных» арифметик $\mathbf{Z}_7^{4-}$, $\mathbf{Z}_{10}^{3-}$, $\mathbf{Z}_5^{5-}$, $\mathbf{Z}_{12}^{2-}$. (Замечание: числа в примерах заданы над $\mathbf{Z}$; поэтому вначале требуется перевести их в соответствующую систему, а потом проводить вычисления.)

## § 3. Двоичная арифметика

Из уже построенных арифметик над $\mathbf{Z}_m^n$ и $\mathbf{Z}_m^{n-}$ легко выделить основы двоичной арифметики. Существуют две так называемые двоичные арифметики. Первая — это *знаковая и модульная форма*, определенная на $\{-, +\} \times \mathbf{Z}_2^n$, т. е. $\mathbf{Z}_2^n$ (определенное в предыдущем параграфе) с добавленным знаком, расширяющим элементы $\mathbf{Z}_2^n$. Знак обычно кодируют в бинарной форме: 0 для «+» и 1 для «—». Вторая арифметика (*двоичная арифметика дополнений*) — это $\mathbf{Z}_2^{n-}$ с элементами $\{0, 1\}$ во всех $n$ позициях. Этот вид двоичной арифметики используется в большинстве компьютеров. Поэтому ограничим наши рассмотрения $\mathbf{Z}_2^{n-}$. Чтобы сделать обсуждение более конкретным, рассмотрим $\mathbf{Z}_2^{5-}$, элементы которого лежат в пределах от 10 000 (= −16) до 01 111 (= +15) (т. е. содержит $32 = 2^5$ различных чисел). Число −1 представляется в $\mathbf{Z}_2^{5-}$ как 11 111. Поэтому легко найти двоичное дополнение. Для этого надо слегка изменить все двоичные цифры, называемые *битами*, чтобы получить дополнение до 1, а затем прибавить 1, чтобы получить дополнение до 2.

Пример 3.1.

$$\begin{array}{r} -(01\,011) \\ 10\,100 \\ + \qquad\quad 1 \\ \hline 10\,101 \end{array}$$

$$(= -2^4 + 2^2 + 2^0 = -11);$$

$$\begin{array}{r} -(10\,110) \\ +\ 01\,001 \\ \hline 01\,010 \end{array}$$

$$(= 2^3 + 2^1 = 10).\ //$$

Очевидно, что могут возникнуть проблемы, вызванные ограниченностью чисел. Мы не можем их избежать, однако следует знать, когда возможна «ошибка». Форма дополнения делает проверку условия переполнения относительно легкой, использующей только значения старшего значащего бита. (В $Z_2^{5-}$ это бит с номером $2^4$.) Этот бит обозначает знак представляемого числа и называется знаковым битом или знаковым разрядом. Перед тем как проверить, какое значение имеет знаковый бит, напомним, что прибавление 1 к максимальному положительному числу в $Z_m^{n-}$ дает максимальное отрицательное число (наибольшее отрицательное число — это отрицательное число, отстоящее дальше всего от нуля). Другими словами, числа повторяются циклическим образом. В $Z_2^{5-}$ мы имеем ситуацию, изображенную на рис. 4.1.

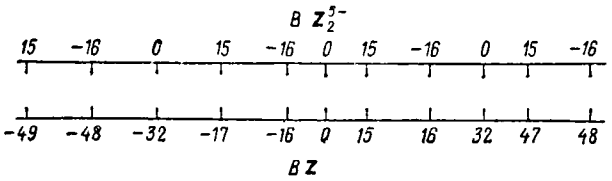



Рис. 4.1

Что же произойдет, если мы сложим два числа $x$ и $y$, где

$$-a \leqslant x < a \text{ и } -a \leqslant y < a \quad (\text{в } Z_2^{5-},\ a = 16)?$$

Сумма $x + y$ будет ограничена:

$$2a \leqslant x + y \leqslant 2a - 2 < 2a - 1.$$

Само по себе это неравенство ничего не дает. Поэтому мы рассмотрим три случая:

(I) если $-a \leqslant x < 0$ и $-a \leqslant y < 0$
тогда $-2a \leqslant x + y < 0$;

(II) если $0 \leqslant x < a$ и $0 \leqslant y < a$
тогда $0 \leqslant x + y \leqslant 2a - 2 < 2a - 1$;

(III) если $-a \leqslant x < 0$ и $0 \leqslant y < a$
тогда $-a \leqslant x + y < a$.

Вначале заметим, что результат в случае (III) находится в требуемых пределах и, следовательно, всегда правильный. Чтобы понять, как могут возникать ошибки в случаях (I) и (II), необходимо вспомнить, что если $z \in \in \mathbf{Z}$ и $-2a \leqslant z < -a$, то число $z$ представимо в конечной арифметике числом $z'$, где $z' = z + 2a$ и $0 \leqslant z' < a$. Аналогично, если $a \leqslant z < 2a - 1$, то $z$ представимо $z''$, где $z'' = z - 2a$ и $-a \leqslant z'' < 0$. Следовательно, ответ будет неправильный, если он в случае (I) положительный или в случае (II) отрицательный.

Чтобы объяснить эти заключения в терминах свойств знакового разряда, рассмотрим различные возможности сложения двух чисел:

а) оба числа отрицательны;

б) оба числа положительны;

в) числа имеют различные знаки.

Анализируя эти случаи, видим, что *переполнение* (ошибка переполнения) встречается тогда и только тогда, когда существует перенос в знаковый разряд или перенос из знакового разряда, *но не оба вместе*. Для иллюстрации этого рассмотрим несколько примеров в $\mathbf{Z}_2^{5-}$. Попытаемся сопоставить эти примеры со случаями (I)—(III) и а)—в), рассмотренными выше.

Пример 3.2. Напомним, что вычисления проводятся в $\mathbf{Z}_2^{5-}$:

$$\begin{array}{r} 1\,0\,1\,0\,1 \\ +\,1\,1\,0\,1\,0 \\ \hline 1\,0\,1\,1\,1\,1 \end{array} \qquad \begin{array}{r} 1\,1\,1\,0\,0 \\ +\,1\,0\,1\,1\,1 \\ \hline 1\,1\,0\,0\,1\,1 \end{array}$$

$$\begin{array}{r} 1\,1\,1\,0\,1 \\ +\,0\,0\,1\,1\,0 \\ \hline 1\,0\,0\,0\,1\,1 \end{array} \qquad \begin{array}{r} 0\,0\,1\,0\,1 \\ +\,0\,0\,1\,1\,1 \\ \hline 0\,1\,1\,0\,0 \end{array}$$

$$\begin{array}{r} 0\,1\,1\,0\,0 \\ +\,0\,1\,0\,1\,0 \\ \hline 1\,0\,1\,1\,0 \end{array}. \quad //$$

Вернемся теперь к умножению и делению. Сначала рассмотрим умножение. Напомним, что в $\mathbf{Z}$ (или, более точно, в $\mathbf{Z}_{10}^n$ для достаточно большого $n$) умножение на 10 можно получить «сдвигом» всех цифр на одну позицию влево и записью в 0-й позиции цифры 0. (В $\mathbf{Z}_m^n$ умножение на $m$ также всегда можно осуществить сдвигом влево.)

Следовательно, мы имеем простой способ умножения на неотрицательные степени числа 2 в $Z_2^n$ — сдвиг каждой цифры влево на соответствующее число позиций.

Пример 3.3. (Вычисления проводятся в $Z_2^{5-}$.)

```
0 0 0 1 1   (3)          1 1 1 1 0   (−2)
0 0 1 1 0   (*2)         1 1 1 0 0   (*2)
0 1 1 0 0   (*2 = 12)    1 1 0 0 0   (*2 = −8)

      0 0 1 0 1   (5)
      0 1 0 1 0   (*2)
      1 0 1 0 0   (*2)
      0 1 0 0 0   (*2 = 8). //
```

Из этих примеров видно, что метод также хорошо работает для отрицательных чисел, но результат будет с ошибкой (переполнения), если на каждом этапе менялся знак и если потом он опять изменился. Для умножения произвольного целого числа (элемента N) используем свойство дистрибутивности умножения по отношению к сложению и представим множитель как сумму степеней числа 2.

Пример 3.4. (Вычисления проводятся в $Z_2^{5-}$.)

$$3 * 5 = 3 * 2^2 + 3 * 2^0 (2^0 = 1)$$

$$(-5) * 3 = (-5) * 2^1 + (-5) * 2^0.$$

Поэтому

```
  0 0 0 1 1          1 1 0 1 1
+ 0 1 1 0 0        + 1 0 1 1 0
  0 1 1 1 1        1 1 0 0 0 1. //
```

Точно так же, как умножение производилось сдвигами влево, деление на положительные степени числа 2 осуществляется сдвигом вправо. (Деление на другие целые числа должно получаться путем сведения к вычитанию степеней числа 2. Этот процесс мы обсуждать не будем.) Однако специального рассмотрения требуют отрицательные числа. Отметим также, что в общем случае ожидаемый результат (т. е. арифметически ожидаемый результат в R) будет не целым, а дробным.

Пример 3.5. Попытаемся в $Z_2^{5-}$ вычислить 12/4, (−6)/2 и 7/4 сдвигом на 2, 1 и 2 позиции соответственно. Имеем

```
0 1 1 0 0          (12)          1 1 0 1 0      (−6)
0 0 0 1 1 | 0 0    (3 = 12/4)    0 1 1 0 1 | 0  (13 ≠ −6/2)

0 0 1 1 1 |        (7)
0 0 0 0 1 | 1 1    (1 ≈ 7/4). //
```

Сдвиг на одну позицию вправо автоматически сводит любое отрицательное число к положительному. В $Z_2^{5-}$ сдвиг переводит $-16$ к $+8$. Чтобы исправить это, следует отнять от результата число 16, что даст $-8$ (т. е. $-16/2$). То же самое можно получить, устанавливая знаковый разряд равным 1. Следовательно, правильный результат достигается использованием знакового разряда, значение которого равно 0 или 1 (в зависимости от знака числа), для того чтобы заполнить «пропуски», создаваемые в результате сдвига вправо.

Следовательно, $(-6)/2$ приводит к

$$11101 = -3.$$

Действие битов (со значением 1), «выпадающих» из числа в результате сдвига вправо, должно усекать результат. Поэтому 7/4 дает 1. Существует общепринятая практика округлять число (вверх независимо от знака) прибавлением к числу утерянного последнего бита. Это соответствует обычному арифметическому правилу округления, поскольку 1 в первом бите остатка представляет собой 0.5. Следовательно, 7/4 дает 2.

Упражнение 4.3.

1. Быстрый способ вычисления дополнения до 2 от данного битового элемента в $Z_2^{n-}$ заключается в следующем. Начиная с правого конца, копируем все идущие подряд нули и первую встретившуюся единицу. Затем все оставшиеся биты изменяем. Показать, что этот способ работает в большинстве случаев, и рассмотреть случай, когда он не работает.

2. Пусть в $Z_2^{5-}$ производятся следующие вычисления. Складывают два числа $x$ и $y$ (обозначим их сумму через $z$). Если от $z$ отнять $y$, то получим некоторый результат $c$, а если от $z$ отнять $c$, то получим некоторое число $d$. Что можно сказать о $c$ и $d$? Как отличаются результаты, если вычисления производятся в $Z_2^{n-}$?

## § 4. Логическая арифметика

Строго говоря, булева арифметика оперирует на множествах $Z_2$ и $Z_2^n$ и, следовательно, включает только числа 0 и 1. Для того чтобы подчеркнуть такую структуру, начнем с рассмотрения логической арифметики на «относительно большом» множестве $Z_5$. Она дает основу многозначной логики. Отсюда легко получить более простой

случай $Z_2$. Возьмем множество $Z_5 = \{0, 1, 2, 3, 4\}$ и операции $\vee$ и $\wedge$, определенные в табл. 4.11.

Таблица 4.11

| $\vee$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 |
| 1 | 1 | 1 | 2 | 3 | 4 |
| 2 | 2 | 2 | 2 | 3 | 4 |
| 3 | 3 | 3 | 3 | 3 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 |

| $\wedge$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 2 | 2 | 2 |
| 4 | 0 | 1 | 2 | 3 | 3 |
| 4 | 0 | 1 | 2 | 3 | 4 |

Упорядочивая $Z_5$ обычным образом (порядок индуцируется Z и R), видим, что

$$a \vee b = \max\{a, b\},$$

$$a \wedge b = \min\{a, b\}.$$

Обе операции коммутативны и ассоциативны, 0 является единицей для $\vee$, а 4 является единицей для $\wedge$; $\wedge$ дистрибутивна по отношению к $\vee$, но не наоборот.

Пример 4.1. Возьмем множество $Z_m$ с естественным порядком элементов. Введем операции $\wedge$ и $\vee$. Рассмотрим шесть возможных случаев упорядочивания трех произвольных элементов $a$, $b$, $c$ из $Z_m$:

(I) $a \leqslant b \leqslant c$;

(II) $a \leqslant c \leqslant b$;

(III) $b \leqslant a \leqslant c$;

(IV) $b \leqslant c \leqslant a$;

(V) $c \leqslant a \leqslant b$;

(VI) $c \leqslant b \leqslant a$.

Использование символа $\leqslant$ является интуитивным, однако может быть обосновано с помощью следующего определения:

$a \leqslant b$ тогда и только тогда, когда $a \vee b = b$.

Для проверки условия дистрибутивности нужно показать, что

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c).$$

Это можно сделать проверкой того, что обе части выражения совпадают для каждого из наборов $a$, $b$ и $c$. Бу-

дем одновременно вычислять и сопоставлять соответствующие выражения:

(I) $a \wedge (b \vee c) = a \wedge c = a,$

$(a \wedge b) \vee (a \wedge c) = a \vee a = a;$

(II) $a \wedge (b \vee c) = a \wedge b = a,$

$(a \wedge b) \vee (a \wedge c) = a \vee a = a;$

(III) $a \wedge (b \vee c) = a \wedge c = a,$

$(a \wedge b) \vee (a \wedge c) = b \vee a = a;$

(IV) $a \wedge (b \vee c) = a \wedge c = c,$

$(a \wedge b) \vee (a \wedge c) = b \vee c = c;$

(V) $a \wedge (b \vee c) = a \wedge b = a,$

$(a \wedge b) \vee (a \wedge c) = a \vee c = a;$

(VI) $a \wedge (b \vee c) = a \wedge b = b,$

$(a \wedge b) \vee (a \wedge c) = b \vee c = b.$

Следовательно, $\wedge$ дистрибутивна по отношению к $\vee$. //

Можно также показать (это как раз тот случай, когда мы не получаем ожидаемого результата), что $\vee$ дистрибутивна по отношению к $\wedge$, т. е. что

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

Проверку этого свойства оставляем в качестве упражнения.

Перед тем как закончить обсуждение общего случая, давайте вернемся к табл. 4.11, определяющим $\vee$ и $\wedge$. Элементы, имеющие одинаковые значения в таблицах, расположены относительно единичных элементов так, как показано на рис. 4.2. На самом деле каждая из этих

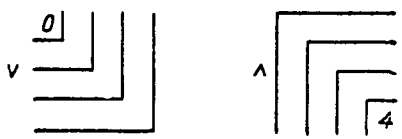



Рис. 4.2

операций является «отражением» другой и связь, которая позволяет одну операцию менять на другую, определяется (в $Z_5$) парами (0, 4), (1, 3), (2, 2), (3, 1), (4, 0). В сущности, это принцип двойственности, который

будет обсуждаться в гл. 5. Возвращаясь к $Z_2$, имеем

| $\vee$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

| $\wedge$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

В $Z_2$ операцию $\vee$ обычно интерпретируют как **или** (результат равен 1, если один из операндов равен 1, включая случай, когда они оба равны 1). Аналогично $\wedge$ читается как **и**. Число 0 является единичным элементом по отношению к **или**, число 1 является единичным элементом по отношению к **и**. Можно распространить эти результаты на более высокие размерности (переходя от $Z_2$ к $Z_2^n$), расширяя компоненты и учитывая, что не существует переноса из одной копии $Z_2$ к другой.

Пример 4.2.

$$\begin{array}{rcccccccccccc} & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ \wedge & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1. \end{array} \;//$$

Упражнение 4.4. Определяя операции $\wedge$ и $\vee$ как минимум и максимум, показать для произвольного $Z_n$, что

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

ГЛАВА 5

# АЛГЕБРАИЧЕСКИЕ СТРУКТУРЫ

В предыдущей главе мы уже познакомились с некоторыми способами определения операций над множествами и научились с ними работать с целью производить имеющие смысл вычисления. Конечно, существует много различных операций, которые могут быть определены на множестве, и, следовательно, в некотором смысле алгебраических структур больше, чем множеств. Однако получается так, что большинство полезных структур (под этим мы подразумеваем структуры, которые описывают естественно возникающие явления и пригодны для вычислений) может быть разбито на небольшое число типов. В этой главе мы вначале введем терминологию, которая имеет отношение ко всем алгебраическим структурам, а потом займемся некоторыми специальными структурами, которые, на наш взгляд, ближе всего относятся к вычислениям. Это нам позволит связать ранее «оборванные» нити рассуждений, с тем чтобы приступить к изучению структур, а также чтобы подвести серьезный математический фундамент под оставшуюся часть книги.

Центральное место в наших рассмотрениях занимают поля, линейная алгебра и булева алгебра. Поля формируют основу простой арифметики, линейная алгебра обеспечивает основу для геометрии и операций с числами, а булева алгебра содержит в себе основные положения элементарной логики. Разумнее начать изучение с таких структур, которые могут рассматриваться как части поля. Затем поля будут расширены до векторных пространств.

Аналогично мы расширим изучение булевой алгебры, включив решетки и свободные полукольца. Некоторые другие структуры будут кратко упомянуты в упражнениях к этой главе.

## § 1. Алгебраические структуры и подструктуры

Определение. *Алгебраической структурой* называется множество вместе с операциями (замкнутыми), определенными на этом множестве. //

Обычно операции имеют некоторые характерные свойства, которые могут быть обоснованы в виде теорем и которые используются в вычислениях. (Структуру вместе со всеми теоремами, правилами вычислений и вывода иногда называют *алгебраической системой.*)

К каждой структуре применимо понятие подструктуры. Чтобы это продемонстрировать, рассмотрим *гипотетическую* структуру, называемую указателем. Пусть $A$ — указатель. Предположим, что имеется только одна операция $\circledast$, определенная на $A$. Следовательно, более точно это может быть записано как $(A, \circledast)$, т. е. указатель состоит из множества $A$ с операцией $\circledast$. Теперь, если $B \subseteq A$ и $(B, \circledast)$ также является указателем, в частности $\circledast$ может быть замкнута на $B$, то $(B, \circledast)$ называется *подуказателем.*

Возьмем другую структуру, состоящую из множества $C$ и операции $\oplus$. ($\oplus$ и $\circledast$ должны иметь один и тот же порядок. Например, если одна из них является бинарной, то и другая должна быть такой же. Можно ввести и другие операции на $C$, однако в настоящее время мы их не рассматриваем). Если существует отображение $\varphi$: $A \to C$ такое, что

$$\varphi(x \circledast y) = \varphi(x) \oplus \varphi(y)$$

для любых $x$ и $y$ из $A$, то $\varphi$ называют *гомоморфизмом.*

Если существует гомоморфизм между $A$ и $C$, то в некотором смысле *образ* $(\varphi(A), \oplus)$ гомоморфизма из $(A, \circledast)$ ведет себя подобно прообразу, так как мы можем выполнить операцию $\circledast$ на $A$, а затем отобразить в $C$ (посредством $\varphi$) или сначала отобразить в $C$, а затем выполнить операцию $\oplus$. В обоих случаях результат будет один и тот же. Поэтому мы можем делать так, как нам удобнее. Эту ситуацию можно пояснить с помощью коммутативных диаграмм, изображенных на рис. 5.1. Диаграмма на рис. 5.1, *а* указывает включаемые множества или структуры, а диаграмма на рис. 5.1, *b* связывает отдельные элементы. На рис. 5.1, *b* справа изображены две различные формы одного и того же результата. Коммутативность диаграммы вытекает из определения операций.

На самом деле мы получаем $\varphi \circ \circledast = \oplus \circ \varphi$, что не является в строгом смысле коммутативностью, так как $\circledast$ и $\oplus$ существенно различны. Однако обе части равенства

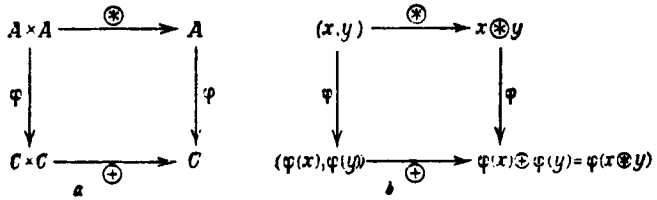



Рис. 5.1

означают комбинации операций одного и того же порядка и, следовательно, подходят под общее определение

*отображение* ∘ *операция* = *операция* ∘ *отображение*.

Рассмотрим пример.

Пример 1.1. Пусть отображение $\theta$, $\theta$: $Z \to Z_{10}$ — остаток от деления на 10. Тогда

$$\theta(20) = 0,$$
$$\theta(17) = 7, \ldots$$

Если мы рассмотрим простейшие системы $(Z, +)$ и $(Z_{10}, +)$ с операцией $+$, определенной естественным образом на $Z$ и на «единичном столбце» для $Z_{10}$, то легко видеть, что $\theta$ является гомоморфизмом. Например,

$$\theta(24+38) = \theta(62) = 2,$$
$$\theta(24)+\theta(38) = 4+8 = 2 \quad (\text{в } Z_{10}).$$

В этом случае диаграмма будет выглядеть так, как это изображено на рис. 5.2. //

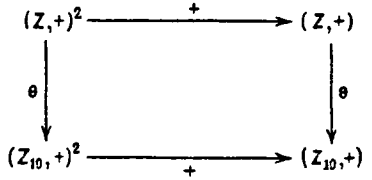



Рис. 5.2

Таким образом, гомоморфизм одной структуры в другую является отображением, которое сохраняет структуру.

Можно вводить ограничения на ранг отображения, чтобы получить, например, сюръективность или инъективность. Поэтому, если отображение является гомомор-

физмом, можно надеяться, что это обеспечит механизм перехода от структуры к структуре (и обратно!) без какой-либо потери информации.

Определение. Гомоморфизм, который является инъекцией, называют *мономорфизмом*, гомоморфизм, который является сюръекцией, называют *эпиморфизмом*, а гомоморфизм, который является биекцией, называют *изоморфизмом*. Если существует изоморфизм между двумя структурами, то говорят, что они *изоморфны*. //

Слово «изоморфно» означает «той же самой формы», и поэтому, кажется, разумно ожидать, что изоморфизм должен быть в состоянии разделить множество всех алгебраических структур на классы эквивалентности (см. упражнение 5.1, 2).

Пример 1.2. Структуры $(\{\varnothing, \mathscr{E}\}, \cap, \cup)$ и $(\{0, 1\}, \wedge, \vee)$ (см. определение в § 4 гл. 4) изоморфны.

Доказательство. Пусть $\varphi(\varnothing) = 0$ и $\varphi(\mathscr{E}) = 1$. Ясно, что $\varphi$ — биекция. Тогда

$$\varphi(\varnothing \cap \varnothing) = \varphi(\varnothing) = 0 = 0 \wedge 0 = \varphi(\varnothing) \wedge \varphi(\varnothing),$$
$$\varphi(\varnothing \cap \mathscr{E}) = \varphi(\varnothing) = 0 = 0 \wedge 1 = \varphi(\varnothing) \wedge \varphi(\mathscr{E}),$$
$$\varphi(\mathscr{E} \cap \varnothing) = \varphi(\varnothing) = 0 = 1 \wedge 0 = \varphi(\mathscr{E}) \wedge \varphi(\varnothing),$$
$$\varphi(\mathscr{E} \cap \mathscr{E}) = \varphi(\mathscr{E}) = 1 = 1 \wedge 1 = \varphi(\mathscr{E}) \wedge \varphi(\mathscr{E}),$$
$$\varphi(\varnothing \cup \varnothing) = \varphi(\varnothing) = 0 = 0 \vee 0 = \varphi(\varnothing) \vee \varphi(\varnothing),$$
$$\varphi(\varnothing \cup \mathscr{E}) = \varphi(\mathscr{E}) = 1 = 0 \vee 1 = \varphi(\varnothing) \vee \varphi(\mathscr{E}),$$
$$\varphi(\mathscr{E} \cup \varnothing) = \varphi(\mathscr{E}) = 1 = 1 \vee 0 = \varphi(\mathscr{E}) \vee \varphi(\varnothing),$$
$$\varphi(\mathscr{E} \cup \mathscr{E}) = \varphi(\mathscr{E}) = 1 = 1 \vee 1 = \varphi(\mathscr{E}) \vee \varphi(\mathscr{E}).$$

Таким образом, $\varphi$ является гомоморфизмом и, следовательно, изоморфизмом. //

В заключение отметим, что структура может быть изоморфна самой себе (имеется в виду изоморфизм, отличный от тривиального) и может также быть изоморфна одной из своих подструктур (это возможно лишь для бесконечных множеств).

Определение. Если область определения и область значений отображения совпадают, гомоморфизм называют *эндоморфизмом*, а изоморфизм называют *автоморфизмом*. //

Пример 1.3. Для заданного множества $A$ структура $(\mathscr{P}(A), \cap, \cup)$ изоморфна $(\mathscr{P}(A), \cup, \cap)$ с отображением $\varphi\colon X \to X'$.

Доказательство. Очевидно, что $\varphi$ инъективно и сюръективно. Если $B, C \in \mathscr{P}(A)$, то

$$\varphi(B \cap C) = (B \cap C)' = B' \cup C' = \varphi(B) \cup \varphi(C),$$
$$\varphi(B \cup C) = (B \cup C)' = B' \cap C' = \varphi(B) \cap \varphi(C).$$

Позднее мы увидим, что эти соотношения явно показывают самодвойственность булевой алгебры множеств и $\varphi$ является автоморфизмом. //

Упражнение 5.1.

1. Показать, что две структуры $(\mathbf{Z}_6, *)$, полученные при решении задачи из упражнения 4.1, 2 изоморфны.

2. Пусть $(A, \circledast)$, $(B, \oplus)$ и $(C, \odot)$ — указатели, а $\varphi\colon A \to B$ и $\theta\colon B \to C$ — изоморфизмы. Показать, что

$$\theta \circ \varphi\colon A \to C, \quad \varphi^{-1}\colon B \to A$$

также являются изоморфизмами.

## § 2. Простейшие операционные структуры

Начнем детальное изучение алгебраических структур с рассмотрения тех из них, которые имеют только одну бинарную операцию. Где это возможно, будем в этом и последующих параграфах представлять структуры в (приблизительно) возрастающем порядке «силы». (Говорят, что структура $A$ слабее структуры $B$, если $A$ можно рассматривать как $B$ «с отброшенной структурой». Как мы увидим позже, некоторые структуры получают путем «слияния» двух более слабых структур, и, следовательно, строгий порядок здесь невозможен.)

Обычно каждую структуру можно определить в терминах основных свойств, а не только в терминах простейших структур.

Определение. *Полугруппой* называется множество $S$ с бинарной операцией $\circledast$, которая удовлетворяет только требованию ассоциативности

$$x \circledast (y \circledast z) = (x \circledast y) \circledast z, \qquad x, y, z \in S. \ //$$

Определение. *Моноидом* называют множество $M$ вместе с бинарной операцией $\circledast$ такой, что

(I) $\circledast$ ассоциативна;

(II) существует $u \in M$ такое, что

$$u \circledast x = x = x \circledast u \quad \text{для всех} \quad x \in M;$$

$u$ называют *единицей по отношению к* $\circledast$. //

Полугруппы и моноиды имеют особое значение при обработке строк символов и теории языков.

Пример 2.1. Пусть $A = \{x, y, z\}$. Рассматривая $x$, $y$ и $z$ просто как символы, а не как имена объектов или «переменных», получаем, что $A$ является *алфавитом*. Определим $A^*$ как множество всех строк символов, принадлежащих $A$. Тогда $A^*$ включает $x$, $y$, $z$, $xx$, $xy$, $yx$, $xxyz$, $zyx$ и т. д.; $A^*$ бесконечно.

На $A^*$ можно определить операцию *конкатенации* $\odot$ следующим образом: если $\alpha, \beta \in A^*$, то $\alpha \odot \beta = \alpha\beta$, т. е. результатом является строка $\alpha$ и сразу же за ней записанная строка $\beta$. Таким образом, имеем

$$xyz \odot z = xyzz, \quad xz \odot yx = xzyx \quad \text{и т. д.}$$

Каждая строка $\alpha$ имеет конечную длину, которая обозначается через $|\alpha|$ и равна числу символов в $\alpha$ (при этом разрешаются повторения). Таким образом,

$$|x| = 1, \ |xy| = 2, \ |xxxzy| = 5.$$

Это несколько похоже на обозначение мощности множества. Заметим, в частности, что

$$\begin{aligned} |x| &= 1, & |\{x\}| &= 1, \\ |xy| &= 2, & |\{x, y\}| &= 2, \\ |yx| &= 2, & |\{y, x\}| &= 2, \end{aligned}$$

но

$$|xyx| = 3, \ |\{x, y, x\}| = |\{x, y\}| = 2.$$

Поэтому нет полной аналогии, хотя существует аналогия с пустым множеством. Для обозначения строки, аналогичной пустому множеству, будем использовать специальный символ $\Lambda$, $\Lambda \in A^*$. Таким образом, $|\Lambda| = 0$ и $\Lambda \odot \alpha = \alpha \odot \Lambda = \alpha$ для всех строк $\alpha$.

Следовательно, для любого алфавита $A$ структура $(A^*, \odot)$ является моноидом, а $\Lambda$ — единица по отношению к $\odot$. //

(Может показаться, что символ * неверно употребляется в обозначении $A^*$, однако это не так: $A^* = R^*(\Lambda)$, где

$$R = \{(\alpha, \beta): \beta = \alpha \odot a, \ a \in A\}.)$$

Приведенный выше пример чрезвычайно важен. Для заданного, достаточно большого алфавита $A$, содержащего, например, все символы, доступные периферийным уст-

ройствам какого-либо компьютера, все языки, используемые в компьютерных системах, являются подмножеством $A^*$. Это понятие является основным при формальном изучении языков (см. гл. 8).

Третьей (и последней) операционной структурой является непосредственное и естественное расширение моноида.

Определение. *Группой* $G$ называют множество с бинарной операцией $\circledast$ такой, что

а) $\circledast$ ассоциативна;

б) существует элемент $u \in G$ (единица по отношению к $\circledast$) такой, что

$$u \circledast x = x = x \circledast u \quad \text{для всех} \quad x \in G;$$

в) каждому элементу $x \in G$ соответствует элемент $y \in G$ такой, что

$$x \circledast y = u = y \circledast x;$$

$y$ называется *обратным* элементом к $x$ по отношению к $\circledast$. //

В случаях, когда групповая операция обозначается символом $\circledast$, единичный элемент обозначается 1, а обратный к элементу $x$ элемент записывается в виде $x^{-1}$. Когда групповая операция обозначается символом $\oplus$, единичный элемент обозначается 0, а обратный к элементу $x$ элемент записывается в виде $-x$.

По сравнению с первыми двумя структурами группы обладают следующими важными свойствами. Внутри группы $(G, \circledast)$ можно решить уравнение

$$a \circledast x = b.$$

Более того, решение легко найти (однако заметим, что для этого требуются все аксиомы группы). Если

$$a \circledast x = b$$

то

$$a^{-1} \circledast (a \circledast a) = a^{-1} \circledast b \quad (a \in G \Rightarrow a^{-1} \in G),$$

$$(a^{-1} \circledast a) \circledast x = a^{-1} \circledast b \quad (\circledast \text{ ассоциативна}),$$

$$u \circledast x = a^{-1} \circledast b \quad \text{(свойство обратных элементов)},$$

поэтому

$$x = a^{-1} \circledast b \quad \text{(свойство единицы)}.$$

Часто к словам «группа» и «моноид» приписывают термин «коммутативный». Это просто означает, что опера-

ция в рассматриваемой структуре удовлетворяет свойству коммутативности, т. е.

$$y \circledast x = x \circledast y \quad \text{для всех} \quad x, y \in M \quad \text{или} \quad G.$$

Можно сделать много полезных выводов из аксиом группы. Рассмотрим простой пример.

Пример 2.2. В группе $(G, *)$

$$(a * b)^{-1} = b^{-1} \circledast a^{-1}.$$

Доказательство.

$$(a * b) * (b^{-1} * a^{-1}) = a * (b * b^{-1}) * a^{-1} =$$
$$= a * 1 * a^{-1} = a * a^{-1} = 1.$$

Следовательно, $b^{-1} * a^{-1}$ является правым обратным элементом к $a * b$. Аналогично можно показать, что он является левым обратным элементом, откуда и следует требуемый результат. //

Группы дают нам первый пример широко используемых изоморфизмов. Эти изоморфизмы (между группами $(\mathbf{R}, +)$ и $(]0, \infty[, *)$) называют логарифмами. Они позволяют выполнять умножение при помощи сложения на основе следующего тождества:

$$a * b = \varphi^{-1}(\varphi(a) + \varphi(b)),$$

где

$$\varphi\colon x \mapsto \log_p(x) \quad \text{для некоторого} \quad p \in ]-1, \infty[.$$

Упражнение 5.2.

1. Доказать единственность единичного элемента и обратных элементов в группе $(G, \circledast)$.

2. В группе $(G, \circledast)$ показать, что если $a \circledast b = a \circledast c$, то $b = c$, а если $x \circledast a = y \circledast a$, то $x = y$.

3. Проверить, что множество перестановок конечного множества образует группу по отношению к операции умножения перестановок.

## § 3. Кольца и поля

По-видимому, простые операционные структуры § 2 не были знакомы читателю. Сейчас мы готовы использовать свойства групп для описания арифметических структур, обсуждавшихся в гл. 4. Наибольший интерес для нас представляют поля (на текущий момент это наиболее идеальные арифметические структуры) и их классификация в терминах размерности. Однако сначала мы

кратко рассмотрим структуры, которые несколько отличаются от полей и называются кольцами.

**3.1. Кольца.** Многие математические конструкции, которые естественно возникают в линейной алгебре (особенно в теории матриц), являются кольцами или включают кольца как подструктуры. Следовательно, примеры колец часто будут появляться в этой главе и в гл. 6. Мы уже изучили одну совокупность колец. Вернемся к ней после введения аксиоматических понятий.

Определение. *Кольцом* называется множество $R$ с двумя определенными на нем бинарными операциями $\circledast$ и $\oplus$ такими, что:

а) $\circledast$ ассоциативна;

б) $\oplus$ ассоциативна;

в) $\oplus$ коммутативна;

г) $\oplus$ имеет единицу, которая называется *нулем* и обозначается 0;

д) существуют обратные элементы относительно $\oplus$;

е) $\circledast$ дистрибутивна по отношению к $\oplus$, т. е.

$$x \circledast (y \oplus z) = (x \circledast y) \oplus (x \circledast z),$$

$$(x \oplus y) \circledast z = (x \circledast z) \oplus (y \circledast z) \quad \text{для всех} \quad x, y, z \in R. \ //$$

Следовательно, система $(\mathbf{Z}_n, *, +)$ при любом $n \in \mathbf{N}$ является кольцом.

Будем говорить, что кольцо *коммутативно*, если умножение $\circledast$ коммутативно, и является *кольцом с единицей*, если существует единица относительно умножения. Как обычно, ее обозначают символом 1. Легко показать, что в кольце $(R, \circledast, \oplus)$ для любых $a, b \in R$ выполняются соотношения

$$0 \circledast a = a \circledast 0 = 0,$$

$$a \circledast (-b) = (-a) \circledast b = -(a \circledast b),$$

$$(-a) \circledast (-b) = a \circledast b.$$

Здесь $-a$ это элемент, обратный к $a$ относительно $\oplus$; $a \oplus (-b)$ записывают обычно как $a - b$, и если $1 \in R$, то 1 единственна.

Пример 3.1. $(\mathbf{Z}_n, *, +)$ является коммутативным кольцом с единицей при любом $n \in \mathbf{N}$. //

В системе $(\mathbf{Z}_n, *, +)$ не всегда возможно «деление». В этом состоит основное отличие между полем и коммутативным кольцом с единицей. Рассмотрим кольцо $(\mathbf{Z}_6, *, +)$. Покажем, что оно не является полем.

В $Z_6$ существует 15 случаев, когда произведение двух элементов может давать нуль, а именно:

(0, 0),

(0, 1), (0, 2), (0, 3), (0, 4), (0, 5),

(1, 0), (2, 0), (3, 0), (4, 0), (5, 0),

(2, 3), (3, 4),

(3, 2), (4, 3).

Очевидно, что существуют утверждения, которые справедливы не для всех арифметических вычислений. При умножении выражений мы явно используем тот факт, что $a * b = 0$ тогда и только тогда, когда $a = 0$ или $b = 0$.

В кольце $(R, \circledast, \oplus)$ нулевые элементы $x$ и $y$ называют *делителями нуля*, если их произведение равно нулю. В случае, когда $R$ не является коммутативным кольцом, $x$ называют *левым делителем нуля*, а $y$ — *правым делителем нуля*. Нетрудно показать (см. упражнение 5.3), что $Z_6$ имеет делители нуля (так как 6 — составное число) и что $Z_p$ не имеет делителей нуля тогда и только тогда, когда $p$ является простым числом.

Если в группе $(G, \circledast)$

$$a \circledast b = a \circledast c,$$

то $b = c$. Однако в случае произвольного кольца это неверно.

Теорема. *Приведенное выше условие имеет место в кольце $R$ тогда и только тогда, когда $R$ не содержит делителей нуля.*

Доказательство. Достаточность. Предположим, что $R$ не имеет делителей нуля. Тогда, если $x \circledast y = x \circledast z$ и $x \neq 0$, то

$$(x \circledast y) - (x \circledast z) = (x \circledast y) - (x \circledast y) = 0,$$

$$(x \circledast y) - (x \circledast z) = (x \circledast y) \oplus (x \circledast (-z)) =$$

$$= (x \circledast (y \oplus (-z)) = x \circledast (y - z).$$

Поэтому $x \circledast (y - z) = 0$; но так как не существует делителей нуля и $x \neq 0$, то отсюда следует, что $y - z = 0$ и, следовательно, $y = z$. Аналогично, если $y \circledast x = z \circledast x$, то $y = z$. Достаточность доказана.

Необходимость. Предположим, что из $a \circledast b = a \circledast c$ следует равенство $b = c$, и пусть $x \circledast y = 0$. Тогда

$$x \circledast y = x \circledast 0$$

(см. упражнение 5.3), и если $x \neq 0$, то $y = 0$. Аналогич-

но, если $y \neq 0$, то

$$x \circledast y = 0 = 0 \circledast y$$

и по предположению $x = 0$. Таким образом, из $x \circledast y = 0$ следует, что или $x = 0$, или $y = 0$. //

Рассмотрим теперь еще одну структуру, перед тем как перейти к изучению полей

Определение. *Областью целостности* называется коммутативное кольцо с единицей, не имеющее делителей нуля, т. е. множество $D$ с двумя бинарными операциями $\circledast$ и $\oplus$ такими, что:

а) сложение $\oplus$ ассоциативно;

б) сложение коммутативно;

в) существует единица по сложению, обозначаемая 0;

г) существуют обратные элементы по сложению (обозначаются $(-x)$);

д) умножение $\circledast$ ассоциативно;

е) умножение коммутативно;

ж) существует единица по умножению (обозначается 1);

з) умножение дистрибутивно по отношению к сложению:

$$(x \circledast (y \oplus z)) = (x \circledast y) \oplus (x \circledast z) \text{ для всех } x, y, z \in D;$$

и) если $x \neq 0$ и $x \circledast y = x \circledast z$, то $y = z$. //

Каждая конечная область целостности является полем, однако существуют примеры бесконечных областей целостности, не являющихся полями.

**3.2. Поля.** Уже работая с понятиями арифметики, мы сталкивались с аксиоматическим определением поля.

Определение. *Полем* называется множество $F$ с двумя определенными на нем бинарными операциями — сложением $\oplus$ и умножением $\circledast$ (обозначается $(F, \circledast, \oplus)$ или же просто $F$), которые удовлетворяют следующим девяти свойствам.

1. Сложение коммутативно:

$$x \oplus y = y \oplus x \text{ для всех } x, y \in F.$$

2. Сложение ассоциативно:

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z \text{ для всех } x, y, z \in F.$$

3. Существует элемент в $F$, который обычно обозначается символом 0, такой, что

$$x \oplus 0 = x \text{ для всех } x \in F;$$

0 называется аддитивной единицей или просто нулем.

4. Каждому элементу $x \in F$ соответствует элемент $y \in F$ такой, что

$$x \oplus y = 0;$$

$y$ называется аддитивным обратным элементом к $x$ и обозначается через $-x$.

5. Умножение коммутативно:

$$x \circledast y = y \circledast x \quad \text{для всех} \quad x, y \in F.$$

6. Умножение ассоциативно:

$$x \circledast (y \circledast z) = (x \circledast y) \circledast z \quad \text{для всех} \quad x, y, z \in F.$$

7. Существует элемент в $F$, который обычно обозначается символом 1, такой, что $1 \neq 0$ и

$$x \circledast 1 = x \quad \text{для всех} \quad x \in F;$$

1 называют мультипликативной единицей или просто единицей.

8. Каждому элементу $x \in F \backslash \{0\}$ соответствует элемент $y \in F$ такой, что

$$x \circledast y = 1;$$

$y$ называется мультипликативным обратным элементом к $x$ и обозначается через $x^{-1}$.

9. Умножение дистрибутивно относительно сложения:

$$x \circledast (y \oplus z) = (x \circledast y) \oplus (x \circledast z) \quad \text{для всех} \quad x, y, z \in F. \ //$$

Пример 3.2. $(\mathbf{R}, *, +)$ является полем, и, следовательно, $(\mathbf{R}, +)$ и $(\mathbf{R}\backslash\{0\}, *)$ — коммутативные группы. $(\mathbf{N}, *, +)$ не является полем, поскольку не существует ни аддитивной единицы, ни аддитивных обратных элементов. $(\mathscr{P}(A), \cap, \cup)$ для заданного множества $A$ не является полем, поскольку не существует обратных элементов. //

В предыдущем определении использовались символы $\circledast$ и $\oplus$ для того, чтобы подчеркнуть, что операции в поле могут отличаться от умножения и сложения. Однако в дальнейшем часто будут рассматриваться поля $(\mathbf{R}, *, +)$ и $(\mathbf{Q}, *, +)$. Поэтому мы будем использовать символы $*$ и $+$.

Перед тем как перейти к доказательству основных утверждений, напомним (вместе с доказательствами) некоторые важные следствия, которые непосредственно извлекаются из определения поля.

Предложение. *Единичный элемент в поле единствен.*

Доказательство. Предположим, что

$$x * e = x \text{ и } x * e' = x \text{ для всех } x \in F.$$

Тогда

$$e = e * e' = e' * e = e'$$

Поэтому $e = e' = 1$.

Для операции сложения доказательство аналогично. //

Предложение. *Обратные элементы в поле единственны.*

Доказательство. Опять рассмотрим случай операции умножения. Возьмем $x \in F \setminus \{0\}$ и допустим, что имеется два элемента $y$ и $z$ таких, что

$$x * y = 1, \quad x * z = 1.$$

Из коммутативности следует, что

$$y * x = 1 = z * x;$$

поэтому

$$y = y * 1 = y * (x * z) = (y * x) * z = 1 * z = 1.$$

Следовательно, $y = z = x^{-1}$.

Единственность обратных элементов по сложению доказывается аналогично. //

Перейдем к основным результатам.

Теорема. *В поле* $(F, *, +)$ *для любых* $a, b \in F$ *справедливы следующие утверждения*:

а) $a * 0 = 0$;

б) $(-a) = a * (-1)$, $-a(+b) = (-a) + (-b)$;

в) $-(-a) = a$, $(-1) * (-1) = 1$;

г) *если* $a \neq 0$, *то* $(a^{-1})^{-1} = a$;

д) $a * b = 0 \Rightarrow a = 0$ *или* $b = 0$;

е) $(-a) * (-b) = a * b$.

Доказательство.

а) $a * 1 = a$ и $a + 0 = a$; поэтому

$$a + (a * 0) = (a * 1) + (a * 0) = a * (1 + 0) = a * 1 = a.$$

Следовательно, $a * 0$ является единицей по сложению. Она единственна; поэтому $a * 0 = 0$.

б) Аналогично

$$a + (a * (-1)) = (a * 1) + (a * (-1)) =$$
$$= a * (1 + (-1)) = a * 0 = 0.$$

Таким образом, $-a = a * (-1)$. Используя это равенство, получаем

$$-(a+b) = (a+b) * (-1) = \\ = (a * (-1)) + (b * (-1)) = (-a) + (-b).$$

в) По определению

$$(-a) + a = 0 \quad \text{и} \quad (-a) + (-(-a)) = 0.$$

Но обратные элементы единственны; поэтому $a = -(-a)$. Таким образом, $1 = -(-1)$. Пусть $x$ равно $-1$. Тогда

$$1 = -(x) = x * (-1) = (-1) * (-1).$$

г) Заметим вначале, что $a^{-1} \neq 0$, так как в противном случае

$$1 = a * a^{-1} = a * 0 = 0,$$

что противоречит свойствам поля. Следовательно, $a^{-1} \neq 0$ и доказательство аналогично доказательству случая в).

д) Возьмем $a \neq 0$. Тогда $a^{-1}$ определено и

$$b = 1 * b = (a^{-1} * a) * b = a^{-1} * (a * b) = a^{-1} * 0 = 0.$$

е) Из случая б) следует

$$(-a) = a * (-1) \quad \text{и} \quad (-b) = b * (-1).$$

Поэтому

$$(-a) * (-b) = (a * (-1)) * (b * (-1)) = \\ = a * ((-1) * (-1)) * b = a * 1 * b = a * b. \ //$$

Для упрощения записи выражений в полях примем обычное соглашение о том, что если нет скобок, то умножение выполняется раньше сложения. Например, выражение $a + b * c$ означает $a + (b * c)$.

Из аксиом поля следует разрешимость линейных уравнений. Это — очень важное свойство, и можно привести аргументы в пользу того, что оно — основное свойство полей.

Линейным уравнением относительно $x$ над полем $F$ называется выражение вида $a * x + b = 0$, где $0$, $a$, $b \in F$.

Теорема. *Если* $a \neq 0$, *то линейное уравнение* $a * x + b = 0$ *имеет единственное решение в* $F$ (*т. е. существует только один элемент поля, при подстановке которого в уравнение получается верное равенство*).

Доказательство.

$a * x + b = 0,$

$a * x + b + (-b) = 0 + (-b),$

$a * x + (b + (-b)) = (-b)$ (ассоциативность и свойство 0),

$a * x + 0 = (-b)$ (по определению 0),

$a * x = (-b)$ (свойство 0),

$a^{-1} * (a * x) = a^{-1} * (-b)$ ($a \neq 0$),

$(a^{-1} * a) * x = (-b) * a^{-1}$ (ассоциативность и коммутативность),

$1 * x = (-b) * a^{-1}$ (по определению 1),

$x = (-b) * a^{-1}$ (свойство 1).

Поле $F$ замкнуто относительно заданных операций. Поэтому элемент $(-b) * a^{-1}$ содержится в $F$. Этот элемент и дает решение уравнения. Более того, так как обратные элементы в $F$ единственны, то $-b$ и $a^{-1}$ определяются из данного уравнения единственным образом, и, следовательно, решение единственно. //

Уравнения, которые получаются из полиномов более высоких степеней, например квадратные уравнения

$$a * x * x + b * x + c = 0$$

с коэффициентами $a$, $b$, $c$ из $\mathbf{R}$, в общем случае неразрешимы в $\mathbf{R}$. Для того чтобы эти уравнения были разрешимы, нужно перейти к расширению поля $\mathbf{R}$ — полю комплексных чисел. Однако полиномиальные уравнения с комплексными коэффициентами всегда разрешимы в поле комплексных чисел: никакого более широкого поля не требуется. Исследование этого интересного факта могло бы увести нас в сторону от более уместных тем.

**3.3. Конечные поля.** До сих пор все упоминаемые поля были бесконечны (содержали множества, имеющие мощность $\aleph_0$ или $\aleph_1$). Обсудим теперь возможность существования конечных полей — полей, содержащих конечное число элементов. Сформулируем основные свойства конечных полей и докажем некоторые из них (доказательства других лежат за пределами этой книги). Вначале приведем некоторую дополнительную информацию.

Пусть $a \in F$. Тогда элементы $a$, $a + a$, $a + a + a$, ... являются элементами поля. Обозначим их через $a$, $2a$, $3a$, ..., $na$, ... (не требуется, чтобы $n \in F$) соответствен-

но. Аналогично $a$, $a * a$, $a * a * a$, ... также являются элементами поля. Обозначим их через $a$, $a^2$, $a^3$, ..., $a^n$, ... соответственно. Предположим, что $a \neq 0$.

Определение. Если существует целое $n \in \mathbf{N}$ такое, что $na = 0$ (и не существует меньшего целого $r \in \mathbf{N}$ такого, что $ra = 0$), то $n$ называют *аддитивным порядком* $a$. Если существует $m \in \mathbf{N}$ такое, что $a^m = 1$ (и не существует меньшего $r \in \mathbf{N}$ такого, что $a^r = 1$), то $m$ называют *мультипликативным порядком* $a$. //

Теорема. *Ненулевые элементы поля $F$ имеют один и тот же аддитивный порядок.*

Доказательство. Возьмем $a$, $b \in F\backslash\{0\}$ и предположим, что аддитивные порядки $a$ и $b$ равны $n$ и $m$ соответственно. Тогда

$$nb = n(a * a^{-1}) * b = (na) * (a^{-1} * b) = 0 * a^{-1} * b = 0.$$

Поэтому $m \leqslant n$. Аналогично

$$ma = m(b * b^{-1}) * a = (mb) * (b^{-1} * a) = 0 * b^{-1} * a = 0.$$

Следовательно, $n \leqslant m$, и поэтому $m = n$. //

Определение. Если в поле $F$ все ненулевые элементы имеют аддитивный порядок $n$, то говорят, что $F$ имеет *характеристику* $n$. Если такого аддитивного порядка не существует, то говорят, что поле имеет характеристику, равную 0. //

Если $|F| = m \in \mathbf{N}$, т. е. $F$ имеет $m$ элементов, то говорят, что $F$ *конечно*. Если $F$ имеет характеристику, равную 0, то оно должно быть бесконечным. (См. упражнение 5.3.)

Теорема. *Характеристика любого конечного поля является простым числом.*

Доказательство. Предположим, что конечное поле $F$ имеет характеристику $n$ и $n = p * q$, где $p$, $q < n$ и $p$, $q \in \mathbf{N}$. Возьмем

$$a \in F\backslash\{0\}.$$

Тогда $0 = na = (p * q)a = p(qa)$. Далее $qa \in F$. Поэтому, если $qa = 0$, выполняется соотношение $n \leqslant q$ (поскольку порядок $a$ равен $n$); в противном случае $qa \in F\backslash\{0\}$, порядок $qa$ также равен $n$ и поэтому $n \leqslant p$. Таким образом, в обоих случаях получаем противоречие. Следовательно, таких $p$ и $q$ не существует и $n$ простое. //

Таким образом, мы получили следующий основной результат.

Теорема. *Конечное поле имеет характеристику $p$ (простое число) и $|F| = p^n$ для некоторого $n \in \mathbf{N}$.*

Доказательство. Мы уже знаем, что $F$ имеет характеристику $p$, причем $p$ простое. Пусть $|F| = q$. Если $p = q$, то утверждение теоремы очевидно. В противном случае возьмем элемент $a_1 \in F \setminus \{0\}$ и положим

$$\mathscr{F}_1 = \{y:\ y = na_1;\ n \in \mathbf{N},\ 1 \leqslant n \leqslant p\},\ |\mathscr{F}_1| = p.$$

Рассмотрим теперь элемент $a_2 \in F \setminus \mathscr{F}_1$, и пусть

$$\mathscr{F}_2 = \{y:\ y = na_1 + ma_2;\ m,\ n \in \mathbf{N},\ 1 \leqslant n \leqslant p,\ 1 \leqslant m \leqslant p\}.$$

Если $\mathscr{F}_2 = F$, то процесс заканчивается; в противном случае рассмотрим $a_3 \in F \setminus \mathscr{F}_2$ и т. д. В конце концов (поскольку $F$ конечно) процесс остановится и мы получим совокупность множеств $\mathscr{F}_1$, $\mathscr{F}_2$, ..., $\mathscr{F}_n$ для некоторого $n \in \mathbf{N}$.

Каждый элемент $f$ из $F$ единственным образом представим в виде

$$f = m_1a_1 + m_2a_2 + \ldots + m_na_n,$$

причем $1 \leqslant m_i \leqslant p$ для всех $i = 1, \ldots, n$. (Доказать это в качестве упражнения.) Следовательно, существует $p^n$ таких выражений, и, таким образом, $|F| = p^n$. //

Итак, любое конечное поле должно иметь $p^n$ элементов при некоторых $p$, $n \in \mathbf{N}$ ($p$ простое). На самом деле для любых таких $p$ и $n$ существует поле порядка $p^n$, однако доказать это не совсем просто.

Рассмотрим в качестве примера поле $(\mathbf{Z}_3, *, +)$, где $*$ и $+$ определены в табл. 5.1. Нетрудно показать, что выполнены условия 1—8 из определения поля.

Таблица 5.1

| $*$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 |
| 2 | 0 | 2 | 1 |

| $+$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 1 | 2 |
| 1 | 1 | 2 | 0 |
| 2 | 2 | 0 | 1 |

Для контраста рассмотрим соответствующую табл. 5.2 для $\mathbf{Z}_4$

Таблица 5.2

| $*$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 |
| 2 | 0 | 2 | 0 | 2 |
| 3 | 0 | 3 | 2 | 1 |

| $+$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
| 1 | 1 | 2 | 3 | 0 |
| 2 | 2 | 3 | 0 | 1 |
| 3 | 3 | 0 | 1 | 2 |

Очевидно, что не существует мультипликативного обратного элемента к 2. Поэтому $\mathbf{Z}_4$ с естественной операцией умножения не является полем. Поле порядка $4 = 2^2$ хотя и существует, но не совпадает с $\mathbf{Z}_4$ (см. упражнение 5.3). Конечные поля заданного порядка строятся достаточно сложным образом — на основе теории многочленов над другими структурами (не полями). Они имеют важное значение в теории кодирования. Мы не будем этим заниматься, поскольку такое приложение является довольно специальным.

**3.4. Упорядоченные поля.** Мы уже видели, что множество $\mathbf{R}$ вместе с обычными операциями сложения и умножения определяет поле. Однако структура поля сама по себе не дает какого-либо упорядочения элементов, которое мы обычно связываем с $\mathbf{R}$. Не все поля могут быть упорядочены. Поэтому мы должны проверить, какие дополнительные условия должны быть выполнены, прежде чем рассматривать это понятие. Обычные свойства порядка можно получить, как и ожидается, многими способами. Начнем с определения свойства положительности. Оно непосредственно приводит к отношению порядка в поле, а затем к понятию длины. Основные результаты этого параграфа будут сформулированы в двух теоремах, а вспомогательные результаты даны в виде упражнений.

Определение. Говорят, что поле $F$ *упорядочено*, если оно содержит непустое подмножество $P$, которое замкнуто относительно операций сложения и умножения, и такое, что для каждого элемента $x$ из $F$ имеет место ровно одно из соотношений

$$x \in P\backslash\{0\},\ x = 0,\ -x \in P\backslash\{0\},$$

$P$ есть множество всех *положительных элементов* $F$. (На этом этапе отметим, что 0 может как включаться в $P$, так и не включаться. Для определенности выберем случай, когда 0 включается в $P$, что согласуется с изложением предыдущей части книги.)

Если $x \in P$, то будем говорить, что элемент $x$ *положителен*, и обозначать этот факт как $x \geqslant 0$. Если $-x \in P\backslash\{0\}$, то будем говорить, что элемент $x$ *отрицателен*, и обозначать это как $x < 0$. Аналогично, если $x$ и $y$ — элементы $F$, то будем говорить, что $x$ *меньше или равно* $y$ (обозначается $x \leqslant y$), тогда и только тогда, когда $y - x \in P$, и что $x$ *меньше* $y$ $(x < y)$, тогда и только тогда когда $y - x \in P\backslash\{0\}$. Символы $<$, $\leqslant$ можно так-

же использовать для записи в противоположную сторону. //

Из этих определений можно получить, что «$\leqslant$» является отношением порядка и выполняются все ожидаемые свойства.

Теорема. *Пусть $F$ — упорядоченное поле и $a$, $b$, $c$, $d \in F$. Тогда:*

а) *если $a \leqslant b$ и $b \leqslant c$, то $a \leqslant c$;*

б) $a \leqslant a$;

в) *если $a \leqslant b$ и $b \leqslant a$, то $a = b$;*

г) *если $a \neq 0$, то $a^2 > 0$;*

д) $1 > 0$;

е) *если $a \leqslant b$, то $a + c \leqslant b + c$;*

ж) *если $a \leqslant b$ и $c \leqslant d$, то $a + c \leqslant b + d$;*

з) *если $a \leqslant b$, $0 < c$ и $d < 0$, то $a * c \leqslant b * c$, $b * d \leqslant a * d$;*

и) *если $0 < a$, то $0 < a^{-1}$, и если $b < 0$, то $b^{-1} < 0$.*

Доказательство.

а) $a \leqslant b$ и $b \leqslant c \Rightarrow (b - a) \in P$, и по определению $(c - b) \in P$. Тогда $c - a = c - b + b - a \in P$, поскольку $P$ замкнуто относительно сложения. Следовательно, $a \leqslant c$.

б) Очевидно, что $a - a = 0 \in P$.

в) $a \leqslant b$ и $b \leqslant a$; поэтому если $b - a = x$, то $x \in P$ и $-x \in P$, что противоречит определению $P$. Поэтому $x = 0$ и $a = b$.

г) Если $a \neq 0$, то или $a \in P$, или $-a \in P$. Поскольку $P$ замкнуто относительно умножения, то $a \in P \Rightarrow a^2 \in P$ и, как показано в п. 3.2,

$$(-a) * (-a) = a^2, \quad \text{откуда} \quad -a \in P \Rightarrow a^2 \in P.$$

д) $1^2 = 1$, поэтому из случая г) следует, что $1 > 0$.

е) Утверждение непосредственно следует из соотношения

$$b - a = (b + c) - (a + c).$$

ж) Требуемый результат получаем из соотношения

$$(b - a) + (d - c) = (b + d) - (a + c),$$

используя замкнутость $P$.

з) Утверждение доказывается аналогично, используя следующие соотношения:

$$(b-a)*c=b*c-a*c$$

и

$$(b-a)*(-d)=a*d-b*d.$$

и) $0<a\Rightarrow a\in P\backslash\{0\}$. Если $a^{-1}=0$, то $1=a*a^{-1}=a*0=0$, а если $a^{-1}\notin P$, то из случая з) имеем $1=a^{-1}*a<0*a=0$. В обоих случаях получили противоречие. Поэтому $a^{-1}\in P\backslash\{0\}$ и, следовательно, $0<a^{-1}$. Оставшаяся часть доказательства проводится аналогично. //

Получим похожие соотношения для понятия величины в упорядоченных полях.

Определение. Если $F$ — упорядоченное поле, то *абсолютным значением* (*величиной*, *длиной* или *модулем*) называется функция

$$x\mapsto\begin{cases}x, & \text{если } x\geqslant 0\\ -x, & \text{если } x<0.\end{cases}$$

Традиционно эту функцию обозначают как $|x|$ ($x$ — аргумент) и читают это как «модуль $x$». //

Сформулируем без доказательств основные результаты, относящиеся к функции $|x|$. (Доказательства оставляем в качестве упражнения.)

Теорема. *Если $F$ — упорядоченное поле и $a$, $b\in F$, то*:

а) $|a|=0$ *тогда и только тогда, когда* $a=0$;

б) $|-a|=|a|$;

в) $|a*b|=|a|*|b|$;

г) *если* $0\leqslant b$, *то* $|a|\leqslant b$ *тогда и только тогда, когда* $-b\leqslant a\leqslant b$;

д) $-|a|\leqslant a\leqslant|a|$;

е) $||a|-|b||\leqslant|a\pm b|\leqslant|a|+|b|$ (*неравенство треугольника*). //

Упражнение 5.3.

1. Доказать, что в кольце $(R, *, +)$ выполняются соотношения:

а) $0*a=a*0=0$;

б) $a*(-b)=(-a)*b=-(a*b)$;

в) $(-a)*(-b)=a*b$.

2. Показать, что если в кольце $(R, *, +)$ для каждого $a \in R$ выполняется соотношение $a * a = a$, то $R$ коммутативно.

3. Показать, что в кольце $\mathbf{Z}_n$ делителями нуля являются только те элементы, которые имеют общие нетривиальные множители с $n$. Следовательно, $\mathbf{Z}_p$, где $p$ простое, не имеет делителей нуля.

4. Показать, что каждая конечная область целостности является полем.

5. Показать, что $(\mathbf{Z}, *, +)$ — область целостности, но не поле.

6. Пусть $p < q$ и $(\mathbf{Z}_p, *_p, +_p)$ и $(\mathbf{Z}_q, *_q, +_q)$ — обычные системы по модулю $p$ и $q$. Доказать, что, хотя они и являются коммутативными кольцами и $\mathbf{Z}_p \subset \mathbf{Z}_q$ кольцо $(\mathbf{Z}_p, *_p, +_p)$ не является подкольцом $(\mathbf{Z}_q, *_q, +_q)$. Показать, что операции $*_q$ и $+_q$ не замкнуты на $\mathbf{Z}_p$.

7. Без использования теорем доказать, что $(\mathbf{Z}_6, *, +)$ не является полем ($*$ и $+$ суть операции по модулю 6).

8. Доказать, что конечное поле имеет ненулевую характеристику и что поле с характеристикой 0 бесконечно.

9. Пусть $a_1, a_2, \ldots, a_n$ определены так же, как и при доказательстве последней теоремы п. 3.3. Доказать, что любое выражение вида

$$m_1a_1 + m_2a_2 + \ldots + m_na_n$$

определяет некоторый элемент поля, причем такое представление единственно.

10. В поле $(F, *, +)$ с операциями, определенными ниже, решить систему линейных уравнений

$$x + d * y = c,$$
$$x * d + y = b,$$

| $*$ | $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|---|
| $a$ | $a$ | $a$ | $a$ | $a$ |
| $b$ | $a$ | $b$ | $c$ | $d$ |
| $c$ | $a$ | $c$ | $d$ | $b$ |
| $d$ | $a$ | $d$ | $b$ | $c$ |

| $+$ | $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|---|
| $a$ | $a$ | $b$ | $c$ | $d$ |
| $b$ | $b$ | $a$ | $d$ | $c$ |
| $c$ | $c$ | $d$ | $a$ | $b$ |
| $d$ | $d$ | $c$ | $b$ | $a$ |

11. Доказать, что если $a * b > 0$, $a, b \in F$, то или $a, b > 0$, или $a, b < 0$.

12. Доказать, что в упорядоченном поле $a^2 + b^2 = 0$ тогда и только тогда, когда $a = b = 0$.

13. Пусть $F$ — упорядоченное поле и $a, b \in F$ такие, что $0 \leqslant a \leqslant b$. Доказать, что $a^2 \leqslant b^2$.

14. Доказать, что каждое поле является областью целостности.

15. В упорядоченном поле, складывая неравенства

$$-|a| \leqslant a \leqslant |a| \text{ и } -|a| \leqslant -a \leqslant |a|,$$

доказать, что:

а) $|a \pm b| \leqslant |a| + |b|$;

б) $||a| - |b|| \leqslant |a \pm b|$.

## § 4. Линейная алгебра

В большинстве элементарных учебников векторы определяют как объекты, обладающие «величиной» и «направлением». Такой подход берет начало из приложений в геометрии и физике. Эти вопросы формально будут обсуждаться в п. 4.2. Дадим более общее определение вектора, для которого понятия величины и направления несущественны.

**4.1. Векторные пространства и линейные преобразования.**

Определение. Пусть $F$ — поле, а $V$ — множество с бинарной операцией $+$. Предположим, что для каждого $a \in F$ и $\mathbf{x} \in V$ определен элемент $a\mathbf{x} \in V$. Тогда, если выполнены аксиомы:

а) $(V, +)$ — коммутативная группа;

б) для всех $\mathbf{x}, \mathbf{y} \in V$ и $a, b \in F$

$$(a + b)\mathbf{x} = a\mathbf{x} + b\mathbf{x},$$

$$a(\mathbf{x} + \mathbf{y}) = a\mathbf{x} + a\mathbf{y},$$

$$(ab)\mathbf{x} = a(b\mathbf{x}),$$

$$1_F\mathbf{x} = \mathbf{x},$$

где $1_F$ — мультипликативная единица в $F$, то говорят, что $V$ является *векторным пространством* над $F$. Элементы $V$ называются *векторами*, операция $+$ называется *сложением векторов*, а отображение

$$\Lambda\colon F \times V \to V,$$

определяемое соотношением $\Lambda(a, \mathbf{x}) = a\mathbf{x}$, называют *умножением вектора на скаляр*. //

Векторное пространство над $F$ может рассматриваться как тройка $(V, +, \Lambda)$, удовлетворяющая приведенным выше аксиомам. Нуль векторного пространства по сложе-

нию обозначают символом 0. Из аксиом следует, что

$$0_F \mathbf{x} = \mathbf{0} \quad \text{для всех} \quad \mathbf{x} \in V,$$

где $0_F$ — аддитивная единица в $F$, и

$$a\mathbf{0} = \mathbf{0} \quad \text{для всех} \quad a \in F.$$

В следующих примерах будет показано, что различные классы множеств обладают структурой векторного пространства.

Пример 4.1.

1. $F^n$ $(n \in \mathbf{N})$ является векторным пространством над $F$ с операциями

$$(a_1, \ldots, a_n) + (b_1, \ldots, b_n) = (a_1 + b_1, \ldots, a_n + b_n),$$
$$a(a_1, \ldots, a_n) = (aa_1, \ldots, aa_n).$$

Нулем $F^n$ является вектор $(0_F, \ldots, 0_F)$. Элементы $a_1, \ldots, a_n$ называются *компонентами* вектора $\mathbf{a} = (a_1, \ldots, a_n)$.

2. Пусть $\mathscr{F}$ — множество всех отображений $f$: $[a, b] \to \mathbf{R}$. Тогда $\mathscr{F}$ является векторным пространством над $\mathbf{R}$ с операциями

$$(f + g)(x) = f(x) + g(x) \quad \text{для всех} \quad f, g \in \mathscr{F},$$
$$(af)(x) = af(x) \quad \text{для всех} \quad a \in \mathbf{R}.$$

3. Пусть $\mathscr{C}$, $\mathscr{C} \subset \mathscr{F}$ — множество всех непрерывных отображений из $\mathscr{F}$. Тогда $\mathscr{C}$ является векторным пространством с операциями, определенными в $\mathscr{F}$. //

Множество $U$, $U \subseteq V$, называется *векторным подпространством* пространства $V$, если оно является векторным пространством с операциями из $V$.

Множество $\{(a_1, \ldots, a_{n-1}, 0_F)\colon a_i \in F\}$ является векторным подпространством пространства $F^n$; $\mathscr{C}$ является векторным подпространством пространства $\mathscr{F}$. Если $U$, $U \subseteq V$,— векторное подпространство пространства $V$, то $\mathbf{0} \in U$.

Векторные пространства $\mathbf{R}^n$ $(1 \leqslant n \leqslant 4)$ возникнут естественным образом в гл. 10. Операции в $\mathbf{R}^n$ имеют геометрическую интерпретацию. Для пространства $\mathbf{R}^2$ это показано на рис. 5.3. Если $\mathbf{r} = (x, y) \in \mathbf{R}^2$, то компоненты $x$ и $y$ измеряются вдоль ортогональных линий, начиная с точки пересечения $O$ (рис. 5.3, *а*). Компоненты $x$ и $y$ откладываются вдоль линий $OX$ (ось $x$) и $OY$ (ось $y$) соответственно. Эти линии проведены под углом 90°

друг к другу, и угол между ними измеряется против часовой стрелки от оси $OX$. Такую систему осей называют *правосторонней* системой координат в $\mathbf{R}^2$. Векторное сложение в $\mathbf{R}^2$ геометрически соответствует правилу параллелограмма, как это показано на рис. 5.3, *с*.

Геометрия векторных пространств $\mathbf{R}^n$ будет рассматриваться ниже, а сейчас мы введем понятия базиса и

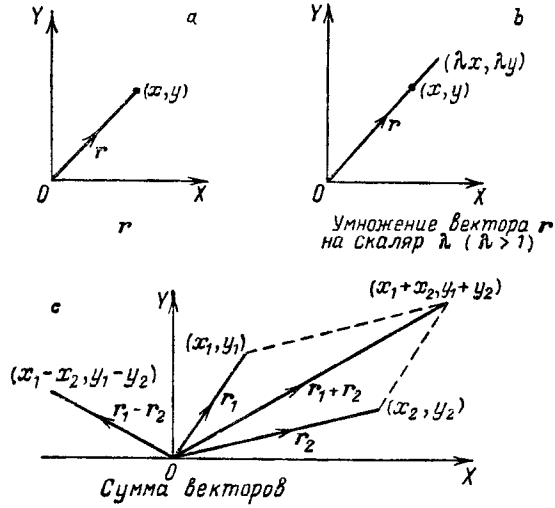



Рис. 5.3

размерности. Если $V$ — векторное пространство над $F$ и $S \subseteq V$, то сумму вида

$$\sum_{i=1}^{n} a_i \mathbf{x}_i, \qquad a_i \in F, \qquad \mathbf{x}_i \in S,$$

называют *линейной комбинацией* векторов из $S$. Говорят, что конечное множество векторов $\{\mathbf{x}_i\colon\ 1 \leqslant i \leqslant k\}$ является *линейно независимым*, если

$$\sum_{i=1}^{k} a_i \mathbf{x}_i = \mathbf{0} \Rightarrow a_1 = a_2 = \ldots = a_k = 0_F;$$

в противном случае множество является *линейно зависимым*. Подмножество $S \subseteq V$ такое, что любой элемент $V$ представим в виде линейной комбинации элементов из $S$, называется *порождающим множеством* пространства $V$ (или же еще говорят, что $S$ порождает $V$). Упорядо-

ченное линейно независимое порождающее множество пространства $V$ называется *базисом* этого пространства.

Пример 4.2. В $\mathbf{R}^3$ вектор $(5, 5, \sqrt{2})$ является линейной комбинацией векторов $(1, 1, 0)$ и $(0, 0, 3)$, так как

$$(5, 5, \sqrt{2}) = 5(1, 1, 0) + \frac{\sqrt{2}}{3}(0, 0, 3).$$

Множество $L = \{(1, 1, 0), (0, 0, 3)\}$ является линейно независимым подмножеством в $\mathbf{R}^3$, так как $a(1, 1, 0) + b(0, 0, 3) = (a, a, 3b) = \mathbf{0}$ тогда и только тогда, когда $a = 0$ и $b = 0$. Однако подмножество $L$ не является базисом, поскольку оно только определяет векторное подпространство

$$\{(x, x, y): x, y \in \mathbf{R}\} \subset \mathbf{R}^3.$$

Базис $B = L \cup \{(1, 0, 0)\}$ «расширяет» $L$ до базиса в $\mathbf{R}^3$. //

Легко показать, что каждый элемент векторного пространства имеет *единственное* представление в фиксированном базисе, так как если $V$ имеет базис $B = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ и

$$\mathbf{x} = \sum_{i=1}^{n} a_i \mathbf{e}_i = \sum_{i=1}^{n} b_i \mathbf{e}_i,$$

то

$$\mathbf{0} = \mathbf{x} - \mathbf{x} = \sum_{i=1}^{n} (b_i - a_i)\mathbf{e}_i.$$

Однако $B$ — линейно независимое множество. Поэтому $b_i = a_i$ для всех $i$, $1 \leqslant i \leqslant n$.

Докажем следующий важный результат.

Предложение. *Пусть* $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ — *порождающее множество пространства* $V$, *а* $L = \{\mathbf{y}_1, \ldots, \mathbf{y}_l\}$ — *линейно независимое множество векторов из* $V$. *Тогда* $m \geqslant l$.

Доказательство. Предположим, что $m < l$. Так как $S$ порождает $V$, то существуют элементы $a_1, \ldots, a_m \in F$ такие, что

$$\mathbf{y}_1 = a_1\mathbf{x}_1 + \ldots + a_m\mathbf{x}_m.$$

Однако $\mathbf{y}_1 \neq \mathbf{0}$, так как $L$ — линейно независимое множество (см. упражнение 5.4), и, следовательно, не все $a_1, \ldots, a_m$ равны нулю. Для определенности положим

$a_1 \neq 0_F$. Тогда

$$\mathbf{x}_1 = a_1^{-1}\mathbf{y}_1 - a_1^{-1}a_2\mathbf{x}_2 - \ldots - a_1^{-1}a_m\mathbf{x}_m,$$

т. е. $\mathbf{x}_1$ является линейной комбинацией $\{\mathbf{y}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$. Так как $S$ порождает $V$, то по доказанному выше множество векторов $\{\mathbf{y}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ также порождает $V$. Аналогично получаем, что $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_3, \ldots, \mathbf{x}_m\}$ порождает $V$. Повторяя этот процесс $m$ раз, получаем, что множество $\{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$ порождает $V$. Следовательно,

$$\mathbf{y}_{m+1} = p_1\mathbf{y}_1 + p_2\mathbf{y}_2 + \ldots + p_m\mathbf{y}_m,$$

где $p_1, \ldots, p_m \in F$ не все равны нулю (так как $\mathbf{y}_{m+1}$ не может быть равно нулю). Отсюда

$$\mathbf{y}_{m+1} - p_1\mathbf{y}_1 - p_2\mathbf{y}_2 - \ldots - p_m\mathbf{y}_m = \mathbf{0},$$

однако последнее невозможно, потому что $\{\mathbf{y}_1, \ldots, \mathbf{y}_l\}$ — линейно независимое множество векторов. Следовательно, $m \geqslant l$. //

Предложение. *Пусть $B$ и $B'$ — базисы векторного пространства $V$ над $F$. Тогда $|B| = |B'|$.*

Доказательство. Пусть $B = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ и $B' = \{\mathbf{e}'_1, \ldots, \mathbf{e}'_m\}$. Тогда из предыдущего предложения следует, что $n \geqslant m$ и $m \geqslant n$, т. е. $m = n$. //

Мощность базиса векторного пространства $V$ называется *размерностью* $V$ и обозначается через $\dim(V)$.

Предложение. $\dim(F^n) = n$.

Доказательство. Определим $B = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$, где

$$\mathbf{e}_i = (0, \ldots, 0, \underset{i\text{-й разряд}}{1_F}, 0, \ldots, 0),$$

и покажем, что $B$ является базисом в $F^n$. Очевидно, что

$$(a_1, \ldots, a_n) = \sum_{i=1}^{n} a_i\mathbf{e}_i;$$

поэтому $B$ порождает $F^n$ и

$$\sum_{i=1}^{n} b_i\mathbf{e}_i = \mathbf{0} \Rightarrow (b_1, \ldots, b_n) = \mathbf{0} \Rightarrow b_1 = 0_F,\ b_2 = 0_F, \ldots, b_n = 0_F.$$

Следовательно, $B$ является базисом в $F^n$ и $\dim(V) = |B| = n$. //

Из данного выше определения следует, что базис всегда состоит из конечного числа векторов, и не во всяких векторных пространствах можно выделить базис (например, в $\mathscr{F}$, $\mathscr{C}$ нет базиса). Понятия базиса и размер-

ности можно расширить на все векторные пространства, однако такое обобщение нам не потребуется. Если пространство $V$ имеет базис, соответствующий данному выше определению, то говорят, что пространство имеет *конечную размерность*, а само пространство называется *конечномерным векторным пространством.*

Рассмотрим теперь гомоморфные отображения между векторными пространствами.

Определение. Пусть $V_1$ и $V_2$ — векторные пространства над полем $F$. Говорят, что отображение $T$: $V_1 \to V_2$ *линейно*, если

$$T(\mathrm{x}+\mathrm{y}) = T\mathrm{x} + T\mathrm{y}, \quad T(a\mathrm{x}) = a(T\mathrm{x}).$$

Если $V_2 = V_1$, то $T$ называют линейным преобразованием пространства $V_1$. //

Далее нас будут интересовать конечномерные векторные пространства над $\mathbf{R}$ и линейные преобразования над ними. В оставшейся части главы через $V$ будем обозначать векторное пространство, а через $\mathrm{End}(V)$ — множество всех линейных преобразований $V$ (эндоморфизмов $V$). Заметим, что большинство приводимых утверждений можно представить в более общем виде.

Перейдем от алгебры $V$ к алгебре $\mathrm{End}(V)$ и покажем, что $\mathrm{End}(V)$ замкнуто по отношению к естественным операциям сложения, умножения и умножения на скаляр. Вначале заметим, что единичное отображение $I_V$ и нулевое отображение $0_V$ являются линейными на $V$, так как по определению

$$I_V\mathrm{x} = \mathrm{x} \quad \text{для всех} \quad \mathrm{x} \in V,$$
$$0_V\mathrm{x} = 0 \quad \text{для всех} \quad \mathrm{x} \in V.$$

Следовательно, для всех $\mathrm{x}, \mathrm{y} \in V$ и $\lambda \in \mathbf{R}$ имеем

$$I_V(\mathrm{x}+\mathrm{y}) = \mathrm{x}+\mathrm{y} = I_V\mathrm{x} + I_V\mathrm{y},$$
$$I_V(\lambda\mathrm{x}) = \lambda\mathrm{x} = \lambda(I_V\mathrm{x}),$$
$$0_V(\mathrm{x}+\mathrm{y}) = 0 = 0+0 = 0_V\mathrm{x} + 0_V\mathrm{y},$$
$$0_V(\lambda\mathrm{x}) = 0 = \lambda 0 = \lambda 0_V\mathrm{x}.$$

Если $S, T \in \mathrm{End}(V)$, то *сумма* $S+T$ и *произведение* $S \circ T$ (*композиция*) определяются формулами

$$(S+T)\mathrm{x} = S\mathrm{x} + T\mathrm{x} \quad \text{для всех} \quad \mathrm{x} \in V$$
$$(S \circ T)\mathrm{x} = S(T\mathrm{x}) \quad \text{для всех} \quad \mathrm{x} \in V.$$

Отметим следующие свойства $\mathrm{End}(V)$ относительно приведенных выше операций.

**Предложение.** *Множество* $(\mathrm{End}(V), \circ, +)$ *является кольцом с единицей.*

**Доказательство.** Укажем основные этапы доказательства. Надо показать, что:

(I) $S, T \in \mathrm{End}(V) \Rightarrow S+T \in \mathrm{End}(V)$ и $S \circ T \in \mathrm{End}(V)$;

(II) $(\mathrm{End}(V), +)$ — коммутативная группа.

Если $S, T, U \in \mathrm{End}(V)$, то:

(III) $S \circ (T \circ U) = (S \circ T) \circ U$;

(IV) $S \circ (T+U) = S \circ T + S \circ U$;

(V) $I_V \circ T = T \circ I_V = T$.

Имеем

(I) $(S+T)(\mathrm{x}+\mathrm{y}) = S(\mathrm{x}+\mathrm{y}) + T(\mathrm{x}+\mathrm{y}) = S\mathrm{x} + S\mathrm{y} + T\mathrm{x} + T\mathrm{y} = (S\mathrm{x}+T\mathrm{x}) + (S\mathrm{y}+T\mathrm{y}) = (S+T)\mathrm{x} + (S+T)\mathrm{y}$.

Аналогично

$$(S+T)(\lambda \mathrm{x}) = S\lambda\mathrm{x} + T\lambda\mathrm{x} = \lambda S\mathrm{x} + \lambda T\mathrm{x} = \lambda(S\mathrm{x}+T\mathrm{x}) = \lambda(S+T)\mathrm{x}.$$

Доказательство того, что $S \circ T \in \mathrm{End}(V)$, оставляем в качестве упражнения.

(II) $(S+(T+U))\mathrm{x} = S\mathrm{x} + (T+U)\mathrm{x} = S\mathrm{x} + (T\mathrm{x}+U\mathrm{x}) = (S\mathrm{x}+T\mathrm{x}) + U\mathrm{x} = (S+T)\mathrm{x} + U\mathrm{x} = ((S+T)+U)\mathrm{x}$.

Следовательно, операция $+$ ассоциативна. Элемент $0_V \in \mathrm{End}(V)$ удовлетворяет условию

$$T + 0_V = 0_V + T = T \quad \text{для всех} \quad T \in \mathrm{End}(V)$$

и является аддитивной единицей $\mathrm{End}(V)$. Для $T \in \mathrm{End}(V)$ определим отображение $-T\colon V \to V$ соотношением

$$(-T)\mathrm{x} = -(T\mathrm{x}) \quad \text{для всех} \quad \mathrm{x} \in V.$$

Легко показать, что $-T \in \mathrm{End}(V)$ и

$$(-T+T) = T + (-T) = 0_V.$$

Поэтому отображение $-T$ является аддитивным, обратным к $T$. Коммутативность $(\mathrm{End}(V), +)$ следует из коммутативности $(V, +)$.

(III) Утверждение следует из результатов гл. 3.

(IV) Для $\mathbf{x} \in V$ имеем

$$(S \circ (T+U))\mathbf{x} = S((T+U)\mathbf{x}) = S(T\mathbf{x}+U\mathbf{x}) =$$
$$= S(T\mathbf{x}) + S(U\mathbf{x}) = (S \circ T)\mathbf{x} + (S \circ U)\mathbf{x} =$$
$$= (S \circ T + S \circ U)\mathbf{x}.$$

(V) Утверждение очевидно. //

Пусть $T \in \mathrm{End}(V)$ и $\lambda \in \mathbf{R}$. Определим отображение $\lambda T$: $V \to V$ следующим образом:

$$(\lambda T)\mathbf{x} = \lambda(T\mathbf{x}) \quad \text{для всех} \quad \mathbf{x} \in V.$$

Легко показать, что $\lambda T \in \mathrm{End}(V)$. Отображение

$$\Lambda\colon \mathbf{R} \times \mathrm{End}(V) \to \mathrm{End}(V),$$

определяемое соотношением $\Lambda(\lambda, T) = \lambda T$, называют *умножением на скаляр*.

Предложение. $(\mathrm{End}(V), +, \Lambda)$ — *векторное пространство над* $\mathbf{R}$.

Доказательство. Из предыдущего утверждения следует, что $(\mathrm{End}(V), +)$ — коммутативная группа; следовательно, нам надо показать, что умножение на скаляр удовлетворяет условиям

$$(\lambda + \mu)T = \lambda T + \mu T, \quad \lambda(S+T) = \lambda S + \lambda T,$$
$$(\lambda\mu)T = \lambda(\mu T), \quad 1_{\mathbf{R}} T = T,$$

где $\lambda, \mu \in \mathbf{R}$ и $S, T \in \mathrm{End}(V)$. Имеем цепочку соотношений

$$((\lambda+\mu)T)\mathbf{x} = (\lambda+\mu)(T\mathbf{x}) = \lambda(T\mathbf{x}) + \mu(T\mathbf{x}) =$$
$$= (\lambda T)\mathbf{x} + (\mu T)\mathbf{x}.$$

Остальные соотношения доказываются аналогично. //

Предложение. *Операции умножения в кольце и умножения на скаляр* $\Lambda$ *в* $\mathrm{End}(V)$ *удовлетворяют соотношению*

$$\lambda(S \circ T) = (\lambda S) \circ T = S \circ (\lambda T),$$

*где* $\lambda \in \mathbf{R}$ *и* $S, T \in \mathrm{End}(V)$.

Доказательство.

$$[(\lambda(S \circ T))\mathbf{x} = \lambda((S \circ T)\mathbf{x}) = \lambda(S(T\mathbf{x})) =$$
$$= (\lambda S)(T\mathbf{x}) = ((\lambda S) \circ T)\mathbf{x},$$
$$(\lambda(S \circ T))\mathbf{x} = \lambda((S \circ T)\mathbf{x}) = \lambda(S(T\mathbf{x})) =$$
$$= S(\lambda(T\mathbf{x})) = (S \circ (\lambda T))\mathbf{x}. \ //$$

Алгебраические структуры, удовлетворяющие таким же свойствам, как и $\mathrm{End}(V)$, называют линейными алгебрами. Дадим строгое определение.

Определение. Четверка $(X, +, \circ, \Lambda)$ называется *линейной алгеброй* над $\mathbf{R}$, если $\Lambda$: $\mathbf{R} \times X \to X$ и

(I) $(X, +, \Lambda)$ — векторное пространство над $\mathbf{R}$;

(II) $(X, \circ, +)$ — кольцо;

(III) $\Lambda$ и $\circ$ удовлетворяют условиям

$$\lambda(\mathbf{x}_1 \circ \mathbf{x}_2) = (\lambda \mathbf{x}_1) \circ \mathbf{x}_2 = \mathbf{x}_1 \circ (\lambda \mathbf{x}_2)$$

для всех $\lambda \in \mathbf{R}$ и $\mathbf{x}_1, \mathbf{x}_2 \in X$. //

Результаты, полученные для $\mathrm{End}(V)$, можно сформулировать следующим образом.

Предложение. $\mathrm{End}(V)$ *с введенными выше операциями является линейной алгеброй с мультипликативной единицей.* //

Если $T \in \mathrm{End}(V)$ и существует преобразование $S$: $V \to V$ такое, что

$$S \circ T = T \circ S = I_V,$$

то (см. упражнение 5.4) $S \in \mathrm{End}(V)$. Тогда $T$ называют *обратимым*, а $S = T^{-1}$ — *обратным к* $T$ *преобразованием*. Обозначим через $\mathrm{Aut}(V)$ множество всех обратимых преобразований из $\mathrm{End}(V)$, т. е. множество автоморфизмов $V$.

Предложение. $(\mathrm{Aut}(V), \circ)$ *является группой.*

Доказательство. Так как $I_V \in \mathrm{Aut}(V)$ и $I_V \circ I_V = I_V$, следовательно, существует $I_V^{-1}$, равное $I_V$. Пусть $S \in \mathrm{Aut}(V)$; тогда

$$S^{-1} \circ S = S \circ S^{-1} = I_V.$$

Поэтому $(S^{-1})^{-1}$ существует и совпадает с $S$. Следовательно, $S^{-1} \in \mathrm{Aut}(V)$. Если теперь $S$, $T \in \mathrm{Aut}(V)$, то

$$(S \circ T) \circ (T^{-1} \circ S^{-1}) = S \circ (T \circ T^{-1}) \circ S^{-1} = S \circ S^{-1} = I_V.$$

Аналогично

$$(T^{-1} \circ S^{-1}) \circ (S \circ T) = I_V.$$

Поэтому $(S \circ T)^{-1}$ существует, и из $S$, $T \in \mathrm{Aut}(V)$ следует, что $S \circ T \in \mathrm{Aut}(V)$. Ассоциативность операции $\circ$ уже доказана. //

**4.2. Структурные изображения в $\mathbf{R}^n$.** Рассмотрим геометрическую интерпретацию пространства $\mathbf{R}^n$, при которой понятия «направление» и «величина» для векторов

имеют геометрический смысл. Вернемся к геометрическому изображению $\mathbf{R}^2$. Мы видим, что если $\mathbf{r}=(x, y)\in \mathbf{R}^2$, то расстояние от точки $(x, y)$ до $(0, 0)$ есть $(x^2+y^2)^{1/2}$. Обозначим это расстояние через $\|\mathbf{r}\|$, которое можно рассматривать как отображение $\|\cdot\|\colon \mathbf{R}^2 \to \mathbf{R}$. Оно называется *длиной*, *модулем* или *нормой*. Рассмотрим точки $\mathbf{r}_1=(x_1, y_1)$ и $\mathbf{r}_2=(x_2, y_2)$

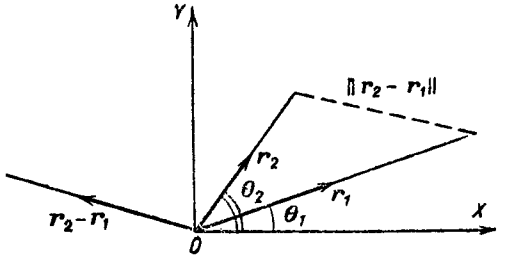



Рис. 5.4

(рис. 5.4). Пусть $\theta_1$ и $\theta_2$ — углы в интервале $[0, \pi]$ между положительной полуосью $OX$ и векторами $\mathbf{r}_1$ и $\mathbf{r}_2$ соответственно. Тогда расстояние между $\mathbf{r}_1$ и $\mathbf{r}_2$ равно $\|\mathbf{r}_2-\mathbf{r}_1\|$, а угол между ними равен $\theta=\theta_2-\theta_1$. Имеем

$$\cos\theta=\cos(\theta_2-\theta_1)=\cos\theta_1\cos\theta_2+\sin\theta_1\sin\theta_2=$$

$$=\frac{x_1}{\|\mathbf{r}_1\|}\frac{x_2}{\|\mathbf{r}_2\|}+\frac{y_1}{\|\mathbf{r}_1\|}\frac{y_2}{\|\mathbf{r}_2\|}=\frac{x_1x_2+y_1y_2}{\|\mathbf{r}_1\|\|\mathbf{r}_2\|}.$$

Выражение $x_1x_2+y_1y_2$ можно использовать для вычисления расстояний и углов в $\mathbf{R}^2$. Определим отображение

$$\Phi\colon \mathbf{R}^2\times\mathbf{R}^2\to\mathbf{R}$$

следующим образом:

$$\Phi(\mathbf{r}_1, \mathbf{r}_2)=x_1x_2+y_1y_2.$$

Тогда

$$\|\mathbf{r}\|=(\Phi(\mathbf{r}, \mathbf{r}))^{1/2},\quad \cos\theta=\frac{\Phi(\mathbf{r}_1, \mathbf{r}_2)}{(\Phi(\mathbf{r}_1, \mathbf{r}_1)\Phi(\mathbf{r}_2, \mathbf{r}_2))^{1/2}}.$$

Угол $\theta$ между двумя векторами в $\mathbf{R}^2$ определяется однозначно при условии, что $0\leqslant\theta\leqslant\pi$. Когда $\theta=0$ или $\theta=\pi$, то говорят, что $r_1$ и $r_2$ *параллельны* (*коллинеарны*). В прикладной математике удобно обозначать $\Phi(\mathbf{r}_1, \mathbf{r}_2)$ через $\mathbf{r}_1\cdot\mathbf{r}_2$ и называть *скалярным произведением* $\mathbf{r}_1$ и $\mathbf{r}_2$. Если $\mathbf{r}_1\neq 0$ и $\mathbf{r}_2\neq 0$, то $\mathbf{r}_1\cdot\mathbf{r}_2=0$ тогда и только тогда, когда $\theta=\pi/2$; в этом случае говорят, что $\mathbf{r}_1$ и $\mathbf{r}_2$ взаимно *ортогональны*, *перпендикулярны* или *нормальны*. Ниже приведены некоторые свойства скалярного произведения.

Предложение.

а) $\mathbf{r}\cdot\mathbf{r}\geqslant 0$ *и* $\mathbf{r}\cdot\mathbf{r}=0$ *тогда и только тогда, когда* $\mathbf{r}=(0,0)$;

б) $\mathbf{r}_1\cdot\mathbf{r}_2=\mathbf{r}_2\cdot\mathbf{r}_1$ *для всех* $\mathbf{r}_1, \mathbf{r}_2\in\mathbf{R}^2$;

в) $\mathbf{r}_1\cdot(\mathbf{r}_2+\mathbf{r}_3)=\mathbf{r}_1\cdot\mathbf{r}_2+\mathbf{r}_1\cdot\mathbf{r}_3$ *для всех* $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\in\mathbf{R}^2$;

г) $\lambda(\mathbf{r}_1\cdot\mathbf{r}_2)=(\lambda\mathbf{r}_1)\cdot\mathbf{r}_2=\mathbf{r}_1\cdot(\lambda\mathbf{r}_2)$, $\mathbf{r}_1, \mathbf{r}_2\in\mathbf{R}^2$, $\lambda\in\mathbf{R}$.

Доказательство.

а) Если $\mathbf{r}=(x, y)\in\mathbf{R}^2$, то $\mathbf{r}\cdot\mathbf{r}=(x^2+y^2)\geqslant 0$ для всех $x, y\in\mathbf{R}$ и $\mathbf{r}\cdot\mathbf{r}=0$ тогда и только тогда, когда $x=0$ и $y=0$.

б) $\mathbf{r}_1\cdot\mathbf{r}_2=x_1x_2+y_1y_2=x_2x_1+y_2y_1=\mathbf{r}_2\cdot\mathbf{r}_1$.

Соотношения в) и г) доказываются аналогично и оставляются в качестве упражнения. //

В более общем случае, если $V$ — векторное пространство над $\mathbf{R}$ и $\Phi$: $V\times V\to\mathbf{R}$ — отображение, удовлетворяющее свойствам а)—г), то $(V, \Phi)$ становится пространством, для которого могут изучаться понятия длины и угла. Отображение $\Phi$ называют *внутренним произведением* для $V$, а $(V, \Phi)$ — *векторным пространством с внутренним произведением*. В частности, если определить $\Phi$: $\mathbf{R}^n\times\mathbf{R}^n\to\mathbf{R}$ $(n\in\mathbf{N})$ соотношением

$$\Phi(\mathbf{a}, \mathbf{b})\equiv\mathbf{a}\cdot\mathbf{b}=\sum_{i=1}^{n}a_ib_i,$$

где $\mathbf{a}=(a_1, \ldots, a_n)$, $\mathbf{b}=(b_1, \ldots, b_n)$, то отображение $\cdot$ будет соблюдать требуемыми свойствами. Определим длину вектора $\mathbf{a}\in\mathbf{R}^n$ как

$$\|\mathbf{a}\|=(\mathbf{a}\cdot\mathbf{a})^{1/2},$$

а косинус угла между двумя векторами $\mathbf{a}$ и $\mathbf{b}$ как

$$\frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|};$$

угол лежит на отрезке $[0, \pi]$.

На $\mathbf{R}^n$ могут быть определены другие внутренние произведения. Внутреннее произведение, введенное выше, называется *обычным* или *евклидовым* внутренним произведением. Оно дает те значения длины и угла, которые ожидались интуитивно.

Когда $n=1$, ясно, что $\cdot$ является лишь умножением в $\mathbf{R}$, и угол между двумя векторами определяют как

$$\arccos\frac{xy}{|x||y|};$$

угол равен или 0, или $\pi$ в зависимости от знака $xy$. Норма $\|\cdot\|$ обобщает понятие модуля $|\cdot|$ в $\mathbf{R}$ и обладает аналогичными свойствами. Например, можно показать, что

$$\|\mathbf{a}\| \geqslant 0 \quad \text{для всех} \quad \mathbf{a} \in \mathbf{R}^n,$$

$$\|\mathbf{a}\| = 0 \quad \text{тогда и только тогда, когда} \quad a = 0,$$

$$\|\lambda \mathbf{a}\| = |\lambda| \|\mathbf{a}\| \quad \text{для всех} \quad a \in \mathbf{R}^n \text{ и } \lambda \in \mathbf{R},$$

$$\|\mathbf{a} + \mathbf{b}\| \leqslant \|\mathbf{a}\| + \|\mathbf{b}\| \quad \text{для всех} \quad \mathbf{a}, \mathbf{b} \in \mathbf{R}^n.$$

Вектор $\mathbf{a} \in \mathbf{R}^n$ такой, что $\|\mathbf{a}\| = 1$ (что эквивалентно $\mathbf{a} \cdot \mathbf{a} = 1$), называется *единичным вектором*. Если $\mathbf{a} \in \mathbf{R}^n \backslash \{0\}$, то $\mathbf{a}/\|\mathbf{a}\|$ — единичный вектор, параллельный $\mathbf{a}$. Единичный вектор обычно обозначается $\widehat{\mathbf{a}}$.

Если $B = \{\widehat{\mathbf{e}}_1, \ldots, \widehat{\mathbf{e}}_n\}$ — базис в $\mathbf{R}^n$ и

$$\mathbf{e}_i \cdot \mathbf{e}_j = \begin{cases} 0, & \text{если } i \neq j, \\ 1, & \text{если } i = j, \end{cases}$$

то базис $B$ называется *ортонормированным*. Ортонормированный базис в $\mathbf{R}^n$, определенный следующим образом:

$$\widehat{\mathbf{e}}_i = (0, \ldots, 0, \underset{\substack{\uparrow \\ i\text{-й разряд}}}{1}, 0, \ldots, 0), \quad 1 \leqslant i \leqslant n,$$

называется *стандартным базисом* в $\mathbf{R}^n$. В $\mathbf{R}^2$ и $\mathbf{R}^3$ стандартные базисы удобно записывать в виде $(\widehat{\mathbf{i}}, \widehat{\mathbf{j}})$ и $(\widehat{\mathbf{i}}, \widehat{\mathbf{j}}, \widehat{\mathbf{k}})$ соответственно. Рассмотрим следующую геометрическую интерпретацию этих базисов. Векторы $\widehat{\mathbf{i}}$ и $\widehat{\mathbf{j}}$ определяют правостороннюю систему осей в $\mathbf{R}^2$, а третья ось $OZ$ перпендикулярна плоскости, содержащей векторы $\widehat{\mathbf{i}}$ и $\widehat{\mathbf{j}}$, и направлена таким образом, чтобы концы векторов $\widehat{\mathbf{i}}$, $\widehat{\mathbf{j}}$ и $\widehat{\mathbf{k}}$ (в указанном порядке) определяли правостороннее движение (рис. 5.5). Это свойство известно как *правило правой руки*. В системах такого типа в $\mathbf{R}^3$ оси называются *правосторонними*.

Определение. Если $\mathbf{a} = (a_1, a_2, a_3) \in \mathbf{R}^3$ и $\mathbf{b} = (b_1, b_2, b_3) \in \mathbf{R}^3$, то *векторным произведением* $\mathbf{a}$ и $\mathbf{b}$ (обозначается $\mathbf{a} \times \mathbf{b}$) по определению называют вектор $\mathbf{a} \times \mathbf{b} = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)$. //

Операция $\times$ может рассматриваться как отображение $\mathbf{R}^3 \times \mathbf{R}^3 \to \mathbf{R}^3$.

**Предложение.** *Если* $\mathbf{a}, \mathbf{b} \in \mathbf{R}^3$, *то*

а) $\|\mathbf{a} \times \mathbf{b}\| = \|\mathbf{a}\| \|\mathbf{b}\| \sin \theta$, *где* $\theta$ — *угол между* $\mathbf{a}$ *и* $\mathbf{b}$;

б) *вектор* $\mathbf{a} \times \mathbf{b}$ *ортогонален векторам* $\mathbf{a}$ *и* $\mathbf{b}$.

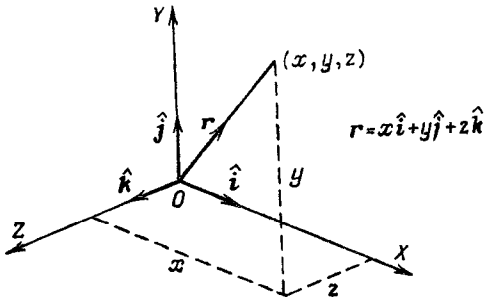



Рис. 5.5

Доказательство.

а) $$\|\mathbf{a} \times \mathbf{b}\|^2 = (a_2 b_3 - a_3 b_2)^2 + (a_3 b_1 - a_1 b_3)^2 + (a_1 b_2 - a_2 b_1)^2 =$$
$$= a_2^2 b_3^2 + a_3^2 b_2^2 - 2a_2 b_3 a_3 b_2 + a_3^2 b_1^2 + a_1^2 b_3^2 -$$
$$- 2a_3 b_1 a_1 b_3 + a_1^2 b_2^2 + a_2^2 b_1^2 - 2a_1 b_2 a_2 b_1 =$$
$$= (a_1^2 + a_2^2 + a_3^2)(b_1^2 + b_2^2 + b_3^2) - (a_1 b_1 + a_2 b_2 + a_3 b_3)^2 =$$
$$= \|\mathbf{a}\|^2 \|\mathbf{b}\|^2 - (\mathbf{a} \cdot \mathbf{b})^2 = \|\mathbf{a}\|^2 \|\mathbf{b}\|^2 \left(1 - \frac{(\mathbf{a} \cdot \mathbf{b})^2}{\|\mathbf{a}\|^2 \|\mathbf{b}\|^2}\right) =$$
$$= \|\mathbf{a}\|^2 \|\mathbf{b}\|^2 (1 - \cos^2 \theta) = \|\mathbf{a}\|^2 \|\mathbf{b}\|^2 \sin^2 \theta.$$

б) Легко показать, что $\mathbf{a} \cdot (\mathbf{a} \times \mathbf{b}) = 0$ и $\mathbf{b} \cdot (\mathbf{a} \times \mathbf{b}) = 0$, откуда и следует требуемый результат. //

Чтобы получить геометрическую интерпретацию $\mathbf{a} \times \mathbf{b}$, заметим, что если

$$\mathbf{a} = (a_1, 0, 0), \quad \mathbf{b} = (b_1, b_2, 0),$$

то

$$\mathbf{a} \times \mathbf{b} = (0, 0, a_1 b_2);$$

поэтому если $a_1 > 0$, то

$$\mathbf{a} \times \mathbf{b} = \begin{cases} |a_1 b_2| \hat{\mathbf{k}} & \text{для } b_2 > 0, \\ -|a_1 b_2| \hat{\mathbf{k}} & \text{для } b_2 < 0, \end{cases}$$

и направление $\mathbf{a} \times \mathbf{b}$ определено таким образом, чтобы выполнялось правило правой руки относительно векторов $\mathbf{a}$, $\mathbf{b}$ и $\mathbf{a} \times \mathbf{b}$. Это правило носит общий характер, поскольку

ку для произвольной пары векторов в правосторонней системе координат всегда можно выбрать способ представления векторов, который определяется векторами **a** и **b**. В результате векторное произведение будет иметь вид

$$\mathbf{a}\times\mathbf{b} = \|\mathbf{a}\|\,\|\mathbf{b}\|\sin\theta\,\hat{\mathbf{n}},$$

где $\hat{\mathbf{n}}$ — единичный вектор, ортогональный **a** и **b**, с направлением, выбираемым по правилу правой руки. Если $\mathbf{a}\times\mathbf{b}=0$, то векторы **a** и **b** линейно зависимы, а если $\|\mathbf{a}\|>0$ и $\|\mathbf{b}\|>0$, то из равенства $\mathbf{a}\times\mathbf{b}=0$ следует, что **a** и **b** параллельны. Пусть **a** и **b** — векторы, изображенные на рис. 5.6. Тогда $\|\mathbf{a}\times\mathbf{b}\|$ — площадь параллелограмма *OACB* и $\mathbf{a}\times\mathbf{b}$ может рассматриваться как *вектор* площади.

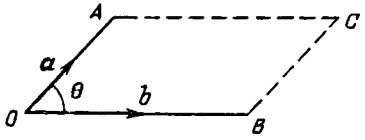


Рис. 5.6

Некоторые свойства векторного произведения приведем ниже; доказательства оставляем в качестве упражнений.

Предложение.

а) $\mathbf{a}\times\mathbf{b} = -\mathbf{b}\times\mathbf{a}$;

б) $\mathbf{a}\times(\mathbf{b}+\mathbf{c}) = \mathbf{a}\times\mathbf{b}+\mathbf{a}\times\mathbf{c}$;

в) $(\lambda\mathbf{a})\times\mathbf{b} = \mathbf{a}\times(\lambda\mathbf{b}) = \lambda(\mathbf{a}\times\mathbf{b})$;

г) $\hat{\mathbf{i}}\times\hat{\mathbf{j}} = \hat{\mathbf{k}}$, $\hat{\mathbf{j}}\times\hat{\mathbf{k}} = \hat{\mathbf{i}}$, $\hat{\mathbf{k}}\times\hat{\mathbf{i}} = \hat{\mathbf{j}}$;

д) $\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = (\mathbf{a}\cdot\mathbf{c})\mathbf{b} - (\mathbf{a}\cdot\mathbf{b})\mathbf{c}$;

е) $\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = \mathbf{b}\cdot(\mathbf{c}\times\mathbf{a}) = \mathbf{c}\cdot(\mathbf{a}\times\mathbf{b}) = -\mathbf{a}\cdot(\mathbf{c}\times\mathbf{b}) = -\mathbf{b}\cdot(\mathbf{a}\times\mathbf{c}) = -\mathbf{c}\cdot(\mathbf{b}\times\mathbf{a})$. //

Выражение $\mathbf{a}\times(\mathbf{b}\times\mathbf{c})$ часто называют *тройным векторным произведением* **a**, **b** и **c**, а $\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})$ — *смешанным произведением*. Геометрически $\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})$ означает объем параллелепипеда с ребрами **a**, **b** и **c**.

Предложение. *Множество* $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}\subset\mathbf{R}^3$ *линейно зависимо тогда и только тогда, когда* $\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})=0$.

Доказательство. Предположим, что **a**, **b** и **c** линейно зависимы. Тогда существуют $\lambda$, $\mu$, $\sigma\in\mathbf{R}$, не все равные нулю, такие, что

$$\lambda\mathbf{a} + \mu\mathbf{b} + \sigma\mathbf{c} = 0.$$

Не ограничивая общности, предположим, что $\lambda \neq 0$. Тогда

$$\mathbf{a} = -\lambda^{-1}(\mu\mathbf{b} + \sigma\mathbf{c}),$$

$$\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c}) = -\lambda^{-1}(\mu\mathbf{b} + \sigma\mathbf{c})\cdot(\mathbf{b}\times\mathbf{c}) =$$

$$= -\lambda^{-1}[\mu\mathbf{b}\cdot(\mathbf{b}\times\mathbf{c}) + \sigma\mathbf{c}\cdot(\mathbf{b}\times\mathbf{c})] = 0.$$

Обратно, если $\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})$, то или

а) один из векторов $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ равен нулю (в этом случае результат очевиден), или

б) вектор $\mathbf{a}$ ортогонален $\mathbf{b}\times\mathbf{c}$.

Однако $\mathbf{b}$ и $\mathbf{c}$ ортогональны к $\mathbf{b}\times\mathbf{c}$; поэтому $\mathbf{a} = \lambda'\mathbf{b} + \mu'\mathbf{c}$ при некоторых $\lambda'$, $\mu' \in \mathbf{R}$ и $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ линейно зависимы. //

Закончим главу кратким рассмотрением вопросов дифференцируемости «векторнозначных» функций. Пусть на $\mathbf{R}^n$ задана обычная норма. Определим производную функции вида

$$f\colon \mathbf{R} \to \mathbf{R}^n.$$

Обобщая одномерный случай, скажем, что $f$ дифференцируема в точке $t$, если существует вектор $\mathbf{F}(t) = (F_1(t), \ldots, F_n(t)) \in \mathbf{R}^n$ такой, что

$$\left\| \frac{\mathbf{f}(t+h) - \mathbf{f}(t)}{h} - \mathbf{F}(t) \right\| \to 0$$

при $h \to 0$, или, что эквивалентно, если $\mathbf{f}$ имеет компоненты $f_1, \ldots, f_n$ такие, что

$$\left\| \left( \frac{f_1(t+h) - f_1(t)}{h} - F_1(t), \ldots, \frac{f_n(t+h) - f_n(t)}{h} - F_n(t) \right) \right\| \to 0$$

при $h \to 0$. Очевидно, что каждая компонента должна стремиться к нулю при $h \to 0$, поэтому $d\mathbf{f}/dt$ существует тогда и только тогда, когда $df_1/dt, \ldots, df_n/dt$ существуют и

$$\frac{d\mathbf{f}}{dt} = \left( \frac{df_1}{dt}, \ldots, \frac{df_n}{dt} \right).$$

Другими словами, чтобы продифференцировать векторнозначную функцию, мы должны продифференцировать ее покомпонентно. Например, если $\mathbf{f}\colon \mathbf{R} \to \mathbf{R}^3$ определена соотношением

$$\mathbf{f}(t) = (2t^2, \ln t, \sin^2 t),$$

то

$$\frac{d\mathbf{f}}{dt} = \left( 4t, \frac{1}{t}, 2\sin t \cos t \right).$$

Пусть $\mathbf{f}\colon \mathbf{R} \to \mathbf{R}^3$ и $\mathbf{g}\colon \mathbf{R} \to \mathbf{R}^3$. Определим функции $\mathbf{f} \cdot \mathbf{g}\colon \mathbf{R} \to \mathbf{R}$ и $\mathbf{f} \times \mathbf{g}\colon \mathbf{R} \to \mathbf{R}^3$. Положим

$$(\mathbf{f} \cdot \mathbf{g})(t) = \mathbf{f}(t) \cdot \mathbf{g}(t), \qquad (\mathbf{f} \times \mathbf{g})(t) = \mathbf{f}(t) \times \mathbf{g}(t).$$

Дифференцирование этих функций производится следующим образом:

$$\frac{d}{dt}(\mathbf{f} \cdot \mathbf{g}) = \mathbf{f} \cdot \frac{d\mathbf{g}}{dt} + \frac{d\mathbf{f}}{dt} \cdot \mathbf{g}, \quad \frac{d}{dt}(\mathbf{f} \times \mathbf{g}) = \mathbf{f} \times \frac{d\mathbf{g}}{dt} + \frac{d\mathbf{f}}{dt} \times \mathbf{g}.$$

Проверку этих формул оставляем в качестве упражнения.

Упражнение 5.4.

1. Показать, что если $V$ — векторное пространство над полем $F$, то

$$0_F \mathbf{x} = \mathbf{0} \quad \text{для всех} \quad \mathbf{x} \in V,$$

$$a\mathbf{0} = \mathbf{0} \quad \text{для всех} \quad a \in F.$$

2. Представить вектор $(a, 1, 3) \in \mathbf{R}^3$, где $a \in \mathbf{R}$, в виде линейной комбинации векторов множества

$$S = \{(1, 1, 0), (0, 2, 0), (0, 0, 4)\}$$

и показать, что $S$ — линейно независимое множество векторов. Является ли $S$ базисом в $\mathbf{R}^3$?

3. Показать, что если $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ — линейно независимое подмножество векторного пространства $V$, то $\mathbf{x}_i \neq \mathbf{0}$ при любом $i$, $1 \leqslant i \leqslant m$.

4. а) Какие из следующих преобразований являются линейными:

$$T_1(x, y) = (a, y),\ a \in \mathbf{R}\backslash\{0\},$$

$$T_2(x, y) = (\lambda x + y, \sigma y),\ \lambda, \sigma \in \mathbf{R}\backslash\{0\},$$

$$T_3(x, y) = (x^2, 0),\ T_4(x, y) = (x, 0)?$$

б) Определить произведения $T_2 \circ T_4$ и $T_4 \circ T_2$.

в) Доказать, что если $T \in \mathrm{End}(V)$, то $T0 = 0$.

5. а) Если $V$ — векторное пространство, то *проекцией* (*проектором*) $V$ называют преобразование $P\colon V \to V$, обладающее свойством

$$(P \circ P)\mathbf{x} = P\mathbf{x} \quad \text{для всех} \quad \mathbf{x} \in V.$$

Доказать, что преобразование $P\colon \mathbf{R}^2 \to \mathbf{R}^2$, определяемое соотношением

$$P(x, y) = \left(\frac{1}{a-b}(ax - y + c), \frac{b}{a-b}(ax - y + c) + c\right)$$

при $a$, $b$, $c \in \mathbf{R}$ и $a \neq b$, является проектором в $\mathbf{R}^2$. При каких условиях $P \in \mathrm{End}(\mathbf{R}^2)$?

б) Какие из определенных в п. 4. а) преобразований являются проекторами?

6. Пусть $T \in \mathrm{End}(V)$. *Нулевым подпространством (ядром)* $T$ называют множество $\mathcal{N}(T)$, определяемое соотношением

$$\mathcal{N}(T) = \{\mathbf{x} \in V:\ T\mathbf{x} = 0\}.$$

Доказать, что $\mathcal{N}(T)$ является векторным подпространством $V$. Доказать также, что образ $T$ является векторным подпространством $V$.

7. Пусть $V$ — векторное пространство над $\mathbf{R}$ в $T \in \mathrm{End}(V)$. Говорят, что $T$ имеет *действительное собственное значение* $\lambda \in \mathbf{R}$, если существует ненулевой вектор $\mathbf{x} \in V$ такой, что

$$T\mathbf{x} = \lambda\mathbf{x};$$

при этом $\mathbf{x}$ называют *собственным вектором* $T$, соответствующим собственному значению $\lambda$. а) Доказать, что если $T \in \mathrm{End}(V)$ такое, что

$$T(x, y) = (x + ay, y),$$

то любой вектор вида $(p, 0)$ при $p \neq 0$ является собственным вектором $T$. Какие у $T$ собственные значения?

б) Пусть $T \in \mathrm{End}(V)$. Обозначим через $V_\lambda$ множество собственных векторов $T$, соответствующих собственному значению $\lambda$. Показать, что $V_\lambda \cup \{0\}$ является векторным подпространством $V$. Доказать аналогичное утверждение для $\mathcal{N}(T)$.

8. Найти собственные значения и собственные векторы преобразований $T_1, T_2 \in \mathrm{End}(\mathbf{R}^2)$:

$$T_1(x, y) = (-y, x),\ T_2(x, y) = (x, -y).$$

Какой геометрический смысл имеют $T_1$ и $T_2$?

9. Доказать, что

а) если $S, T \in \mathrm{End}(V)$, то $S \circ T \in \mathrm{End}(V)$;

б) если при $T \in \mathrm{End}(V)$ существует преобразование $S\colon V \to V$ такое, что

$$S \circ T = T \circ S = I_V,$$

то $S \in \mathrm{End}(V)$;

в) если $T \in \mathrm{Aut}(V)$, то $\mathcal{N}(T) = \{0\}$.

10. Доказать, что если $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3 \in \mathbf{R}^2$ и $\lambda \in \mathbf{R}$, то

а) $\mathbf{r}_1 \cdot (\mathbf{r}_2 + \mathbf{r}_3) = \mathbf{r}_1 \cdot \mathbf{r}_2 + \mathbf{r}_1 \cdot \mathbf{r}_3$;

б) $\lambda(\mathbf{r}_1 \cdot \mathbf{r}_2) = (\lambda \mathbf{r}_1) \cdot \mathbf{r}_2 = \mathbf{r}_1 \cdot (\lambda \mathbf{r}_2)$;

в) $|\mathbf{r}_1 \cdot \mathbf{r}_2| \leqslant \|\mathbf{r}_1\| \, \|\mathbf{r}_2\|$;

г) $\|\mathbf{r}_1 - \mathbf{r}_2\|^2 = \|\mathbf{r}_1\|^2 + \|\mathbf{r}_2\|^2 - 2\|\mathbf{r}_1\| \, \|\mathbf{r}_2\| \cos\theta$,

где $\theta$ — угол между $\mathbf{r}_1$ и $\mathbf{r}_2$;

д) $|\|\mathbf{r}_1\| - \|\mathbf{r}_2\|| \leqslant \|\mathbf{r}_1 + \mathbf{r}_2\| \leqslant \|\mathbf{r}_1\| + \|\mathbf{r}_2\|$

(последнее неравенство известно как *неравенство треугольника*). Дать геометрические иллюстрации этим результатам.

В действительности вышесказанное имеет место для любого пространства со скалярным произведением; в частности, результаты справедливы для $\mathbf{R}^n (n \in \mathbf{N})$ с обычным внутренним (скалярным) произведением.

11. Вычислить единичные векторы, параллельные

а) $\mathbf{a} = (1, 1, 1)$;

б) $\mathbf{b} = (1, p, 0)$, $p \in \mathbf{R}$.

Определить единичный вектор, ортогональный $a$ и $b$ одновременно.

12. Пусть $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbf{R}^3$ и $\lambda \in \mathbf{R}$. Доказать, что

а) $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$;

б) $\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}$;

в) $(\lambda \mathbf{a}) \times \mathbf{b} = \mathbf{a} \times (\lambda \mathbf{b}) = \lambda (\mathbf{a} \times \mathbf{b})$;

г) $\hat{\mathbf{i}} \times \hat{\mathbf{j}} = \hat{\mathbf{k}}$, $\hat{\mathbf{j}} \times \hat{\mathbf{k}} = \hat{\mathbf{i}}$, $\hat{\mathbf{k}} \times \hat{\mathbf{i}} = \hat{\mathbf{j}}$;

д) $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}$;

е) $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b}) = -\mathbf{a} \cdot (\mathbf{c} \times \mathbf{b}) =$
$= -\mathbf{b} \cdot (\mathbf{a} \times \mathbf{c}) = -\mathbf{c} \cdot (\mathbf{b} \times \mathbf{a})$.

13. Используя результаты 5.4, 12, доказать, что операция $\times$: $\mathbf{R}^3 \times \mathbf{R}^3 \to \mathbf{R}^3$ не ассоциативна, т. е. в общем случае $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) \neq (\mathbf{a} \times \mathbf{b}) \times \mathbf{c}$.

14. Пусть $\mathbf{f}, \mathbf{g}$: $\mathbf{R} \to \mathbf{R}^3$. Доказать, что

а) $\frac{d}{dt}(\mathbf{f} \cdot \mathbf{g}) = \mathbf{f} \cdot \frac{d\mathbf{g}}{dt} + \frac{d\mathbf{f}}{dt} \cdot \mathbf{g}$;

б) $\frac{d}{dt}(\mathbf{f} \times \mathbf{g}) = \mathbf{f} \times \frac{d\mathbf{g}}{dt} + \frac{d\mathbf{f}}{dt} \times \mathbf{g} = -\frac{d\mathbf{g}}{dt} \times \mathbf{f} - \mathbf{g} \times \frac{d\mathbf{f}}{dt}$;

в) если $\|\mathbf{f}(t)\| = a$ для всех $t$, где $a \in \mathbf{R}$ — постоянная, то $d\mathbf{f}/dt$ ортогонален к $\mathbf{f}$ при всех $t$. Провести вычисления при

$$\mathbf{f}(t) = (a \cos t, \, a \sin t, \, 0), \quad \mathbf{g}(t) = (0, \, 1, \, t).$$

## § 5. Решетки и булевы алгебры

Булева алгебра является одним из математических объектов, с которыми мы уже сталкивались во введении. Мы покажем, что существует не одна булева алгебра, а много. Следовательно, любое преждевременное использование этого объекта может привести к недоразумению. Тем не менее мы начнем изучение с введения более общей структуры, называемой решеткой.

Некоторые из решеток имеют важное значение в абстрактной теории вычислений, возникшей из понятия аппроксимации. Одна программа аппроксимирует другую, если она осуществляет те же самые вычисления на подмножестве данных, доступных той программе. Если же рассматривать результат работы программы как множество, то одна программа аппроксимирует другую, когда для некоторых входных данных она выдает в качестве результата подмножество $B \subseteq A$, где $A$ — множество выходных данных второй программы. Это достаточно простые и, вероятно, очевидные способы аппроксимации программ. При этом возникает множество элементов, связанных некоторым отношением порядка.

Вычисление является результатом выполнения программы (на некоторых данных), которая написана на каком-либо языке (см. гл. 8). Для общности рассмотрений свойства вычислений надо выводить не в терминах рассматриваемой программы, а в терминах языка, используемого для записи программы. В то же время мы должны быть в состоянии определить результат выполнения программы «язык $X$» с помощью формального определения. К сожалению, количество деталей, требующихся для описания даже простых примеров, достаточно велико и, следовательно, ничего не определяет. Однако в п. 5.1 мы дадим теоретические обоснования некоторых ключевых результатов. Это будет следовать (п. 5.2) из формального использования булевой алгебры, а затем (п. 5.3) из краткого рассмотрения некоторых общих приложений.

**5.1. Решетки.** Напомним, что бинарное отношение $\rho$ на множестве $S$ является частично упорядоченным отношением, если оно рефлексивно, транзитивно и антисимметрично. Следовательно, $(S, \rho)$ — частично упорядоченное множество, и, если не возникает двусмысленности, $\rho$ можно записать как $\leqslant$, а $(S, \leqslant)$ обозначить просто через $S$. Частично упорядоченное множество называется *линейно упорядоченным* (или *цепью*), если для

любых $x, y \in S$ или $x \leqslant y$, или $y \leqslant x$, или же выполнены оба эти отношения. На самом деле любой частичный порядок можно представить в виде объединения линейных порядков. Поэтому возникает естественный и полезный способ изображения отношений порядка.

Заметим, что любое конечное, линейно упорядоченное множество $(A, \leqslant)$ можно представить следующим образом:

$$a_1 \leqslant a_2 \leqslant \ldots \leqslant a_n;$$

здесь рефлексивность и транзитивность не требуют доказательств. Легко видеть, что рис. 5.7 является «очевидным» представлением $(A, \leqslant)$. Другими словами, мы записываем $A$ как $(a_1, a_2, \ldots, a_n)$.

Предложение. *Частичное упорядочение на конечном множестве может быть представлено как объединение линейных порядков на некоторых подмножествах.*

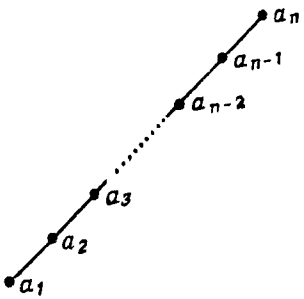



Рис. 5.7

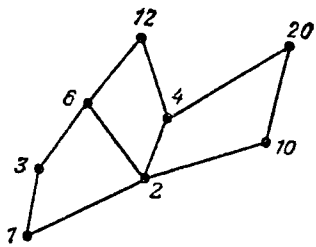



Рис. 5.8

Доказательство этого факта оставляем в качестве упражнения. //

Используя этот результат, можно представить любой частичный порядок (конечный или бесконечный, однако бесконечные отношения сложно изобразить на конечных листах бумаги за конечное время!) изображением множества соответствующей цепей. Полученная диаграмма называется *диаграммой Хасса.*

Пример 5.1. Отношение $\rho = \{(x, y)$: $x$ — множитель $y\}$, определенное на множестве {1, 2, 3, 4, 6, 10, 12, 20}, дает диаграмму Хасса (изображенную на рис. 5.8) и может быть разбито на линейно упорядоченные подмножества {(1, 2, 4, 12), (1, 3, 6, 12), (2, 6), (4, 20), (2, 10, 20)}. Этому разбиению эквивалентно следующее

множество линейных порядков: {(2, 6, 12), (1, 3, 6), (1, 2, 10, 20), (2, 4, 20), (4, 12)}. Заметим, что по сравнению с отношениями, изображенными на диаграммах в § 2 гл. 2, хотя $1px$ для всех $x \in \{1, 2, 3, 4, 6, 10, 12, 20\}$. на этом рисунке отсутствуют восемь стрел, выходящих из 1. Это достигается *неявным* представлением свойств рефлексивности и транзитивности. //

Пусть дано $(A, \leqslant)$ и $B \subseteq A$. Разумно задать вопрос: будет ли $B$ ограничено сверху (снизу) элементами множества $A$? Далее мы можем искать наименьшую верхнюю грань (наибольшую нижнюю грань), которая обозначается sup (inf) (читается супремум (инфимум)). Эти понятия были полностью определены в § 5 гл. 2. Будем их использовать для характеристики решетки.

Определение. *Решеткой* называется частично упорядоченное множество $(A, \leqslant)$, в котором каждая пара элементов имеет супремум и инфимум. Для заданных $x, y \in A$ эти грани будем записывать следующим образом:

$$x \wedge y = \inf(\{x, y\}), \quad x \vee y = \sup(\{x, y\}). \text{ //}$$

Не всякое частично упорядоченное множество является решеткой. Например, частично упорядоченное множество из примера 5.1 не является решеткой, поскольку $12 \vee 20$ не определено.

Определив операции $\wedge$ и $\vee$ между парами элементов в частично упорядоченном множестве, расширим это понятие естественным образом. Положим

$$\bigwedge X = \bigwedge_{x \in X} x = \inf X, \quad \bigvee X = \bigvee_{x \in X} x = \sup X,$$

Это обозначает $\sup X$ и $\inf X$ конечного непустого множества $X$.

Легко показать, что существует много специальных видов решеток, в которых можно производить различные операции. Ограничимся рассмотрением трех таких типов.

Определение. Решетка $L$, обозначаемая $(L, \wedge, \vee)$, *дистрибутивна*, если она подчиняется дистрибутивным законам

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z),$$
$$x \vee (y \wedge z) = (x \vee y) \wedge (x \wedge z)$$

для всех $x, y, z \in L$. //

Не все решетки являются дистрибутивными.

Пример 5.2. Решетка, изображенная на рис. 5.9, не является дистрибутивной, поскольку

$$b \wedge (d \vee c) = b \wedge e = b,$$

тогда как

$$(b \wedge d) \vee (b \wedge c) = a \vee a = a. \ //$$

Предложение. *Пусть в дистрибутивной решетке* $(L, \wedge, \vee)$ *выполнены соотношения*

$$x \vee y = x \vee z, \quad x \wedge y = x \wedge z.$$

*Тогда* $y = z$.

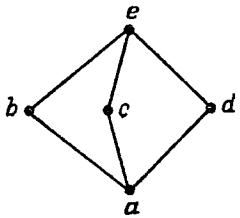


Рис. 5.9

Доказательство. Сначала заметим, что из определения inf и sup

а) $a \wedge b = b \wedge a, \quad a \vee b = b \vee a;$

б) $a \wedge b \leqslant a \leqslant a \vee b;$

в) $(a \wedge b) \vee a = a, \quad a \wedge (a \vee b) = a.$

Поэтому

$$y = y \vee (y \wedge x) = y \vee (z \wedge x) = (y \vee z) \wedge (y \vee x) =$$
$$= (z \vee y) \wedge (z \vee x) = z \vee (y \wedge x) = z \vee (z \wedge x) = z. \ //$$

Определение. Предположим, что $(L, \wedge, \vee)$ — решетка и $0, 1 \in L$ такие, что $0 \leqslant x \leqslant 1$ для всех $x \in L$ (об элементах 0 и 1 скажем немного позже). Тогда

$$x \vee 1 = 1, \quad x \wedge 1 = x, \quad x \wedge 0 = 0, \quad x \vee 0 = x$$

для любого $x \in L$. Такая решетка называется решеткой с дополнениями, если для любого $x \in L$ существует $\bar{x} \in L$ такой, что

$$x \wedge \bar{x} = 0, \quad x \vee \bar{x} = 1$$

($\bar{x}$ называют дополнением $x$). //

Предложение. *Если* $(L, \wedge, \vee)$ — *дистрибутивная решетка с дополнениями, то дополнения единственны.*

Доказательство. Предположим, что $x, y, z \in L$ и

$$x \vee y = x \vee z (= 1), \quad x \wedge y = x \wedge z (= 0).$$

Тогда из предыдущего предложения следует, что $y = z$. //

Третий, и последний специальный тип решеток, который мы определим, необычен в том смысле, что он не дает нам ничего нового для конечных решеток. Чтобы подчеркнуть ключевой момент исследования, дадим другое определение решетки в форме предложения.

Предложение. *$L$ является решеткой тогда и только тогда, когда $\vee X$ и $\wedge X$ существуют для любого непустого конечного подмножества $X$ из $L$.*

(Этот факт можно доказать индукцией по числу элементов множества $X$; оставляем его в качестве упражнения.) //

Если $L$ — решетка и в ней определен элемент $\wedge L$, то он обозначается символом 0 и называется наименьшим элементом $L$. Аналогично, если в $L$ существует элемент $\vee L$, то он обозначается символом 1 и называется наибольшим элементом $L$; по определению $\vee \varnothing = 0$.

Определение. Решетка $L$ называется *полной*, если $\vee X$ и $\wedge X$ существуют для *всех* подмножеств $X$ из $L$. //

Все конечные решетки являются полными. Рассмотрим, однако, множество $\mathbf{Q}$ с обычным отношением порядка $\leqslant$ и бесконечное множество аппроксимаций числа $\pi$, каждое из которых имеет на один десятичный знак больше. Верхняя грань этой последовательности, очевидно, есть $\pi$, однако $\pi \in \mathbf{R} \setminus \mathbf{Q}$, и, следовательно, $(\mathbf{Q}, \leqslant)$ не является полной решеткой. Решетка $(\mathbf{R}, \leqslant)$ является полной, и $\mathbf{Q} \subseteq \mathbf{R}$.

Можно показать, что любая решетка может быть расширена до полной решетки, однако мы не будем заниматься этим вопросом.

**5.2. Булевы алгебры.** Мы уже интенсивно занимались алгеброй множеств и упоминали об относительно «логичной» арифметике. Сейчас дадим формальное определение общей булевой алгебры, названной так в честь математика XIX в. Дж. Буля. Алгебра множеств — это частный случай булевой алгебры, и, хотя различные булевы алгебры структурно подобны, следует заметить, что не все они включают в себя множества обычным образом.

Определение. *Булевой алгеброй* называют множество $\mathscr{B}$ вместе с тремя операциями $\vee$, $\wedge$ и $^-$ ($\vee$ называют операцией *или*, $\wedge$ — операцией *и*, а $^-$ — операцией *дополнения* или же операцией *не*; кроме того, первые две операции часто называют дизъюнкцией и конъюнкцией соответственно). Бинарные операторы $\vee$ и $\wedge$ и унарный оператор $^-$ (обычно записывается над операндом, например $\bar{a}$) вместе с двумя различными элементами $\mathscr{B}$, которые обозначаются символами 0 и 1, удовлетворяют следующим аксиомам.

Для произвольных элементов $a$, $b$ и $c$ в $\mathscr{B}$:

а) $a \vee b = b \vee a$; б) $a \vee (b \vee c) = (a \vee b) \vee c$;

в) $a \vee 0 = a$; г) $a \vee \bar{a} = 1$;

д) $a \wedge b = b \wedge a$; е) $a \wedge (b \wedge c) = (a \wedge b) \wedge c$;

ж) $a \wedge 1 = a$; з) $a \wedge \bar{a} = 0$;

и) $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$;

к) $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$.

Таким образом, *и* и *или* коммутативны, ассоциативны и дистрибутивны одна по отношению к другой. Каждая из этих операций имеет единичный элемент, и, когда элемент комбинируется вместе со своим дополнением посредством *и/или*, в результате получается единица по отношению к *или/и* соответственно.

Названия операторов, использованные выше, являются именами, которые непосредственно связаны с компьютерной логикой, используемой при построении схем. В других булевых алгебрах может быть более подходящим читать $\vee$ как объединение, наименьшая верхняя грань или supremum, а $\wedge$ как пересечение, наибольшая нижняя грань или infimum. (Иногда операцию $\wedge$ обозначают также &.) Операция $\bar{a}$ может записываться как $a'$ или $\neg a$. //

Булеву алгебру можно определить как дистрибутивную решетку с дополнением. Поэтому по аналогии с булевой алгеброй $(\mathscr{P}(X), \cup, \cap, ')$ для данного непустого множества $X$ из результатов предыдущего параграфа можно вывести ряд следствий. Доказательства некоторых из них оставлены в качестве упражнений. Наиболее важными из этих следствий являются следующие.

Инволютивный закон (или закон двойного отрицания). Дополнение дополнения $x$, $x \in \mathscr{B}$ (т. е. в рассматриваемой булевой алгебре), есть $x$.

Закон поглощения. Для любых $a$, $b \in \mathscr{B}$ справедливы соотношения

$$a \wedge (a \vee b) = a, \quad a \vee (a \wedge b) = a.$$

Закон идемпотентности. Любой элемент $\mathscr{B}$ идемпотентен по отношению к операциям $\wedge$ и $\vee$.

Законы де Моргана. Для любых $a$, $b \in \mathscr{B}$ справедливы соотношения

$$\overline{a \wedge b} = \bar{a} \vee \bar{b}, \quad \overline{a \vee b} = \bar{a} \wedge \bar{b}.$$

Из инволютивного закона и законов Моргана следует, что если мы берем булеву алгебру $\mathscr{B}$ (более точно,

$(\mathscr{B}, \vee, \wedge, \bar{\ })$) и образуем новую систему или путем отображения каждого $x \in \mathscr{B}$ в $\bar{x}$, или использованием операций $\wedge$ и $\vee$, то $(\mathscr{B}, \wedge, \vee, \bar{\ })$ также булева алгебра. Этот факт известен как *принцип двойственности*. Действительно, каждый из законов Моргана может быть получен из других путем отображения каждого элемента в его дополнение.

Сейчас мы выведем теорему, которая показывает, в какой стандартной форме могут записываться булевы выражения; непосредственным следствием этого является утверждение, что достаточно двух операторов, чтобы описать все бинарные функции. Как всегда, введем вначале необходимую терминологию.

Определим две общеупотребительные логические связки следующим образом. Говорят, что булевы формулы $A$ и $B$ *эквивалентны* (записывается в виде $A \equiv B$ или $A \leftrightarrow B$), если они выражают одну и ту же функцию. В простейшем случае это — отношение эквивалентности.

Таблица 5.3

| $A$ | $B$ | $A \rightarrow B$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Аналогично говорят, что формула $A$ имплицирует формулу $B$ (записывается *) $A \rightarrow B$), если имеют место условия, изображенные в табл. 5.3. (Таблицы такого типа часто называют *таблицами истинности*.)

Заметим, что стрелки используются во многих различных контекстах. Поэтому надо обращать особое внимание на расшифровку их значения. Таким образом, $A \rightarrow B$ это то же самое, что и $(\neg A) \vee B$, а $A \equiv B$ это то же самое, что и $(A \rightarrow B) \wedge (B \rightarrow A)$. Логически $A \rightarrow B$ означает «если $A$, то $B$» (т. е. если $A$ справедливо, то $B$ также справедливо и обычно так и читается. Другие названия для символов $\rightarrow$ и $\leftrightarrow$ это *условный* и *безусловный* операторы соответственно.

Обычно вводят операции, которые обозначают отри-

*) Это выражение называют также *импликацией*.— *Примеч. ред.*

цания бинарных связок, такие, как

$$A \not\equiv B = \neg(A \equiv B), \quad A \uparrow B = \neg(A \wedge B),$$
$$A \not\to B = \neg(A \to B), \quad A \downarrow B = \neg(A \vee B).$$

Сейчас можно описать все бинарные операции $\{0, 1\}^2 \to \{0, 1\}$ (здесь стрелка находится между двумя множествами и, следовательно, обозначает множества, которые являются областью определения и множеством значений; не следует путать с логическим оператором $\to$) в краткой форме, как показано в табл. 5.4. Более того, мы

Таблица 5.4

| A B | 0 0 | 1 0 | 0 1 | 1 1 | Функция |
|---|---|---|---|---|---|
| $f_0$ | 0 | 0 | 0 | 0 | 0 |
| $f_1$ | 0 | 0 | 0 | 1 | $A \wedge B$ |
| $f_2$ | 0 | 0 | 1 | 0 | $B \not\to A$ |
| $f_3$ | 0 | 0 | 1 | 1 | $B$ |
| $f_4$ | 0 | 1 | 0 | 0 | $A \not\to B$ |
| $f_5$ | 0 | 1 | 0 | 1 | $A$ |
| $f_6$ | 0 | 1 | 1 | 0 | $A \not\equiv B$ |
| $f_7$ | 0 | 1 | 1 | 1 | $A \vee B$ |
| $f_8$ | 1 | 0 | 0 | 0 | $A \downarrow B$ |
| $f_9$ | 1 | 0 | 0 | 1 | $A \equiv B$ |
| $f_{10}$ | 1 | 0 | 1 | 0 | $\neg A$ |
| $f_{11}$ | 1 | 0 | 1 | 1 | $A \to B$ |
| $f_{12}$ | 1 | 1 | 0 | 0 | $\neg B$ |
| $f_{13}$ | 1 | 1 | 0 | 1 | $B \to A$ |
| $f_{14}$ | 1 | 1 | 1 | 0 | $A \uparrow B$ |
| $f_{15}$ | 1 | 1 | 1 | 1 | 1 |

можем описать все отображения из $\{0, 1\}^n$ в $\{0, 1\}$, используя только $\uparrow$ или только $\downarrow$ (первая из них называется штрихом Шеффера и обозначается «|»; в вычислительном контексте $\uparrow$ называют «не и», а $\downarrow$ называют *) «не или»). Заметим, что операции $\uparrow$ и $\downarrow$ являются удобными сокращениями, но они, например, неассоциативны. Следовательно, по определению $A \uparrow B \uparrow C$ есть $(A \wedge B \wedge C)'$, а, например, не $((A \wedge B)' \wedge C)'$. Говорят, что множества операторов $\{\uparrow\}$ и $\{\downarrow\}$ *адекватны*. Сейчас мы сформулируем теорему и дадим ее конструктивное доказательство, т. е. не только докажем справедливость утверждения теоремы, но и дадим метод получения результата.

Теорема. *Любое отображение* $\{0, 1\}^n \to \{0, 1\}$ *может быть представлено в виде формулы, содержащей только оператор* $\uparrow$ *или только оператор* $\downarrow$.

Доказательство. Рассмотрим произвольное отображение $f$: $\{0, 1\}^n \to \{0, 1\}$ от переменных $p_1, \ldots, p_n$ ($n \in \mathbf{N}$). Функция может быть полностью определена

*) Функция $A \downarrow B$ называется также *стрелкой Пирса.— Прим. ред.*

таблицей истинности, имеющей $2^n$ строк *). Утверждение очевидно, если в каждой строке таблицы результат равен 1 (тогда $f = 1$) или в каждой строке результат равен 0 (тогда $f = 0$). В противном случае существует $m$ строк таблицы, в которых результат равен 1.

Рассмотрим выражения **) $B_1, \ldots, B_m$; каждое $B_i$ соответствует упорядоченному набору $(B_{i1}, \ldots, B_{in})$, где $B_{ij}$ равно или $p_j$, или $\neg p_j$ в зависимости от того, равно $p_j$ единице или нулю для данной строки таблицы. Тогда

$$f = B_1 \vee B_2 \vee \ldots \vee B_m = (B_1 \vee B_2 \vee \ldots \vee B_m)'' =$$
$$= (B_1' \wedge B_2' \wedge \ldots \wedge B_m')' = (B_1' \uparrow B_2' \uparrow \ldots \uparrow B_m'),$$
$$B_i' = (B_{i1} \wedge B_{i2} \wedge \ldots \wedge B_{in})' = B_{i1} \uparrow B_{i2} \uparrow \ldots \uparrow B_{in}.$$

Более того, если какое-либо $B_{ij}$ соответствует нулю в таблице, то надо брать выражение $\neg p_j$, которое можно получить при помощи тождества

$$(\neg a) \equiv (a \uparrow a).$$

Итак, утверждение доказано. В компактном виде этот результат можно записать следующим образом:

а)
$$f = \bigvee_{i=1}^{m} \left( \bigwedge_{j=1}^{n} B_{ij} \right), \qquad (*)$$
$$f = \mathop{\uparrow}_{i=1}^{m} \left( \mathop{\uparrow}_{j=1}^{n} B_{ij} \right).$$

б) Аналогично, применяя законы де Моргана к (*) или же рассматривая строки, содержащие нуль, получим

$$f = \mathop{\downarrow}_{i=1}^{2^n - m} \left( \mathop{\downarrow}_{j=1}^{n} \neg B_{ij} \right). \qquad (**)$$

Следствием этого результата является тот факт, что каждое выражение выводимо, а отсюда в свою очередь следует, что число элементов в булевой алгебре, порожденной $p_1, \ldots, p_n$, равно $2^n$. //

---

*) Каждая строка соответствует некоторому двоичному набору длины $n$ и содержит значения функции на этом наборе.— *Примеч. ред.*

**) Пусть $(\alpha_{11}, \ldots, \alpha_{1n}), \ldots, (\alpha_{m1}, \ldots, \alpha_{mn})$ — все наборы, на которых функция $f$ равна 1, $1 \leqslant m < 2^n$. Для каждого такого набора образуем конъюнкцию $B_i = B_{i1} \wedge \ldots \wedge B_{in}$, где $B_{ik} = p_k$, если $\alpha_{ik} = 1$, и $B_{ij} = \neg p_k$, если $\alpha_{ik} = 0$ $(k = 1, \ldots, n)$. Легко видеть, что $f = B_1 \vee \ldots \vee B_m$.— *Примеч. ред.*

Пример 5.3. Получим представление для функции $f$, используя только операцию $\uparrow$; таблица истинности функции $f$ дана в виде табл. 5.5.

Таблица 5.5

| $x$ | $y$ | $z$ | $f$ | $x$ | $y$ | $z$ | $f$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

Используя описанный метод, получаем

$$f = (x' \wedge y' \wedge z) \vee (x' \wedge y \wedge z') \vee (x \wedge y' \wedge z') \vee \vee (x \wedge y' \wedge z) \vee (x \wedge y \wedge z).$$

Таким образом,

$$f' = (x' \wedge y' \wedge z)' \wedge (x' \wedge y \wedge z')' \wedge \wedge (x \wedge y' \wedge z')' \wedge (x \wedge y' \wedge z)' \wedge (x \wedge y \wedge z)',$$

и поэтому

$$f = (x' \uparrow y' \uparrow z) \uparrow (x' \uparrow y \uparrow z') \uparrow (x \uparrow y' \uparrow z') \uparrow \uparrow (x \uparrow y' \uparrow z) \uparrow (x \uparrow y \uparrow z),$$

где $x' = x \uparrow x$, $y' = y \uparrow y$, $z' = z \uparrow z$. //

Выражение (*) называют *дизъюнктивной нормальной формой* (ДНФ) — она имеет вид дизъюнкции конъюнкций (*или* от *и*); выражение, полученное в (**)*), называется *конъюнктивной нормальной формой* (КНФ).

В заключение отметим, что утверждение в булевой алгебре (логическая формула), которое всегда принимает значение 1, называется *тавтологией*, а выражение, которое всегда принимает значение 0, называется *противоречием*.

**5.3. Некоторые приложения булевой алгебры.** Существует ряд проблем, например в комбинаторике, которые можно решить наилучшим образом в подходящей булевой алгебре. Другой путь применения булевой алгебры заключается в моделировании реальной ситуации, или,

*) На самом деле конъюнктивной нормальной формой называют выражение вида $f = \bigwedge_{i=1}^{2^n - m} \left( \bigvee_{j=1}^{n} \neg B_{ij} \right)$ (см. **) из доказательства теоремы.— *Примеч. ред.*

другими словами, в *интерпретации* булевой алгебры в терминах, относящихся к рассматриваемой задаче.

Этот метод может быть применен в областях, которые так широко распространены, что у них отсутствует классификация. Мы рассмотрим только одно приложение, а именно комбинационные и переключательные схемы *). Другие области приложений будут представлены в виде отдельно выбранных примеров и упражнений.

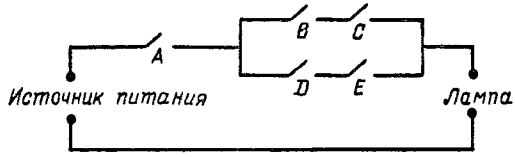



Рис. 5.10

С этой точки зрения необходимо подчеркнуть, что мы занимаемся только некоторыми вопросами построения и преобразования схем; формальное исследование лежит за пределами этой книги. Сначала давайте посмотрим, как выражения булевой алгебры могут быть использованы для описания и определения схемы, состоящей из проводников и переключателей. Схема, изображенная на рис. 5.10, состоит из пяти переключателей, источника питания, лампы и соединяющих проводников. Из диаграммы нетрудно видеть, что ток течет по замкнутой цепи тогда и только тогда, когда переключатель $A$ замкнут и (или)

переключатели $B$ и $C$ оба замкнуты

или

переключатели $D$ и $E$ оба замкнуты.

Алгебраически это можно записать в виде

$$A \wedge ((B \wedge C) \vee (D \wedge E)).$$

Перед тем как продолжить изложение, сделаем несколько замечаний о представлении элементов электрических схем с переключателями в виде диаграмм. Единственный сегмент схемы, в котором будет отличие между рассматриваемыми примерами,— это верхняя часть схемы, содержащая переключатели; следовательно, можно

*) Эти схемы называются также схемами из функциональных элементов и контактными схемами, соответственно.— *Примеч. ред.*

не изображать остальной части схемы. Внутри схемы некоторые переключатели могут быть взаимосвязаны (рис. 5.11, *а*) и, возможно, могут работать так, что когда один переключатель включен, то другой должен быть выключен и наоборот (рис. 5.11, *в*). Основные методы

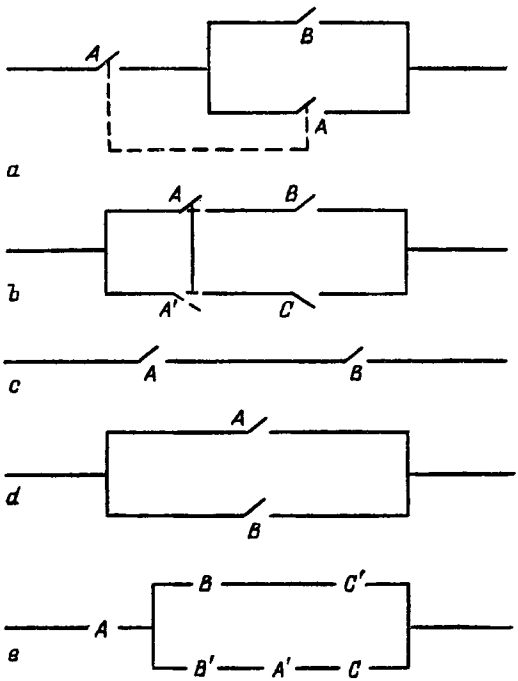



Рис. 5.11

расположения переключателей — это последовательный (рис. 5.11, *с*), что соответствует связке *и* (т. е. $A \wedge B$), и параллельный (рис. 5.11, *d*), что соответствует связке *или* (т. е. $A \vee B$). Отметим, наконец, что на диаграмме есть совершенно лишние переключатели, и их для удобства можно обозначать теми же «именами» (рис. 5.11, *е*). Таким образом, любую операцию в булевой алгебре можно представить при помощи схемы и наоборот. Следовательно, для получения эквивалентных схем надо перейти от первоначальной схемы к соответствующему ей выражению в булевой алгебре, затем к «более простому» (или более желаемому) выражению и, наконец, к схеме, соответствующей этому выражению.

Пример 5.4. Упростить схему, изображенную на рис. 5.12. Этой схеме соответствует следующее выражение:

$$X \wedge ((((X' \wedge Y) \vee (Y' \wedge Z)) \wedge W') \vee (Z \wedge X \wedge W)) =$$
$$= X \wedge ((X' \wedge Y \wedge W') \vee (Y' \wedge Z \wedge W') \vee (Z \wedge X \wedge W)) =$$
$$= (X \wedge X' \wedge Y \wedge W') \vee (X \wedge Y' \wedge Z \wedge W') \vee (X \wedge Z \wedge X \wedge W) =$$
$$= (X \wedge Y' \wedge Z \wedge W') \vee (X \wedge Z \wedge W) = (X \wedge Z) \wedge ((Y' \wedge W') \vee W).$$

Последнее выражение соответствует схеме, изображенной на рис. 5.13.

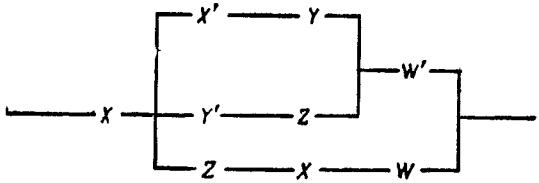



Рис. 5.12

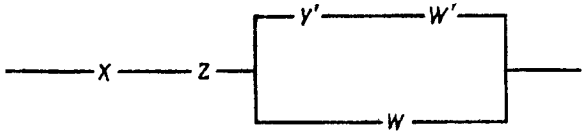



Рис. 5.13

Мы уже уделили достаточно внимания схемам с переключателями. Перейдем теперь к комбинационным схемам. Основные компоненты таких схем — элементы. В наиболее общей форме *элемент* является устройством, имеющим $n$ входов и $m$ выходов ($m, n \in \mathbf{N}$). На каждый вход может подаваться один из двух бинарных сигналов; представим их как 0 и 1, однако любая другая система из двух различных значений также нам подходит. В реальной ситуации обычно мы имеем сигналы в окрестности 0 вольт (например, от 0 до 0,8 вольта), которые представляются нулем, и в другой окрестности — в районе 4 вольт (скажем, между 3 и 5 вольтами), которые представляются единицей. Отсюда определяются выходные значения (0 и 1). Можно ожидать, что нам понадобится достаточно много различных элементов, чтобы представить все возможные функции $\{0, 1\}^n \to \{0, 1\}$, однако, как мы увидим далее, это не так. Схемам, не имеющим временны́х задержек (в дальнейшем только такие схемы мы и будем рассматривать), соответствует един-

ственный тип элементов. Однако это не очевидно. Структура схем станет более ясной, если вначале мы рассмотрим три типа элементов, изображенных на рис. 5.14. Самым

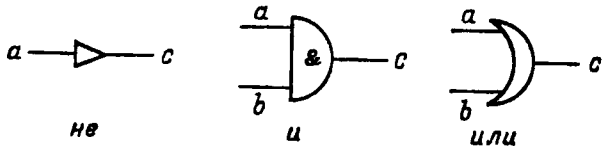



Рис. 5.14

простым является элемент **не**; его действие определяется тождеством

$$c = \neg a \,(c = \text{не } a),$$

определяемым табл. 5.6.

Таблица 5.6

| INPUT($a$) | OUTPUT($c$) |
|---|---|
| 0 | 1 |
| 1 | 0 |

Аналогично элементы **и** и **или** определяются табл. 5.7.

Таблица 5.7

| INPUTS | | OUTPUTS ($c$) | |
|---|---|---|---|
| $a$ | $b$ | **и** | **или** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Элементы **и** и **или** могут также иметь несколько входов; в этом случае они определяются аналогично, т. е. выходной сигнал элемента **и** равен 1 тогда и только тогда, когда на *всех* входах сигнал равен 1, и выходной сигнал элемента **или** равен 1 тогда и только тогда, когда *какой-либо* из входных сигналов равен 1. Алгебраически выходной сигнал элемента **не** имеет вид: $\neg$ (входной сигнал), выходной сигнал элемента **и** имеет вид: (сигнал первого входа) $\wedge$ (сигнал второго входа) $\wedge \ldots$ и, наконец, выходной сигнал элемента **или** имеет вид: (сигнал первого входа) $\vee$ (сигнал второго входа) $\vee \ldots$ Что-

бы представить более сложные функции, элементы можно «соединять» различными способами.

Пример 5.5. Рассмотрим схему (более точно, часть схемы, которая нас интересует), изображенную на

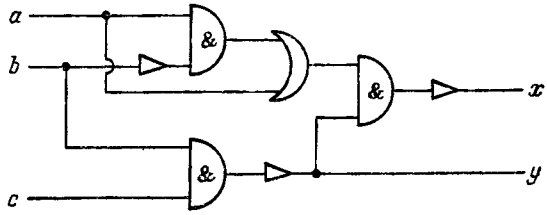



Рис. 5.15

рис. 5.15. Возвращаясь «назад» от выходов схемы, получаем

$$x = ((a \vee (a \wedge b')) \wedge (y))',$$
$$y = (b \wedge c)'.$$

Поэтому

$$x = (((a \vee a) \wedge (a \vee b')) \wedge (b \wedge c)')' =$$
$$= ((a \wedge (a \vee b')) \wedge (b \wedge c)')' = (a \wedge (b \wedge c)')'$$

(по закону поглощения).

Таким образом, одно из возможных упрощений схемы изображено на рис. 5.16. //

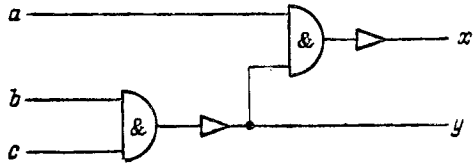



Рис. 5.16

Следовательно, мы можем использовать булевы выражения для анализа и упрощения сложных схем. Рассмотренный пример показывает, как можно осуществлять такие операции.

Существуют также устройства, которые реализуют другие логические операции. Наиболее важными являются элементы не и и не или; они изображены на рис. 5.17. Используя множество таких элементов, при помощи булевых выражений можно получить требуемую схему непосредственно по таблице истинности.

Пример 5.5 (продолжение). Не останавливаясь на формальностях, заметим, что каждое выражение в скобках в окончательной форме $f$ (см. выше) представляет вход для последнего элемента на рис. 5.18 и может быть вычислено с помощью элементов предшествующего ряда


Рис. 5.17

(Части схемы, обведенные штриховыми линиями, практически не требуются, поскольку для многих электрических элементов дополнение к выходу также доступно из вспомогательного выхода.) Отметим, что полученная схема в общем случае не будет минимальной в смысле числа используемых элементов. //

В завершение рассмотрим два примера, которые показывают, как булева алгебра может быть использована в задачах иного рода.

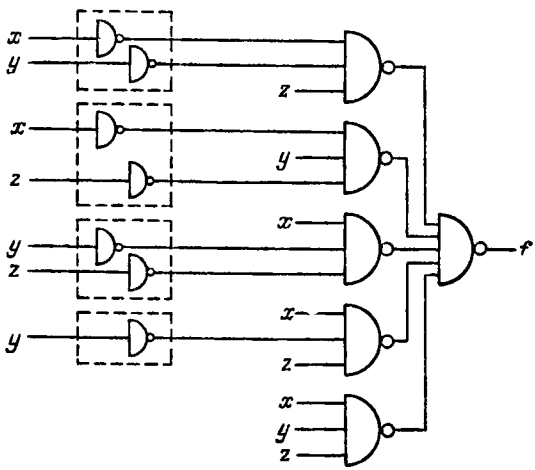



Рис. 5.18

Пример 5.6. Предположим, что справедливы следующие утверждения:

а) знание структур данных необходимо для совершенствования дисциплины ума;

б) только опыт программирования может создать дисциплинированный ум;

в) для того чтобы написать компилятор, надо иметь возможность анализировать задачи;

г) недисциплинированный ум не может анализировать задачи;

д) всякий, кто писал структурные программы, может рассматриваться как опытный программист.

Можно ли из этих предположений определить справедливость нижеследующих утверждений:

а′) опыт написания структурных программ необходим для того, чтобы быть в состоянии написать компилятор;

б′) знание структур данных является частью опыта программирования;

в′) анализ задач невозможен теми, кто игнорирует структуры данных;

г′) опытный программист, который писал структурные программы, в состоянии анализировать задачи и имеет дисциплинированный ум, является программистом, который мог бы написать компилятор?

Чтобы ответить на эти вопросы, мы могли бы исследовать логические следствия утверждений, однако для того, чтобы проиллюстрировать используемую технику в более сложных ситуациях, воспользуемся нашим знанием булевой алгебры. Сначала необходимо закодировать наши утверждения. Пусть

$\mathscr{E}$ — множество всех программистов,

$U$ — те из них, кто знает структуры данных,

$V$ — те из них, кто имеет дисциплинированный ум,

$W$ — те из них, кто является опытными программистами,

$X$ — те из них, кто мог бы написать компилятор,

$Y$ — те из них, кто может анализировать задачи,

$Z$ — те из них, кто может писать структурные программы. Тогда имеем

а) $U \supseteq V$; б) $W \supseteq V$; в) $X \subseteq Y$;

г) $V \supseteq Y$; д) $W \subseteq Z$;

а′) $Z \supseteq X$; б′) $U \supseteq W$; в′) $Y \subseteq U$;

г′) $W \cap Z \cap Y \cap V \supseteq X$.

Теперь видно, что а′) непосредственно следует из в), г), б) и д) до транзитивности; аналогично в′) следует из г) и а). Также

$$W \cap Z \cap Y \cap V = W \cap Y \cap V = Y \cap V = Y \supseteq X$$

Таким образом, г′) также справедливо. Утверждение б′) не может быть выведено из а)—д), поскольку мы только знаем, что $W \subseteq Z$, и цепочка на этом заканчивается. //

Пример 5.7. Алиса сказала, что Барбара и Клара говорят правду, а Клара сказала, что Элспет и Фиона или обе говорят правду, или обе лгут. С другой стороны, Деби считает, что по крайней мере или Алиса, или Барбара говорят правду, тогда как Барбара утверждает, что только одна из двух — Алиса или Фиона — была правдивой. Элспет считает, что Алиса и Барбара всегда говорят правду, однако Фиона уверена, что Барбара и Клара обе не могли сказать правду. Кому мы должны верить?

Рассмотрим множество утверждений вида
«Алиса (не) говорит правду»,
«Барбара (не) говорит правду»,
. . . . . .

и пусть $A$ содержит только те утверждения, в которых Алиса говорит правду. Пусть $B$, $C$, $D$, $E$ и $F$ определены аналогично. Тогда, если $S$ — непустое множество пересечений таких, что

$$S \subseteq A \cap B' \cap C,$$

то $S$ включает утверждения, в которых

«Алиса говорит правду»,
«Барбара лжет»,
«Клара говорит правду».

Из первого утверждения следует, что можно вывести следующий факт: если Алиса говорит правду, то то же самое делают Барбара и Клара. Это можно закодировать следующим образом:

$$\text{если } x \in A, \text{ то } x \in B \cap C;$$

следовательно,

$$A \subseteq B \cap C. \tag{1}$$

Аналогично другие утверждения могут быть записаны в виде

$$C \subseteq (E \cap F) \cup (E' \cap F'), \tag{2}$$

$$D \subseteq A \cup B, \tag{3}$$

$$B \subseteq (E \cup F) \setminus (E \cap F) = (E \cap F') \cup (E' \cap F), \tag{4}$$

$$E \subseteq A \cap B, \tag{5}$$

$$F \subseteq (B \cap C)'. \tag{6}$$

Из (2), (4) и (1) следует, что $B \cap C = \varnothing$, и поэтому $A = \varnothing$; тогда из (5) следует, что $E = \varnothing$. Эти выводы мы смогли получить непосредственно из условий задачи. До сих пор наши рассуждения были эффективны потому, что мы пользовались тем, что правдивый человек говорит правду. Однако пока мы не использовали ничего о справедливости того, что говорит лжец. Если мы ограничимся фактором, что все, что говорит лжец, неверно, то можем заменить множество включений из (1)—(6) на эквивалентные. Например, если Барбара и Клара правдивы, то, поскольку Алиса говорит, что это так, Алиса тоже правдива. Следовательно,

$$A = B \cap C, \tag{1a}$$

$$C = (E \cap F) \cup (E' \cap F'), \tag{2a}$$

$$D = A \cup B, \tag{3a}$$

$$B = (E \cap F') \cup (E' \cap F), \tag{4a}$$

$$E = A \cap B, \tag{5a}$$

$$F = (B \cap C)'. \tag{6a}$$

Как и раньше, $A = \varnothing$ и $E = \varnothing$, и поэтому Алиса и Элспет никогда не говорят правду. Из (2a), (4a) и (6a) следует, что $F = \mathscr{E}$. Таким образом, из (3a) и (4a) получаем $D = B = F = \mathscr{E}$, откуда с помощью подстановки в (2a) следует $C = \varnothing$.

Отсюда следует, что возможно только одно утверждение такое, что

$$x \in A' \cap B \cap C' \cap D \cap E' \cap F;$$

это означает, что Алиса, Клара и Элспет лгут, а Барбара, Деби и Фиона говорят правду. //

Упражнение 5.5.

1. Показать, что в булевой алгебре $(\mathscr{B}, *, +, ')$ для $x, y, z \in \mathscr{B}$ справедливы соотношения:

а) $(x + y) * (x' + y) = y$;

б) $(z + x) * (z' + y) = (z * y) + (z' * x)$;

в) $((x * z) + (y * z'))' = (x' * z) + (y' * z')$.

2. Доказать, что если в $(\mathscr{B}, *, +, ')$ выполняются соотношения $x * y = x * z$ и $x + y = x + z$, то $y = z$.

3. Из схем, изображенных на рис. 5.19 и 5.20, получить «простые» схемы, эквивалентные исходным.

4. Получить алгебраическое представление схемы, изображенной на рис. 5.21, и дать табличное представление выходов $x$ и $y$ в зависимости от значений $a$ и $b$ на входе. Использовать две такие схемы для построения схемы,

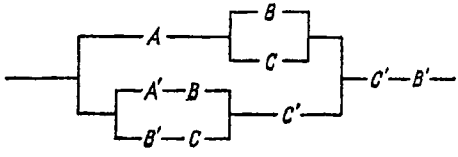



Рис. 5.19

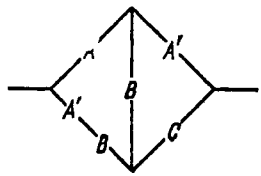



Рис. 5.20

соответствующей функции, определяемой табл. 5.8. Связать это с арифметикой из § 3 гл. 4.

5. Построить схему, реализующую операцию не и, которая имеет четыре входа и выдает на выходе 1 тогда и

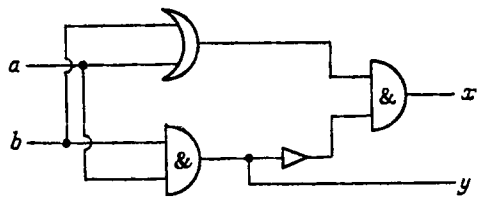



Рис. 5.21

только тогда, когда все входы равны 0.

6. Пусть выполнены указанные здесь условия:

а) все гонщики импульсивны;

б) все хорошие программисты меланхоличны;

в) никто не может быть и импульсивным и меланхоличным;

г) читатель меланхоличен.

Таблица 5.8

| Входы | | | Выходы | | Входы | | | Выходы | |
|---|---|---|---|---|---|---|---|---|---|
| $P$ | $Q$ | $R$ | $S$ | $T$ | $P$ | $Q$ | $R$ | $S$ | $T$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

Является ли читатель хорошим программистом и/или гонщиком?

7. Организаторы международной конференции по компьютерам решили, что для того, чтобы на встрече не доминировали коммерческие интересы, будет только один магазин, которым будут пользоваться вместе все производители компьютеров. Сами производители будут ответственны за определение того, кто принимает участие в представительстве. Десять компаний — пять европейских и пять американских — дали знать, что они хотят принять участие в конференции. (Обозначим эти компании через $A$, $B$, $C$, $D$, $E$ и $F$, $G$, $H$, $I$, $J$ соответственно.) Однако из-за обязательств по контрактам и торговой политики должны появиться различные ограничения. Европейские предписания требуют, чтобы $G$ и $I$ не могли одновременно принимать участие в конференции; аналогично не могут одновременно принять участие $F$, $G$ и $J$. Ограничения американских производителей исключают участие $A$ и $D$, пока $G$ не примет участия; аналогично исключаются $C$ и $D$. Если $E$ присутствует на конференции, то должно присутствовать и $J$, однако если они принимают участие вдвоем, то $B$ не может быть там. И наконец, $B$ и $C$ не могут вместе принимать участие в конференции.

Может ли быть достигнуто соглашение, по которому по три компании из каждой группы могут собраться вместе, не нарушая условий? Если так, то кто именно примет участие?

## § 6. Замкнутые полукольца

В завершение главы мы опишем довольно узкие, но важные структуры, которые позволяют осуществлять операции на подмножествах бесконечных множеств. Пока у нас нет понятий матриц и графов, мы не можем начинать обсуждение приложений, и, следовательно, мы дадим только аксиоматическое определение и покажем принципиальную разницу между полукольцами и похожими на них на первый взгляд полями.

Определение. *Замкнутым полукольцом* называется множество $S$ с двумя бинарными операциями $\circledast$ и $\oplus$ такими, что

а) $\oplus$ ассоциативна;

б) существует единичный элемент по отношению к $\oplus$, который будем обозначать символом 0;

в) $\circledast$ ассоциативна;

г) существует единичный элемент по отношению к $\circledast$, который будем обозначать символом 1;

д) для всех $x \in S$

$$x \circledast 0 = 0 = 0 \circledast x;$$

е) $\oplus$ коммутативна:

$$x \oplus y = y \oplus x \quad \text{для всех} \quad x, y \in S;$$

ж) $\oplus$ идемпотентна:

$$x \oplus x = x \quad \text{для всех} \quad x \in S;$$

з) $\circledast$ дистрибутивна относительно $\oplus$:

$$x \circledast (y \oplus z) = (x \circledast y) \oplus (x \circledast z) \quad \text{для всех} \quad x, y, z \in S,$$

$$(x \oplus y) * z = (x \circledast z) \oplus (y \circledast z) \quad \text{для всех} \quad x, y, z \in S;$$

и) сумма счетного числа элементов из $S$ существует и единственна, т. е. не зависит от порядка суммирования;

к) $\circledast$ дистрибутивна относительно бесконечных счетных сумм, т. е.

$$\left(\sum_i a_i\right) \circledast \left(\sum_j b_j\right) = \sum_{i,j} (a_i \circledast b_j),$$

где

$$\sum_i a_i = a_1 \oplus a_2 \oplus \ldots, \quad \sum_j b_j = b_1 \oplus b_2 \oplus \ldots,$$

$$\sum_{i,j} (a_i \circledast b_j) = (a_1 \circledast b_1) \oplus (a_1 \circledast b_2) \oplus \ldots$$

$$\ldots \oplus (a_2 \circledast b_1) \oplus (a_2 \circledast b_2) \oplus \ldots \oplus (a_3 \circledast b_1) \oplus$$

$$\oplus (a_3 \circledast b_2) \oplus \ldots \oplus \ldots \quad //$$

Эта достаточно странная структура часто используется в алгоритмах замыкания для графов (гл. 7 и 8). Чтобы проиллюстрировать результаты, которые могут быть получены в замкнутых полукольцах, сравним системы $(Z_2, *, +)$ и $(Z_2, \wedge, \vee)$, где операции определены табл. 5.9.

Таблица 5.9

| * | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| $\wedge$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

| $\vee$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

Рассмотрим определение операции замыкания. Результат $a^*$ применения этой операции к элементу $a$ записываем в виде

$$a^* = \sum_{i=0}^{\infty} a^i,$$

$$a^0 = 1, \quad a^1 = a, \quad a^2 = a * a \quad \text{и} \quad a^i = a^{i-1} * a$$

(в $(\mathbf{Z}_2, \wedge, \vee)$ сложением является $\vee$, а умножением $\wedge$). В замкнутом полукольце $(\mathbf{Z}_2, \wedge, \vee)$ такое определение имеет смысл и допустимо. В частности,

$$1^* = 1^0 \vee 1^1 \vee 1^2 \vee \ldots = 1 \vee 1 \vee 1 \vee \ldots = 1$$

(в силу аксиом ж) и и)). Однако если мы попытаемся те же самые вычисления произвести в поле $(\mathbf{Z}_2, *, +)$, то получим

$$1^* = 1^0 + 1^1 + 1^2 + \ldots = 1 + 1 + 1 + \ldots$$

Эту сумму можно вычислить следующим способом:

$$\underbrace{1+1}+\underbrace{1+1}+\underbrace{1+1}+\ldots = 0+0+0+\ldots = 0$$

или же

$$1+\underbrace{1+1}+\underbrace{1+1}+\underbrace{1+1}+\ldots = 1+0+0+0+\ldots = 1,$$

т. е. значение суммы не определено. Следовательно, понятие такого рода замыкания не имеет смысла, если его применять к полю $\mathbf{Z}_2$ (которое запрещает аксиомы ж) и и) полукольца).

Упражнение 5.6. Проверить, что $(\{\varnothing, \mathscr{E}\}, \cap, \cup)$ является замкнутым полукольцом.

ГЛАВА 6

# МАТРИЦЫ

Наше изложение теории *конечных* матриц (множества матриц) является более общим, чем принятое в большинстве книг, в которых матрицы обычно определяют как линейные преобразования на векторных пространствах, используя знакомство с координатной геометрией. Хотя иногда мы будем возвращаться (§ 3) к этой интерпретации, в основном будем придерживаться точки зрения, что матрицы являются реализациями абстрактных алгебраических структур в вычислительных целях. Тогда алгебра абстрактных структур определяет способы, как надо комбинировать матрицы.

Сначала мы определим (§ 1) матричное представление бинарных отношений над конечными множествами. Затем последует более общее рассмотрение требуемых свойств абстрактной системы для того, чтобы матричная реализация была разумной. А в завершение, используя результаты предыдущих исследований, будет рассмотрен важный случай векторного пространства над R.

## § 1. Матрицы и бинарные отношения на конечных множествах

Формально *матрицей* над множеством $S$ называется отображение

$$M\colon \mathbf{N}_p \times \mathbf{N}_q \to S, \quad p, q \in \mathbf{N}.$$

Обычно образ $(i, j)$ обозначают через $M_{ij}$ и изображают всю функцию массивом элементов из $S$, т. е.

$$M = \begin{bmatrix} M_{11} & M_{12} \dots M_{1q} \\ M_{21} & M_{22} \dots M_{2q} \\ \dots\dots\dots\dots \\ M_{p1} & M_{p2} \dots M_{pq} \end{bmatrix}.$$

Говорят, что эта матрица имеет $p$ *строк* и $q$ *столбцов* и имеет *размер* $p \times q$. Матрица размера $p \times q$ имеет $p * q$

элементов. Когда $p = q$, матрицу называют *квадратной*. Множество всех матриц $p \times q$ над $S$ обозначают через $\mathcal{M}(p, q, S)$. Множество $\mathcal{M}(p, p, S)$ будем обозначать через $\mathcal{M}(p, S)$.

Рассмотрим бинарное отношение $\rho$ между множествами $A$ и $B$, где

$$A = \{a_1, a_2, \ldots, a_p\}, \quad B = \{b_1, b_2, \ldots, b_q\},$$

т. е. $|A| = p$, $|B| = q$.

Упорядочение элементов в этих множествах выбрано произвольно, однако, однажды выбранное, оно далее остается фиксированным. Пусть это отношение $\rho$ определено посредством выбора пар $(a, b)$, где $a \in A$, $b \in B$.

Рассмотрим матрицу $M$ над $\{0, 1\}$, т. е. $M$: $\mathbf{N}_p \times \mathbf{N}_q \to \{0, 1\}$, и свяжем элементы $M$ с отношением $\rho$ биекцией

$$\varphi: \mathcal{P}(A \times B) \to \mathcal{M}(p, q, \mathbf{Z}_2)$$

($\varphi$ отображает *произвольное* отношение между $A$ и $B$ в матрицу $p \times q$ над $\{0, 1\}$);

$$\varphi: \rho \to M,$$

причем

$$(\varphi(\rho))_{ij} = M_{ij} = \begin{cases} 1, & \text{если } (a_i, b_j) \in \rho, \\ 0, & \text{если } (a_i, b_j) \notin \rho. \end{cases}$$

В случае, когда полезно подчеркнуть, что матрица $M$ была получена из отношения $\rho$, мы будем обозначать ее через $M(\rho)$.

Пример 1.1. Возьмем случай $|A| = 4$, $|B| = 3$ и

$$\rho = \{(a_1, b_2), (a_1, b_3), (a_2, b_1), (a_3, b_1), (a_4, b_2)\}.$$

Тогда соответствующая матрица $M$ имеет вид

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad //$$

Таким образом, мы имеем способ табулирования или кодирования отношения и можем закодировать отношение посредством $\varphi$ или декодировать посредством $\varphi^{-1}$. Этот процесс является отображением $(i, j)$ в $A \times B$ или $M$ соответственно. Такое представление более удобно, чем теоретико-множественный способ определения отношений, поскольку с ним можно обращаться формальным образом. Оно становится даже более пригодным для вычислений,

если наложить некоторую структуру на множество, из которого получается матрица. Возьмем опять {0, 1} и определим на этом множестве логическое сложение (**или**) и умножение (**и**). Тогда, если $M$ и $N$ — матрицы $p \times q$, соответствующие отношениям $\rho$ и $\sigma$, то матрица $Q$, представляющая отношение $\tau$, где

$$\tau = \{(a, b): (a, b) \in \rho \text{ или } (a, b) \in \sigma\},$$

определяется следующим образом $Q_{ij} = (M_{ij} \text{ или } N_{ij}) = M_{ij} + N_{ij}$ (логическое сложение). Следовательно, имеет смысл называть $Q$ *суммой* матриц $M$ и $N$ и писать

$$Q = M + N,$$

подразумевая, что $Q$, $M$ и $N$ имеют один и тот же размер и $Q$ вычисляется по правилу покомпонентного сложения

$$Q_{ij} = M_{ij} + N_{ij}.$$

Это — пример использования коммутативной диаграммы, изображенной на рис. 6.1, где производится операция на одном множестве с использованием операции на другом множестве посредством подходящего отображения $\varphi$. С этой диаграммой обычно связывается тождество

$$\varphi(\rho \cup \sigma) = \varphi(\rho) + \varphi(\sigma).$$

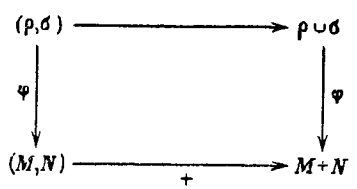


Рис. 6.1

С помощью этого тождества можно дать более точное определение сложения матриц:

$$M + N = \varphi(\rho) + \varphi(\sigma) = \varphi(\rho \cup \sigma) = \varphi(\varphi^{-1}(M) \cup \varphi^{-1}(N)).$$

Пример 1.1 (продолжение). Пусть $A$ и $B$ те же, что и раньше, и пусть

$$\sigma = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_3, b_2)\}.$$

Тогда

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

Дальше будет видно, что

$$\rho \cup \sigma = \{(a_1, b_1), (a_1, b_2), (a_1, b_3), (a_2, b_1), (a_3, b_1), (a_3, b_2), (a_4, b_2)\}$$

и, что эквивалентно,

$$M + N = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \ //$$

Более того, если мы возьмем множество $C = \{c_1, c_2, c_3, c_4, c_5\}$ и рассмотрим отображение $\pi$ между $B$ и $C$, определенное следующим образом:

$$\pi = \{(b_1, c_1), (b_1, c_5), (b_2, c_2), (b_3, c_4), (b_3, c_5)\},$$

то оно может быть представлено в виде матрицы $P$, где

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

Очевидно, что отношение $\pi \circ \rho$ между $A$ и $C$ корректно определено и, следовательно, будет соответствовать матрице $4 \times 5$. Обозначим эту матрицу через $S$. Как ее можно вычислить? Для этого надо вычислить $S_{ij}$ для всех $i$, $j$, где $1 \leqslant i \leqslant 4$, $1 \leqslant j \leqslant 5$. В силу биекции $S_{ij} = 1$ тогда и только тогда, когда $(a_i, c_j) \in \pi \circ \rho$. Однако это так, если только существует некоторое $b \in B$ такое, что $(a_i, b) \in \rho$ и $(b, c_j) \in \pi$, т. е.

$$(a_i, c_j) \in \pi \circ \rho \equiv (a_i, b_1) \in \rho \text{ и } (b_1, c_j) \in \pi$$
$$\text{или } (a_i, b_2) \in \rho \text{ и } (b_2, c_j) \in \pi,$$
$$\text{или } (a_i, b_3) \in \rho \text{ и } (b_3, c_j) \in \pi;$$

или же, что эквивалентно,

$$S_{ij} = M_{i1} * P_{1j} + M_{i2} * P_{2j} + M_{i3} * P_{3j} = \sum_{k=1}^{3} M_{ik} * P_{kj}.$$

Матрица $S$, вычисленная по такому правилу, называется *произведением* $M$ и $P$ и обозначается через $M * P$ или просто $MP$.

Рассмотрим опять естественное (коммутативное) отношение между двумя рассматриваемыми операторами (рис. 6.2). Тогда

$$M * P = \varphi(\varphi^{-1}(P) \circ \varphi^{-1}(M)).$$

Замечание. Изменение порядка $\varphi$ зависят от способа определения матрицы отношения; если (вместо этого) мы определим матрицу отношения следующим

образом:

$M_{ij} = 1$ тогда и только тогда, когда $(a_j, b_i) \in \rho$,

то изменения порядка не будет. Хотя с математической точки зрения было бы более желательно иметь один и тот же порядок, это нарушило бы сложившуюся практику. Соответствующие диаграммы в § 3 не меняют порядок, однако эти соглашения естественны для вопросов, изучаемых в этом параграфе.

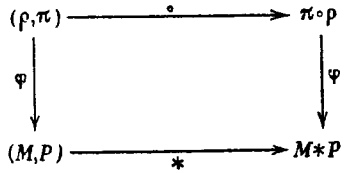


Рис. 6.2

Пример 1.1 (продолжение). Выполним вычисления, соответствующие определенным выше отношениям. Получаем

$$MP = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad //$$

Если матрицы $M$ и $N$ имеют одинаковый размер, то их сумма существует и определяется формулой

$$(M + N)_{ij} = M_{ij} + N_{ij},$$

а если матрицы $P$ и $M$ согласованы ($M$ имеет размерность $p \times q$, а $P$ — размерность $q \times r$), то умножение матрицы $M$ на $P$ возможно и определяется следующим образом:

$$(MP)_{ij} = \sum_{k=1}^{q} M_{ik} * P_{kj}.$$

Хотя матрицы рассматриваются над $(\mathbf{Z}_2, \wedge, \vee)$, мы используем символы $*$ и $+$ для того, чтобы иметь возможность обобщения введенных выше операций (см. § 2). С этого момента обозначения $\wedge$ и $\vee$ будут использоваться лишь в тех случаях, когда общие операции им неадекватны.

В заключительной части этого параграфа ограничимся рассмотрением матриц, представляемых отношениями на конечном множестве $A$, где $|A| = n$. Тогда все матрицы согласованы и их сумма и произведение всегда определены.

Из покомпонентного определения сложения сразу следует, что сложение матриц коммутативно и существует

*нулевая* $(n \times n)$*-матрица* 0: $0_{ij} = 0$ для всех $i$, $j$; $1 \leqslant i$, $j \leqslant n$. С другой стороны, умножение матриц, вообще говоря, некоммутативно, однако существует единица, которая называется *единичной* $(n \times n)$*-матрицей* и определяется следующим образом: $I$: $I_{ij} = 1$, если $i = j$, и $I_{ij} = 0$, если $i \neq j$. Так, если $X$ — матрица $n \times n$ и $Y = XI$, то

$$Y_{ij} = \sum_{p=1}^{n} X_{ip} * I_{pj} = \sum_{\substack{p=1 \\ (p \neq j)}}^{n} X_{ip} * I_{pj} + X_{ij} * I_{jj},$$

Так как все $I_{pj} = 0$, за исключением случая $p = j$, то в сумме все члены, исключая те, где $p = j$, равны нулю. Кроме того, $I_{jj} = 1$. Поэтому

$$Y_{ij} = X_{ij}, \quad \text{т. е.} \quad Y = X.$$

Следовательно, $X = XI$. Аналогично $IX = X$; поэтому

$$IX = X = XI. \ //$$

К сожалению, обратная по умножению матрица может не существовать; однако если она существует, то она единственна. Если матрица имеет обратную, то она называется обратимой.

Пример 1.2. Не существует матрицы $X$ такой, что

$$X \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = I.$$

Доказательство. Вычисление произведения дает

$$\begin{bmatrix} 0 & X_{12} \\ 0 & X_{22} \end{bmatrix}.$$

Следовательно, какие бы значения компонент матрицы $X$ не рассматривались, элемент (1, 1) произведения никогда не будет равен 1, откуда и следует требуемый результат. //

Таким образом, множество квадратных матриц заданного размера с определенными на нем операциями умножения и сложения образует кольцо.

Используя далее связь между бинарными отношениями на множестве и матрицами над $(\mathbf{Z}_2, \wedge, \vee)$, дадим следующие определения.

*Транспонированной* матрицей $M$ называется матрица $M^T$ такая, что

$$M^T_{ij} = M_{ji}$$

(поэтому, если $M$ получена из отношения $\sigma$, то $M^T$ может быть получена из отношения $\sigma^{-1}$); *транзитивное замыка-*

*ние* $M^+$ и *рефлексивное замыкание* $M^*$ (изоморфны соответственно $\sigma^+$ и $\sigma^*$) определяются следующим образом:

$$M^+ = \sum_{n=1}^{\infty} M^n, \quad M^* = \sum_{n=0}^{\infty} M^n,$$

где $M^0 = I$, $M^1 = M$ и $M^{n+1} = MM^n$ ($n \in \mathbf{N}$). (В некоторых случаях эти замыкания нельзя определить корректно (чтобы соответствующие ряды сходились), однако над $(\mathbf{Z}_2, \bigwedge, \bigvee)$ определение корректно, поскольку это — замкнутое полукольцо.)

В заключение заметим, что матрицы могут быть частично упорядочены путем поэлементного сравнения, а именно

$M \leqslant N$ тогда и только тогда, когда $M_{ij} \leqslant N_{ij}$ для всех $i, j$.

Из данного определения следует, что

$M \leqslant N$ тогда и только тогда, когда $M + N = N$,

при условии что $+$ является операцией «максимум», подобной или.

Упражнение 6.1.

1. Пусть $A$ — конечное множество и $|A| = n$, а $M$ — матрица над $(\mathbf{Z}_2, \bigwedge, \bigvee)$, соответствующая некоторому бинарному отношению на $A$. Доказать, что

$$M^+ = \sum_{p=1}^{n} M^p.$$

(Следствием этого является тот факт, что вместо полукольца $(\mathbf{Z}_2, \bigwedge, \bigvee)$ мы можем рассматривать булеву алгебру $(\mathbf{B}, \bigwedge, \bigvee, \neg)$, где $\mathbf{B} = \{0, 1\}$. Поэтому мы часто будем обозначать множество матриц $n \times n$ через $\mathcal{M}(n, \mathbf{B})$ и называть их *булевыми матрицами*.)

2. Доказать, что если $M$ — конечная квадратная матрица над $(\mathbf{Z}_2, \bigwedge, \bigvee)$, то

$$M^* = (I + M)^+.$$

Указание: см. задачу 1.

3. Показать, что если матрица $M$ над $(\mathbf{Z}_2, \bigwedge, \bigvee)$ такая, что $I \leqslant M$, то $M^n \leqslant M^{n+1}$ для любого $n \in \mathbf{N}$. В качестве следствия доказать, что если $M$ имеет размер $n \times n$ и $p \geqslant m$, то

$$M^* = (I + M)^p, \quad M^* = M^{2^q}$$

для некоторого $q$ такого, что $2^q \geqslant m$.

4. Показать, что если существует обратная к $M$ матрица $M^{-1}$ (т. е. $M^{-1}M = MM^{-1} = I$), то она единственна.

Доказать также, что если $N$ обратима и согласована с $M$, то

$$(NM)^{-1} = M^{-1}N^{-1}.$$

5. Доказать, что если $A$, $B$ и $C$ — согласованные матрицы такие, что

$$A * B = 0 = A * C,$$

то отсюда не следует равенство $B = C$. //

## § 2. Матрицы над другими алгебраическими структурами

Мы уделяем много внимания матрицам над полукольцами $(Z_2, \wedge, \vee)$ и $(B, \wedge, \vee, \neg,)$ однако ничего не сказали о матрицах, чьи элементы принадлежат другим структурам. Сначала мы рассмотрим вопрос о том, что надо требовать от структуры $(S, *, +)$ для того, чтобы матрицы над $S$ обладали нужными свойствами. Затем обсудим технику вычислений, применяемую в ситуациях, когда операция замыкания корректно определена.

**2.1. Обобщенные матрицы.** Пусть $(\mathcal{M}, \circledast, \oplus)$ обозначает множество $\mathcal{M}(n, (S, *, +))$ матриц $n \times n$ над $(S, *, +)$ с операциями $\circledast$ и $\oplus$, определенными в терминах $*$ и $+$. Можно легко определить матрицы над произвольным непустым множеством, однако индуцированные при этом операции над матрицами могут быть некорректными и обладать неестественными свойствами. Рассмотрим вначале операцию сложения.

Если $M$ и $N$ — согласованные матрицы над $S$, тогда по определению

$$(M \oplus N)_{ij} = M_{ij} + N_{ij}.$$

Для получения $\oplus$ использовалась только одна операция сложения $+$. Поэтому сложение матриц выполняется просто. Однако для того, чтобы операция $\oplus$ была коммутативна и ассоциативна, теми же самыми свойствами должно обладать $(S, +)$. Отсюда следует, что нулевая матрица $\mathcal{M}$ существует тогда и только тогда, когда существует двусторонняя*) единица 0 по отношению к $+$ в $S$. Аналогично существование аддитивных обратных элементов в $\mathcal{M}$ зависит от существования аддитивных обратных элементов в $S$. Поэтому сложение $\oplus$ на $\mathcal{M}$ всегда может быть

*) Левая и правая.— *Примеч. ред.*

определено. Причем, хотя для того, чтобы $(\mathcal{M}, \oplus)$ стало коммутативной группой, требуется выполнение многих свойств на $(S, +)$, многое можно получить даже тогда, когда операция $+$ не обладает достаточным набором хороших свойств.

В противоположность этому умножение в $\mathcal{M}$ требует гораздо большего от $S$. По аналогии с матрицами над $(\mathbf{Z}_2, \wedge, \vee)$ положим

$$(M \circledast N)_{ij} = \sum_k M_{ik} * N_{kj}.$$

Поскольку суммирование проводится в $S$, то для корректного определения умножения в $\mathcal{M}$ необходимо, чтобы результат не зависел от порядка суммирования. Это будет выполнено, если *сложение* в $S$ ассоциативно.

Не следует ожидать, что умножение в $\mathcal{M}$ будет коммутативно; даже (так как $(\mathcal{M}, \circledast)$, зависит от $(S, *)$ и $(S, +)$) для того, чтобы обеспечить ассоциативность в $(\mathcal{M}, \circledast)$, требуется больше, чем ассоциативность в $(S, *)$. Рассмотрим следующий пример, в котором будут выполнены левая и правая дистрибутивность $*$ над $+$, ассоциативность в $(S, *)$ и ассоциативность и коммутативность в $(S, +)$.

Пример 2.1. Пусть $M$, $N$ и $P$ — матрицы размерности $1\times 2$, $2\times 3$ и $3\times 1$ соответственно. Тогда

$$((M\circledast N)\circledast P)_{11}=\sum_{j=1}^{3}(M\circledast N)_{1j}*P_{j1}=\sum_{j=1}^{3}\left(\sum_{i=1}^{2}M_{1i}*N_{ij}\right)*P_{j1}=$$

$$=((M_{11}*N_{11})+(M_{12}*N_{21}))*P_{11}+((M_{11}*N_{12})+$$

$$+(M_{12}*N_{22}))*P_{21}+((M_{11}*N_{13})+(M_{12}*N_{23}))*P_{31}=$$

$$=((M_{11}*N_{11})*P_{11})+((M_{12}*N_{21})*P_{11})+$$

$$+((M_{11}*N_{12})*P_{21})+((M_{12}*N_{22})*P_{21})+((M_{11}*N_{13})*P_{31})+$$

$$+((M_{12}*N_{23})*P_{31})=(M_{11}*(N_{11}*P_{11}))+(M_{12}*(N_{21}*P_{11}))+$$

$$+(M_{11}*(N_{12}*P_{21}))+(M_{12}*(N_{22}*P_{21}))+(M_{11}*(N_{13}*P_{31}))+$$

$$+(M_{12}*(N_{23}*P_{31}))=(M_{11}*(N_{11}*P_{11}))+(M_{11}*(N_{12}*P_{21}))+$$

$$+(M_{11}*(N_{13}*P_{31}))+(M_{12}*(N_{21}*P_{11})+(M_{12}*(N_{22}*P_{21}))+$$

$$+(M_{12}*(N_{23}*P_{31}))=M_{11}*((N_{11}*P_{11})+(N_{12}*P_{21})+$$

$$+(N_{13}*P_{31}))+M_{12}*((N_{21}*P_{11})+(N_{22}*P_{21})+(N_{23}*P_{31}))=$$

$$=\sum_{i=1}^{2}M_{1i}*\left(\sum_{j=1}^{3}N_{ij}*P_{j1}\right)=(M\circledast(N\circledast P))_{11}$$

При проведении выкладок предполагалось, что $+$ ассоциативна, поэтому некоторые скобки были опущены. //

Пока все это выглядит не очень оптимистично, поскольку лишь небольшое число структур может удовлетворять условиям, накладываемым на $(S, *, +)$. Заметим также, что для существования единичной матрицы требуется, чтобы:

а) $0 * x = 0 = x * 0$ для всех $x \in S$, где $0$ — аддитивная единица в $S$ (напомним, что это условие не является аксиомой поля);

б) $S$ должно иметь двустороннюю мультипликативную единицу.

Тем не менее можно проверить, что если $(S, *, +)$ — *кольцо* или *поле*, тогда все вышеуказанные условия выполнены и $(\mathscr{M}, \circledast, \oplus)$ — *кольцо*.

Понятия упорядочения и замыкания матриц над более узкими структурами, такими как упорядоченные поля и т. п., становятся слишком сложными и поэтому обсуждаться не будут.

Как уже отмечалось в упражнении 6.1, мы можем рассматривать $\mathscr{M}(n, (\mathbf{Z}_2, \wedge, \vee))$ как $\mathscr{M}(n, \mathbf{B})$. Аналогично мы можем обобщить вычисления на другие родственные алгебраические структуры, однако требуется тщательность, чтобы сформулировать условия на то, какую систему использовать. Например, из $\mathbf{Z}_2 \subseteq \mathbf{Z}_3$ путем обычного включения $\mathbf{Z}_2 \subset\!\!\rightarrow \mathbf{Z}_3$ ($A \subset\!\!\rightarrow B$ обозначает тождественное отображение на $A$, где $A \subseteq B$) следует, что

$$A \in \mathscr{M}(n, \mathbf{Z}_2) \Rightarrow A \in \mathscr{M}(n, \mathbf{Z}_3).$$

Пусть теперь дана матрица из 0 и 1. Спрашивается, определена она на $\{0, 1\}$ или же на $\{0, 1, 2\}$? Это необходимо знать для перехода к построению арифметики. Конечно (как будет показано в § 3), все это зависит от того, что́ мы пытаемся вычислять. Аналогичные проблемы возникают с любыми включениями, особенно $\mathscr{M}(n, \mathbf{B})$, $\mathscr{M}(n, \mathbf{Z}_2)$, $\mathscr{M}(n, \mathbf{Z})$, $\mathscr{M}(n, \mathbf{Z}_m)$, $\mathscr{M}(n, \mathbf{Q})$ и $\mathscr{M}(n, \mathbf{R})$.

**2.2. Алгоритм Уоршолла.** В этом разделе мы опишем быстрый способ вычисления (транзитивного или рефлексивного) замыкания квадратных матриц над $(\mathbf{Z}_2, \wedge, \vee)$. Этот способ — один из вариантов метода Уоршолла; представим его в виде программы. Если $M$ — матрица размерности $n \times n$ над $(\mathbf{Z}_2, \wedge, \vee)$, то ее можно преобразовать в $M^+$ следующим образом:

**for** $j$ **from** 1 **to** $n$ **do**
  **for** $i$ **from** 1 **to** $n$ **do**

**if** $i \neq j$ **and** $M_{ij} = 1$ **then**
**for** $k$ **from** 1 **to** $n$ **do**
$M_{ik} \leftarrow M_{ik} \vee M_{jk}$.

Чтобы вычислить $M^*$, мы можем прибавить $I$ или в начале, или в конце, используя соотношение

$$M^* = (M + I)^+ = M^+ + I.$$

Пример 2.2. Пусть

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix},$$

тогда, используя алгоритм, заменим 0 на 1 в следующих элементах:

(1, 4), (1, 5), (3, 4), (3, 5), (5, 2), (5, 5), (2, 2,), (2, 3), (3, 3).

Получим матрицу

$$M^+ = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \text{ //}$$

Мы не будем доказывать, что программа выполняет нужные действия, однако отметим, что смысл программы состоит в том, что если $i \neq j$ и $M_{ij} = 1$, то $(i, j) \in \sigma^p$ для некоторого $p \leqslant n$, где $\sigma$ — отношение, порожденное $M$. Поэтому заменим $i$-ю строку на дизъюнкцию $i$-й и $j$-й строк, чтобы указать, что все, что относится к $j$, также относится и к $i$.

Очевидно, что при $i = j$ выполнение этого шага не дает никакого выигрыша. Основная сложность проверки правильности программы — убедиться в том, что ничего не пропущено. Читатель сможет доказать это самостоятельно, прочитав гл. 7.

Метод Уоршолла дает значительный выигрыш по сравнению с прямым использованием определения замыкания; он может быть еще улучшен, однако это требует использования более сложных структур данных, и поэтому мы не будем больше этим заниматься.

Этот метод можно многими путями приспособить для получения различных оценок, связанных с матрицами (отношениями), простейшим из которых является понятие

«расстояния» между элементами множества, связанными отношением σ. Расстояние между двумя точками $x$ и $y$ ($d(x, y)$) равно наименьшему $n$ такому, что $n \in \mathbf{N}$ и $y \in \sigma^n(x)$. Если $M$ — матрица, соответствующая σ, то, заменяя циклы следующим образом, получим требуемый результат:

**if** $M_{ij} \neq 0$ **then**
  **for** $k$ **from** 1 **to** $n$ **do**
    **if** $M_{jk} \neq 0$ **and**
      ($M_{ik} = 0$ **or** $M_{ik} > M_{ij} + M_{jk}$)
    **then** $M_{ik} \leftarrow M_{ij} + M_{jk}$.

Здесь используется арифметика над $\mathbf{Z}$ (а не над $\mathbf{Z}_2$).

Пример 2.3. Применение описанной выше процедуры к матрице из предыдущего примера дает

$$\begin{bmatrix} 0 & 1 & 1 & 2 & 2 \\ 0 & 3 & 2 & 1 & 1 \\ 0 & 1 & 3 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & 3 \end{bmatrix}. \quad //$$

Упражнение 6.2.

1. Определение. Говорят, что две $(n \times n)$-матрицы $X$ и $Y$ над полем $F$ *подобны над* $F$, если существует обратимая матрица $P$ над $F$ такая, что

$$X = P^{-1}YP. \quad //$$

Показать, что отношение, индуцируемое подобием на $\mathcal{M}(n, F)$, является отношением эквивалентности и что матрица $I$ подобна только себе.

2. Показать, что если $A, B \in \mathcal{M}(n, F)$ для некоторого $n \in \mathbf{N}$ и некоторого поля $F$, тогда

$$(A + B)^T = A^T + B^T, \qquad (AB)^T = B^T A^T.$$

3. Определение. *Стохастической матрицей* называется действительная матрица над $(\{x: 0 \leqslant x \leqslant 1\}, *, +)$ такая, что сумма элементов в каждой строке равна 1. //

Показать, что множество $3 \times 3$ стохастических матриц не замкнуто по отношению к сложению, но замкнуто по отношению к умножению. Установить, какие свойства поля (R) при этом использовались.

## § 3. Матрицы и векторные пространства

Результаты § 2 показывают, как можно применять матрицы для проверки выполнения отношений между конечными множествами соответствующих размеров. Цель

этого параграфа — показать, как использовать матрицы над полем $(\mathbf{R}, *, +)$ (далее просто, $\mathbf{R}$) для того, чтобы выполнять некоторые преобразования, в частности линейные преобразования в векторном пространстве $V$ над $\mathbf{R}$. В замечании из § 4 гл. 5 было установлено, что если $T \in \mathrm{End}(V)$ — множество линейных преобразований $V$, то

$$\{(x, Tx): x \in T\} \subseteq V \times V$$

является бинарным отношением над $V$. Мы ищем результат применения линейного преобразования на $V$ в $\mathcal{M}(n, \mathbf{R})$, где $n = \dim(V)$.

Возможны два обобщения. Первое — рассмотрение произвольных полей, второе — линейные преобразования между векторными пространствами произвольных размерностей, которые могут быть определены аналогичным образом в $\mathcal{M}(n, m, \mathbf{R})$. Так как эти обобщения нам не потребуются, то в дальнейшем рассматривать их не будем.

**3.1. Матричные представления линейных преобразований.** Операции сложения и умножения матриц в $\mathcal{M}(n, \mathbf{R})$ неявно были определены в § 2. Если $A \in \mathcal{M}(n, \mathbf{R})$ имеет элементы $A_{ij}$ и $\lambda \in \mathbf{R}$, определим $\lambda A \in \mathcal{M}(n, \mathbf{R})$ как матрицу с элементами $\lambda A_{ij}$. Эту операцию называют *умножением матрицы на скаляр*. Поскольку $\mathcal{M}(n, \mathbf{R})$ играет центральную роль в оставшейся части этой главы, то полезно перечислить все его свойства относительно определенных выше операций. Единичную матрицу в $\mathcal{M}(n, \mathbf{R})$ будем обозначать через $I$ при всех $n \in \mathbf{N}$.

Предложение. *$\mathcal{M}(n, \mathbf{R})$ является линейной алгеброй.* //

Мы не даем доказательства этого факта. Рекомендуем вначале проверить выполнение аксиом линейной алгебры для $\mathcal{M}(2, \mathbf{R})$, после чего будет видно, как можно построить доказательство в общем случае. Надо показать, что:

а) $\mathcal{M}(n, \mathbf{R})$ является векторным пространством;

б) $\mathcal{M}(n, \mathbf{R})$ — кольцо;

в) умножение матрицы на скаляр обладает следующим свойством:

$$\lambda(AB) = (\lambda A)B = A(\lambda B) \text{ для всех } \lambda \in \mathbf{R},\ A,\ B \in \mathcal{M}(n, \mathbf{R}).$$

Полезно иметь специальное обозначение для подмножества обратимых матриц из $\mathcal{M}(n, \mathbf{R})$. Обозначим это подмножество через

$$GL(n, \mathbf{R}) = \{A \in \mathcal{M}(n, \mathbf{R}): A^{-1} \text{ существует}\}.$$

Каждое $GL(n, \mathbf{R})$, $n \in \mathbf{N}$, определяет группу по умноже-

нию. Эти группы называют *полными линейными группами*.

Пусть $V$ — векторное пространство размерности $n$ над $\mathbf{R}$ и $T \in \mathrm{End}(V)$. Если $B = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ — базис в $V$, то очевидно, что $T\mathbf{e}_i \in V$ для всех $i$, $1 \leqslant i \leqslant n$. Следовательно, должны существовать $t_{ij} \in \mathbf{R}$ $(1 \leqslant i, j \leqslant n)$ такие, что

$$T\mathbf{e}_1 = t_{11}\mathbf{e}_1 + t_{21}\mathbf{e}_2 + \ldots + t_{n1}\mathbf{e}_n,$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$T\mathbf{e}_n = t_{1n}\mathbf{e}_1 + t_{2n}\mathbf{e}_2 + \ldots + t_{nn}\mathbf{e}_n.$$

Пусть $A_T$ — матрица вида

$$A_T = \begin{bmatrix} t_{11} & t_{22} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ t_{n1} & t_{n2} & \cdots & t_{nn} \end{bmatrix};$$

тогда ее называют *матрицей преобразования $T$ в базисе $B$*. Для данного $B$ матрица $A_T$ единственна; таким образом, мы можем определить отображение

$$\varphi^B\colon \mathrm{End}(V) \to \mathscr{M}(n, \mathbf{R})$$

следующим образом:

$$\varphi^B(T) = A_T.$$

Так как $A_T$ можно вычислить, то ее можно использовать для нахождения $T$.

Предложение. *Пусть $T \in \mathrm{End}(V)$ соответствует матрица $A_T$ в базисе $B = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$. Тогда, если $\mathbf{x} \in V$ — вектор*

$$\mathbf{x} \in \sum_{i=1}^{n} \lambda_i \mathbf{e}_i,$$

*то*

$$T\mathbf{x} = \sum_{i=1}^{n} \mu_i \mathbf{e}_i \in V,$$

*где*

$$\begin{bmatrix} \mu_1 \\ \cdots \\ \mu_n \end{bmatrix} = A_T \begin{bmatrix} \lambda_1 \\ \cdots \\ \lambda_n \end{bmatrix}.$$

Доказательство. Пусть $\mathbf{x} = \sum_{i=1}^{n} \lambda_i \mathbf{e}_i$. Тогда

$$T\mathbf{x} = \sum_{i=1}^{n} \lambda_i T\mathbf{e}_i = \sum_{i=1}^{n} \lambda_i \left( \sum_{j=1}^{n} t_{ij}\mathbf{e}_j \right) = \sum_{i=1}^{n} \left( \sum_{j=1}^{n} t_{ij}\lambda_j \right) \mathbf{e}_i. \quad /\!/$$

Следовательно, в данном базисе линейное преобразование $T$ можно выполнить с помощью произведения $A_T\Lambda$, где $A_T$ имеет размерность $n \times n$, а $\Lambda$ — вектор (матрица размерности $n \times 1$)

$$\Lambda = \begin{bmatrix} \lambda_1 \\ \cdots \\ \lambda_n \end{bmatrix},$$

соответствующий $\mathbf{x} \in V$. На самом деле можно установить гораздо больше. Ниже мы установим важные свойства $\varphi^B$.

Предложение. *Отображение* $\varphi^B$: $\mathrm{End}(V) \to \mathscr{M}(n, \mathbf{R})$ *является изоморфизмом линейной алгебры с обычными операциями в* $\mathrm{End}(V)$ *и* $\mathscr{M}(n, \mathbf{R})$ *на* $\mathbf{R}$, *причем сужение* $\varphi^B$ *на группу* $\mathrm{Aut}(V) \subseteq \mathrm{End}(V)$ *является группой изоморфизмов на* $GL(n, \mathbf{R})$.

Доказательство. Отметим основные моменты. Чтобы избежать большого количества индексов, ограничимся случаем $n = 2$. В общем случае доказательство аналогично. Изоморфизм линейной алгебры следует из следующих утверждений:

а) $\varphi^B$ биективно;

б) $\varphi^B(I_V) = I$;

в) $\varphi^B(ST) = \varphi^B(S)\varphi^B(T)$ для всех $S, T \in \mathrm{End}(V)$;

г) $\varphi^B(\lambda S + \mu T) = \lambda\varphi^B(S) + \mu\varphi^B(T)$ для всех $S, T \in \mathrm{End}(V)$ и $\lambda, \mu \in \mathbf{R}$.

Докажем некоторые из этих утверждений; остальные оставим в качестве упражнений. Пусть $\{\mathbf{e}_1, \mathbf{e}_2\}$ — базис в $V$.

а) Чтобы доказать, что $\varphi^B$ инъективно, необходимо показать, что

$$\varphi^B(S) = \varphi^B(T) \Rightarrow S = T.$$

Если

$$S\mathbf{e}_1 = s_{11}\mathbf{e}_1 + s_{21}\mathbf{e}_2, \qquad T\mathbf{e}_1 = t_{11}\mathbf{e}_1 + t_{21}\mathbf{e}_2,$$

то

$$\varphi^B(S) = \varphi^B(T) \Rightarrow s_{11} = t_{11}, \qquad s_{21} = t_{21}.$$

Поэтому $S\mathbf{e}_1 = T\mathbf{e}_1$. Аналогично $S\mathbf{e}_2 = T\mathbf{e}_2$. Следовательно, $S$ и $T$ совпадают на базисных элементах $\mathbf{e}_1$ и $\mathbf{e}_2$. Однако для всех $\mathbf{x} \in V$ выполняется соотношение

$$\mathbf{x} = \lambda\mathbf{e}_1 + \mu\mathbf{e}_2;$$

таким образом,

$$S\mathbf{x} = \lambda S\mathbf{e}_1 + \mu S\mathbf{e}_2 = \lambda T\mathbf{e}_1 + \mu T\mathbf{e}_2 = T(\lambda\mathbf{e}_1 + \mu\mathbf{e}_2) = T\mathbf{x},$$

т. е. $S = T$.

Доказательство сюръективности оставляем в качестве упражнения.

б) $I_V\mathbf{x} = \mathbf{x}$ для всех $\mathbf{x} \in V$; следовательно,

$$I_V\mathbf{e}_1 = \mathbf{e}_1 + 0\mathbf{e}_2, \quad I_V\mathbf{e}_2 = 0\mathbf{e}_1 + \mathbf{e}_2,$$

$$\varphi^B(I_V) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I.$$

в) Пусть

$$\varphi^B(S) = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}, \quad \varphi^B(T) = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix};$$

тогда

$$T\mathbf{e}_1 = t_{11}\mathbf{e}_1 + t_{21}\mathbf{e}_2,$$

$$ST\mathbf{e}_1 = t_{11}S\mathbf{e}_1 + t_{21}S\mathbf{e}_2 = t_{11}(s_{11}\mathbf{e}_1 + s_{21}\mathbf{e}_2) + t_{21}(s_{12}\mathbf{e}_1 + s_{22}\mathbf{e}_2) =$$
$$= (t_{11}s_{11} + t_{21}s_{12})\mathbf{e}_1 + (t_{12}s_{21} + t_{21}s_{22})\mathbf{e}_2.$$

Аналогично

$$ST\mathbf{e}_2 = (t_{12}s_{11} + t_{22}s_{12})\mathbf{e}_1 + (t_{12}s_{21} + t_{22}s_{22})\mathbf{e}_2.$$

Следовательно,

$$\varphi^B(ST) = \varphi^B(S)\varphi^B(T),$$

что и требовалось доказать.

г) Утверждение доказывается аналогично утверждению в).

Чтобы показать, что сужение $\varphi^B$ на $\mathrm{Aut}(V)$ является группой изоморфизмов, используем свойства б)—г). Пусть $T \in \mathrm{Aut}(V)$; тогда

$$I = \varphi^B(I_V) = \varphi^B(TT^{-1}) = \varphi^B(T)\varphi^B(T^{-1}),$$

следовательно,

$$T \in \mathrm{Aut}(V) \Rightarrow \varphi^B(T) \in GL(n, \mathbf{R}),$$
$$\varphi^B(T)^{-1} = \varphi^B(T^{-1}). \ //$$

Этот результат играет существенную роль, так как имеет много важных следствий. Выведем некоторые из них. В фиксированном базисе отображение $T \to A_T$ из $\mathrm{End}(V)$ в $\mathscr{M}(n, \mathbf{R})$ биективно (каждому преобразованию

соответствует единственная матрица и наоборот). Далее

$$A_{S\circ T} = A_S A_T;$$

это эквивалентно тому, что диаграмма на рис. 6.3 коммутативна. На практике это просто означает, что матрица произведения преобразований $S \circ T$ в $\mathrm{End}(V)$ может быть вычислена путем умножения матрицы $A_S$ на матрицу $A_T$. Нет необходимости явно определять $S \circ T$, а затем вычислять $A_{S\circ T}$ на основе определения. Аналогичные результаты получаются и для $S + T$. Кроме того, для сужения $\varphi^B$ на обратимые преобразования $\mathrm{Aut}(V)$ получаем

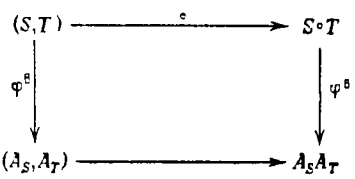


Рис. 6.3

$$A_{T^{-1}} = A_T^{-1};$$

это означает, что для вычисления $A_{T^{-1}}$ необходимо лишь найти матрицу, обратную к $A_T$.

Подчеркнем еще раз, что изоморфизм $\varphi^B$ зависит от базиса; в другом базисе мы будем иметь другой изоморфизм между $\mathrm{End}(V)$ и $\mathcal{M}(n, \mathbf{R})$. Таким образом, элемент $\mathcal{M}(n, \mathbf{R})$ может рассматриваться как представление элемента $\mathrm{End}(V)$ в фиксированном базисе или как представление различных элементов $\mathrm{End}(V)$ в различных базисах. В гл. 10, где результаты этого параграфа будут применяться к $\mathbf{R}^n$ $(n = 2, 3, 4)$, отображение $\varphi^B$ будет рассматриваться в стандартном базисе $\{\widehat{\mathbf{e}}_1, \ldots, \widehat{\mathbf{e}}_n\}$, как это определялось в § 4 гл. 5. Для этого базиса, если $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{R}^n$, то

$$\mathbf{x} = \sum_{i=1}^{n} x_i \widehat{\mathbf{e}}_i, \quad T\mathbf{x} = A_T \begin{bmatrix} x_1 \\ \cdots \\ x_n \end{bmatrix}.$$

Пусть отображение $T \in \mathrm{End}(V)$ имеет собственный вектор $\mathbf{x} \in V$, соответствующий собственному значению $\sigma \in \mathbf{R}$. Тогда, если в базисе $B = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ пространства $V$ вектор $\mathbf{x}$ имеет вид

$$\mathbf{x} = \sum_{i=1}^{n} \lambda_i \mathbf{e}_i,$$

то

$$A_T \begin{bmatrix} \lambda_1 \\ \cdots \\ \lambda_n \end{bmatrix} = \sigma \begin{bmatrix} \lambda_1 \\ \cdots \\ \lambda_n \end{bmatrix}.$$

Поэтому координатные вектора в представлении $\mathbf{x}$ являются собственными векторами $A_T$, соответствующими тому же самому собственному значению.

**3.2. Некоторые другие понятия теории матриц.** Определим теперь на $\mathcal{M}(n, \mathbf{R})$ отображение, называемое *детерминантом* (*определителем*):

$$\det\colon \mathcal{M}(n, \mathbf{R}) \to \mathbf{R}.$$

Было бы естественным ввести это понятие и исследовать его свойства в § 4 гл. 5, так как det не зависит от базиса. Это означает, что если $A_T \in \mathcal{M}(n, \mathbf{R})$ и $A_{T'} \in \mathcal{M}(n, \mathbf{R})$ — матрицы отображения $T \in \mathrm{End}(V)$ в базисах $B$ и $B'$ соответственно, то $\det A_T = \det A_{T'}$. Однако, чтобы определить det на $\mathrm{End}(V)$, мы должны были бы ввести понятия из тензорной алгебры. Вместо этого дадим хорошо известное определение det на $\mathcal{M}(n, \mathbf{R})$ и установим некоторые из наиболее важных его свойств.

Определение det будем давать при помощи рекурсии. Вначале определим det для случая $\mathcal{M}(2, \mathbf{R})$. Пусть

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

тогда

$$\det A = a_{11}a_{22} - a_{21}a_{12}.$$

Если $A \in \mathcal{M}(n, \mathbf{R})$ имеет элементы $a_{ij}$ $(1 \leqslant i, j \leqslant n)$, то *минором* элемента $a_{kl}$ называется матрица $A^{(k,l)} \in \in \mathcal{M}(n-1, \mathbf{R})$, полученная из $A$ путем вычеркивания $k$-й строки и $l$-го столбца. Теперь det: $\mathcal{M}(n, \mathbf{R}) \to \mathbf{R}$ можно определить рекурсивно для всех $n \in \mathbf{N}$ как

$$\det A = \sum_{l=1}^{n} (-1)^{l+1} a_{1l} \det A^{(1,l)}.$$

Иногда эту формулу называют разложением по первой строке. Можно показать, что если использовать любую другую строку или столбец для формирования соответствующего выражения, то сумма будет равна $\det A$.

Другими словами,

$$\det A = \begin{cases} \sum\limits_{l=1}^{n} (-1)^{l+r} a_{rl} \det A^{(r,l)}, & 1 \leqslant r \leqslant n, \\ \sum\limits_{r=1}^{n} (-1)^{l+r} a_{rl} \det A^{(r,l)}, & 1 \leqslant l \leqslant n. \end{cases}$$

Для малых значений $n$ каждая из этих формул подходит

для непосредственного вычисления $\det A$. Чтобы минимизировать число операций разложения, следует начинать со строки или столбца, содержащих наибольшее количество нулей. Подчеркнем, однако, что эти разложения являются в общем случае не подходящими для вычисления и существуют более эффективные вычислительные процедуры для нахождения det. Некоторые важные свойства det приведены ниже.

Предложение. *Отображение* $\det\colon \mathcal{M}(n, \mathbf{R}) \to \mathbf{R}$ *удовлетворяет следующим условиям:*

а) $\det I = 1$;

б) $\det A = \det A^T$ *для всех* $A \in \mathcal{M}(n, \mathbf{R})$;

в) $\det \lambda A = \lambda^n \det A$, $\lambda \in \mathbf{R}$, $A \in \mathcal{M}(n, \mathbf{R})$;

г) $\det AB = \det A \det B$;

д) $A \in GL(n, \mathbf{R})$ *тогда и только тогда, когда* $\det A \neq 0$. //

Мы не будем давать доказательства этих утверждений. Предлагаем читателю проверить их для $\mathcal{M}(2, \mathbf{R})$. Заметим также, что свойство д) характеризует $GL(n, \mathbf{R})$.

Удобно выделить некоторые матрицы с «особыми» свойствами, которые будут использоваться в дальнейшем. Если $A \in \mathcal{M}(n, \mathbf{R})$ и $A = A^T$, то говорят, что $A$ *симметрична*; если $A = -A^T$, то матрица $A$ *кососимметрична*. Если

$$AA^T = A^TA = I \qquad (\text{т. е. } A^{-1} = A^T),$$

тогда говорят, что $A$ *ортогональна*. Множество всех ортогональных матриц $\mathcal{M}(n, \mathbf{R})$ обозначается через $O(n)$. Через $SO(n)$ обозначим подмножество $O(n)$, состоящее из матриц с единичным детерминантом. Элементы $SO(n)$ называют *специальными ортогональными матрицами*.

Предложение.

а) $O(n)$ *является подгруппой* $GL(n, \mathbf{R})$;

б) $SO(n)$ *является подгруппой* $O(n)$.

Доказательство.

в) Надо показать, что $O(n)$ замкнуто по умножению и для каждой $A \in O(n)$ существует обратная матрица $A^{-1} \in O(n)$. Если $A, B \in O(n)$, то $AA^T = A^TA = I$ и $BB^T = B^TB = I$; следовательно,

$$(AB)(AB)^T = AB(B^TA^T) = A(BB^T)A^T = AA^T = I.$$

Аналогично $(AB)^TAB = I$, и, следовательно, $AB \in O(n)$. Если $A \in O(n)$, то

$$(A^{-1})(A^{-1})^T = A^T(A^T)^T = A^TA = I.$$

Следовательно, $A^{-1} \in O(n)$.

б) Доказательство в этом случае осуществляется подобным образом и оставляется в качестве упражнения. //

Завершим этот параграф кратким обсуждением функций от матриц. Подобно det, это следовало бы рассматривать в § 4 гл. 5. Однако у нас нет аппарата для рассмотрения сумм с бесконечным числом слагаемых в $\mathrm{End}(V)$. Результаты в $\mathscr{M}(n, \mathbf{R})$, приведенные ниже, достаточны для наших целей.

Пусть $A \in \mathscr{M}(n, \mathbf{R})$, тогда матрица $A^2 \in \mathscr{M}(n, \mathbf{R})$ — это по определению матрица $AA$. Аналогично можно определить $A^k$ для всех $k \in \mathbf{N}$, $k > 2$; положим $A^0 = I$. Следовательно, если

$$p\colon \mathbf{R} \to \mathbf{R}$$

есть полиномиальная функция

$$p(x) = \sum_{i=0}^{N} \alpha_i x^i, \quad \text{где} \quad \alpha_i \in \mathbf{R} \quad \text{и} \quad n \in \mathbf{N},$$

то определим $p(A) \in \mathscr{M}(n, \mathbf{R})$ как

$$p(A) = \sum_{i=0}^{N} \alpha_i A^i.$$

Такая матрица существует, поскольку $\mathscr{M}(n, \mathbf{R})$ — векторное пространство. Эту идею можно обобщить. Если $f\colon \mathbf{R} \to \mathbf{R}$ разлагается в сходящийся ряд

$$f(x) = \sum_{i=0}^{\infty} a_i x^i,$$

то имеем смысл выражение $\sum_{i=0}^{\infty} a\, A^i$. Множество $\mathscr{M}(n, \mathbf{R})$ может быть идентифицировано с $\mathbf{R}^{n^2}$. На $\mathbf{R}^{n^2}$ можно определить норму, как указано в § 4 гл. 5. Тогда это индуцирует норму на $\mathscr{M}(n, \mathbf{R})$, и в этом случае бесконечные суммы имеют смысл. В частности, если $A \in \mathscr{M}(n, \mathbf{R})$, то можно показать, что

$$\lim_{N \to \infty} \sum_{k=0}^{N} \frac{A^k}{k!}$$

всегда существует в $\mathscr{M}(n, \mathbf{R})$ и записывается как $\exp A$

или же $e^A$. Когда $n=1$, exp является обычной экспоненциальной функцией, однако при $n>1$ функция exp ведет себя совершенно иным образом (см. упражнение 6.3).

Упражнение 6.3.

1. Доказать, что $\mathcal{M}(2, \mathbf{R})$ с обычными операциями является линейной алгеброй.

2. 1) Определить матрицы линейных преобразований

$$T_1(x, y)=(2x+2y, -x-y),$$
$$T_2(x, y)=(2x+y, -x) \qquad (*)$$

пространства $\mathbf{R}^2$ в

а) стандартном базисе $\mathbf{R}^2$;

б) базисе $B'=\{(1, 0), (1, 1)\}$.

2) Вычислить координаты вектора $\mathbf{a}=(-1, 7)$ в базисе $B'$ и определить вектор $T_1T_2\mathbf{a}$, используя матричное представление.

3) С помощью понятия детерминанта определить, какие из преобразований (*) обратимы. Является ли произведение $T_1T_2$ обратимым?

3. Пусть

$$A=\begin{bmatrix}1 & 2\\ 2 & 4\end{bmatrix}, \quad B=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}, \quad C=\begin{bmatrix}0 & 2\\ -2 & 0\end{bmatrix}.$$

Определить, является ли каждая из этих матриц симметричной, кососимметричной, ортогональной, обратимой.

4. Показать, что у кососимметричной матрицы элементы, стоящие на диагонали, равны нулю.

5. 1) Доказать, что собственные значения симметричной $(2\times 2)$-матрицы всегда действительны.

2) Доказать, что собственные числа ненулевой кососимметричной $(2\times 2)$-матрицы не являются действительными.

6. Пусть $A\in\mathcal{M}(n, \mathbf{R})$. Доказать, что

$$(A\mathbf{a})\cdot\mathbf{b}=\mathbf{a}\cdot(A^T\mathbf{b}) \text{ для всех } \mathbf{a}, \mathbf{b}\in\mathbf{R}^n,$$

и использовать этот факт для доказательства того, что если $A$ симметрична, то собственные вектора, соответствующие *различным* собственным значениям, ортогональны.

7. 1) Пусть

$$A=\begin{bmatrix}1 & 2\\ -1 & 3\end{bmatrix} \qquad (**)$$

и $p(x)=x^2-4x+5$. Доказать, что $p(A)=0$.

2) Использовать (**) для вычисления обратной матрицы $A^{-1}$.

8. 1) Пусть

$$A = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}.$$

Используя индукцию, доказать, что

$$\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}^n = \begin{bmatrix} 1 & na \\ 0 & 1 \end{bmatrix}$$

для всех $n \in \mathbf{N}$, т. е. показать, что $e^A = eA$, и выписать выражение для $\det e^A$.

2) Пусть

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

и $\lambda \in \mathbf{R}$. Определить матрицу $e^{\lambda A}$.

9. 1) Пусть $A, B \in \mathcal{M}(n, \mathbf{R})$. Какие должны быть выполнены условия, чтобы выполнялось соотношение

$$e^{(A+B)} = e^A e^B?$$

2) Использовать предыдущую формулу для доказательства того, что $e^A$ всегда имеет обратную.

10. Пусть $A \in \mathcal{M}(n, \mathbf{R})$. *След* матрицы $A$ (обозначается $\operatorname{tr} A$) определяется по формуле

$$\operatorname{tr} A = \sum_{i=1}^{n} A_{ii}.$$

Если $A \in \mathcal{M}(n, \mathbf{R})$ — матрица с элементами

$$A_{ij} = \begin{cases} \lambda_i & \text{при } i = j, \\ 0 & \text{при } i \neq j, \end{cases}$$

то, используя индукцию, показать, что

$$\det A = \prod_{i=1}^{n} \lambda_i = \lambda_1 \lambda_2 \ldots \lambda_n,$$

т. е. доказать, что в данном случае

$$\det e^A = e^{\operatorname{tr} A}.$$

11. Пусть $A \in \mathcal{M}(n, \mathbf{R})$ имеет собственный вектор $\mathbf{x} \in \mathbf{R}^n$, соответствующий собственному значению $\lambda \in \mathbf{R}$. Доказать, что $\mathbf{x}$ является собственным вектором $e^A$, соответствующим собственному значению $e^\lambda$.

12. 1) Доказать, что если $A \in GL(n, \mathbf{R})$, то $\det A^{-1} = (\det A)^{-1}$.

2) Доказать, что если $A \in O(n)$, то $\det A = \pm 1$.

13. Доказать, что $SO(2)$ является подгруппой $O(2)$.

ГЛАВА 7

# ТЕОРИЯ ГРАФОВ

## § 1. Вводные понятия

Многие отношения на конечных множествах могут быть изображены в виде рисунков (см. § 3 гл. 2), с которыми можно работать при помощи соответствующих матриц. Перед тем как определить конструкции этих рисунков, необходимо быть уверенными в том, что это не повлечет за собой никаких двусмысленностей. Введем необходимые понятия.

Пусть $V$ — конечное множество и

$$I_V = \{(v, v): v \in V\}.$$

Положим

$$V^2_{\neq} = V^2 \setminus I_V = \{(v_1, v_2): v_1 \neq v_2\}$$

и определим на $V^2_{\neq}$ отношение эквивалентности следующим образом:

$$(v_1, v_2) \sim (w_1, w_2), \quad \text{если } (v_1, v_2) = (w_1, w_2) \text{ или } (v_1, v_2) = (w_2, w_1).$$

Важное свойство отношения $\sim$ сформулировано в следующем предложении.

Предложение. *Отношение $\sim$ является отношением эквивалентности на $V^2_{\neq}$.* //

Доказательство оставляем в качестве упражнения.

Множество эквивалентных классов, определенное таким образом, обозначим через $V^2_{\neq}/\sim$. Каждый класс эквивалентности содержит ровно два элемента, так как если $(v_1, v_2) \in V^2_{\neq}$, то $[(v_1, v_2)] = \{(v_1, v_2), (v_2, v_1)\}$. Здесь $[(v_1, v_2)]$ — класс эквивалентности, содержащий $(v_1, v_2)$. Сейчас мы в состоянии дать строгое определение графа.

Определение. *Графом* $G$ называется пара $G = (V, E)$, где $V$ — непустое конечное множество *вершин*, а $E$ — подмножество $V^2_{\neq}/\sim$. //

Другими словами, можно сказать, что *граф* $G$ есть пара $G=(V, E)$, где $V$ — непустое конечное множество *вершин*, а $E$ — множество неупорядоченных пар различных вершин.

Множество $E$ называют множеством *ребер* графа, $|V|$ обозначает число вершин $G$, $|E|$ — число ребер $G$.

Следующий результат выражает связь между графами и классами отношений на конечных множествах.

Предложение.

а) *Граф* $G=(V, E)$ *определяет нерефлексивное симметричное отношение на* $V$.

б) *Нерефлексивное симметричное отношение на конечном множестве* $V$ *определяет граф.*

Доказательство.

а) Пусть $G=(V, E)$ — граф. Определим отношение $R(E)$ на $V$ следующим образом: $v_1R(E)v_2$ тогда и только тогда, когда $[v_1, v_2]\in E$. Отношение $R(E)$ нерефлексивно, так как $vR(E)v$ тогда и только тогда, когда $[v, v]\in E$, но $[v, v]\notin E$, поскольку $(v, v)\notin V^2_{-}$. $R(E)$ симметрично для $v_1R(E)v_2$ тогда и только тогда, когда $[v_1, v_2]\in E$, однако $[v_1, v_2]=\{(v_1, v_2), (v_2, v_1)\}=[v_2, v_1]$. Следовательно, $v_1R(E)v_2$ тогда и только тогда, когда $v_2R(E)v_1$.

б) Если $R$ — нерефлексивное симметричное отношение на $V$, то $R\subset V^2_{-}$.

Нерефлексивность $R$ означает, что $(v, v)\notin R$ для любого $v\in V$, поэтому $R\subset V^2_{-}$.

Симметричность $R$ означает, что $(v_1, v_2)\in R$ тогда и только тогда, когда $(v_2, v_1)\in R$. Определим $E$ формулой $E=R/\sim$, тогда $G=(V, E)$ есть искомый граф.

Графы могут быть представлены матрицами с булевыми элементами. Многие из свойств графов могут быть определены из их матричных представлений путем алгебраических преобразований. Это станет понятным из последующего изложения.

Определение. *Матрица смежности* $A\in\mathcal{M}(n, \mathbf{B})$ графа $G=(V, E)$, где $|V|=n$, определяется следующим образом:

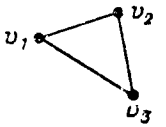
$$A_{ij}=\begin{cases}1, & \text{если } [v_i, v_j]\in E,\\ 0 & \text{в противном случае.}\end{cases}$$

Говорят, что вершины $v_i$ и $v_j$ являются *смежными*, если $A_{ij}=1$. Ясно, что $A_{ii}=0$ $(i=1, \ldots, n)$ и $A=A^T$. Таким образом, $A$ симметрична, и в обозначениях § 1 гл. 6 $A=A(R(E))$. //

Изображение графа $G=(V, E)$ получается путем расположения различных точек на $\mathbb{R}^2$ для каждой $v \in V$, причем, если $[v, w] \in E$, мы проводим линию, соединяющую вершины $v$ и $w$.

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

Матрица смежности




Изображение

Рис. 7.1

Пример 1.1. Пусть

1. $V=\{v_1, v_2, v_3\}$, $E=\{[v_1, v_2], [v_2, v_3], [v_1, v_3]\}$, $|V|=3$, $|E|=3$.

Этот граф изображен на рис. 7.1.

2. $V=\{v_1, v_2, v_3, v_4, v_5\}$,

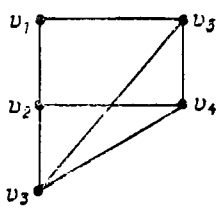
$$E=\{[v_1, v_2], [v_1, v_5], [v_2, v_3], [v_2, v_4], [v_3, v_5], [v_3, v_4], [v_4, v_5]\},$$

$$|V|=5, \quad |E|=7.$$

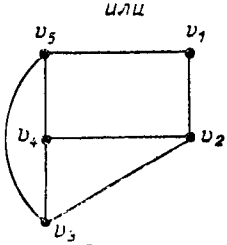
Этот граф изображен на рис. 7.2.

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Матрица смежности




или

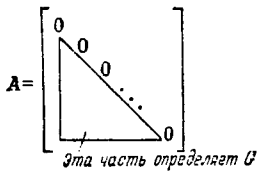



Изображение

Рис. 7.2

Графы являются скорее «топологическими», чем «геометрическими» объектами, т. е. они выражают больше от-

ношения между вершинами, чем расположение вершин и ребер в пространстве. Таким образом, граф может быть изображен бесконечным количеством разных, но «эквивалентных» способов. Однако изображения графов могут вводить в заблуждение. Например, из пересечения двух ребер на рисунке не следует, что точка пересечения является вершиной (см. первую диаграмму на рис. 7.2). Ясно, что нижней (верхней) треугольной части матрицы смежности достаточно, чтобы определить граф.




Читатель уже знаком с понятиями подструктуры и изоморфизма или же с эквивалентностью алгебраических систем. Дадим следующие определения.

Определение. Говорят, что граф $H=(V_1, E_1)$ является *подграфом* графа $G=(V, E)$, если $V_1 \subseteq V$ и $E_1 \subseteq E$. Если $V_1 = V$, то говорят, что $H$ является *остовным подграфом* $G$. Если $V_1$ — непустое подмножество вершин графа $(V, E)$, то подграф $(V_1, E_1)$, порожденный $V_1$, определяют как

$$[v, w] \in E_1 \Leftrightarrow v, w \in V_1 \text{ и } [v, w] \in E. \ //$$

Определение.

а) Пусть $G_1=(V_1, E_1)$ и $G_2=(V_2, E_2)$ — графы. Будем говорить, что $G_1$ и $G_2$ *эквивалентны*, если существует биекция $f$: $V_1 \to V_2$ такая, что

$$vR(E_1)w \Rightarrow f(v)R(E_2)f(w).$$

б) Пусть $G=(V, E)$ — произвольный граф. Определим отображение

$$\delta: V \to \mathbf{N} \cup \{0\}$$

следующим образом: величина $\delta(v)$ равна числу ребер, содержащих вершину $x \in V$. Назовем $\delta(v)$ *степенью* вершины $v$.

Следующее предложение выражает два простых, но важных факта о свойствах графов.

Предложение.

а) $\sum_{v \in V} \delta(v) = 2|E|$.

б) *В любом графе число вершин нечетной степени четно.*

Доказательство. Каждое ребро дважды входит в сумму, откуда и следует утверждение.

в) Пусть $V_e \subseteq V$ — множество вершин четной степени, а $V_0 \subseteq V$ — множество вершин нечетной степени. Заметим, что

$$V = V_e \cup V_0 \text{ и } V_e \cap V_0 = \varnothing;$$

следовательно,

$$\sum_{v \in V} \delta(v) = \sum_{v \in V_e} \delta(v) + \sum_{v \in V_0} \delta(v),$$

$$2|E| = 2k + \sum_{v \in V_0} \delta(v).$$

(Ясно, что $\sum_{v \in V_e} \delta(v) = 2k$, где $k$ — некоторое целое.) Таким образом,

$$\sum_{v \in V_0} \delta(v) = 2(|E| - k),$$

т. е. четно, однако каждое $\delta(v)$ в левой части нечетно, поэтому $|V_0|$ четно. //

Во многих приложениях теории графов о топологии графа имеется дополнительная информация, относящаяся к $V$, или к $E$, или к обоим множествам одновременно. Чтобы конкретизировать вышесказанное, определим понятие помеченного графа и дадим несколько примеров.

Определение.

1) Пусть $S_V$ и $S_E$ — множества меток. *Пометкой* или *распределением меток графа* $G = (V, E)$ называется пара функций

$f$: $V \to S_V$ — распределение меток вершин,

$g$: $E \to S_E$ — распределение меток ребер.

2) Пусть граф $G = (V, E)$ помечен с помощью функций $f$ и $g$, а $G_1 = (V_1, E_1)$ помечен с помощью $f_1$ и $g_1$. Графы $G$ и $G_1$ называются *эквивалентно помеченными*, если существует биекция $h$: $V \to V_1$, такая что

а) $G$ и $G_1$ эквивалентны как непомеченные графы;

б) $f(v) = f_1(h(v))$ для всех $v \in V$, поэтому соответствующие вершины имеют одну и ту же пометку;

в) $g([v, w]) = g_1([h(v), h(w)])$ для всех $v, w \in V$, т. е. соответствующие ребра имеют одну и ту же пометку.

Часто бывают помеченными только ребра или же только вершины. Вышесказанное применимо и в этом случае. Тогда

$$\left.\begin{aligned} &f\colon V \to S_V,\\ &g = \text{const},\end{aligned}\right\}\ \text{помечены только вершины;}$$

$$\left.\begin{aligned} &f = \text{const},\\ &g\colon E \to S_E,\end{aligned}\right\}\ \text{помечены только ребра. //}$$

Ребра или вершины (или те и другие вместе) помеченного графа несут информацию, которая дополняет или заменяет обычную идентификацию с помощью имен.

Пример 1.2.

1. Пусть

$$V = \{v_1, v_2, v_3, v_4\}, \quad E = \{[v_1, v_3], [v_2, v_3], [v_3, v_4]\},$$

$$f\colon V \to \{\text{города Великобритании}\}, \quad g\colon E \to \mathbf{N},$$

$f(v_1)$ — Лондон, $g([v_1, v_3]) = 105$,
$f(v_2)$ — Кардифф, $g([v_2, v_3]) = 196$,
$f(v_3)$ — Бирмингем, $g([v_3, v_4]) = 292$,
$f(v_4)$ — Эдинбург.

Этот граф изображен на рис. 7.3.

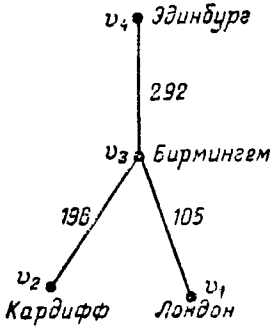



Рис. 7.3

2. Пусть графы

$$G_1 = (\{v_1, v_2, v_3\}, \{[v_1, v_2], [v_1, v_3], [v_2, v_3]\},$$

$$G_2 = (\{w_1, w_2, w_3\}, \{[w_1, w_2], [w_1, w_3], [w_2, w_3]\})$$

помечены так же, как указано на рис. 7.4; $G_1$ и $G_2$ являются эквивалентно помеченными графами (вершины не помечены). //

Упражнение 7.1.

1. Построить доказательство первого предложения этого параграфа.

2. Изобразить графы, представленные следующими матрицами смежности:

а) $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$; б) $\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$.

3. Определить матрицы смежности графов, представленных на рис. 7.5.

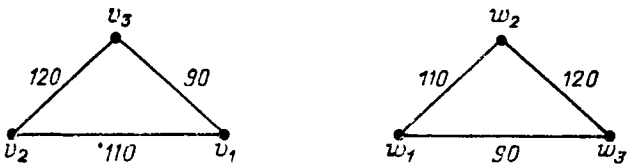



Рис. 7.4

4. Начертить подграф, порожденный вершинами $\{v_2, v_3, v_4, v_6\}$ графа на рис. 7.5, *a*.

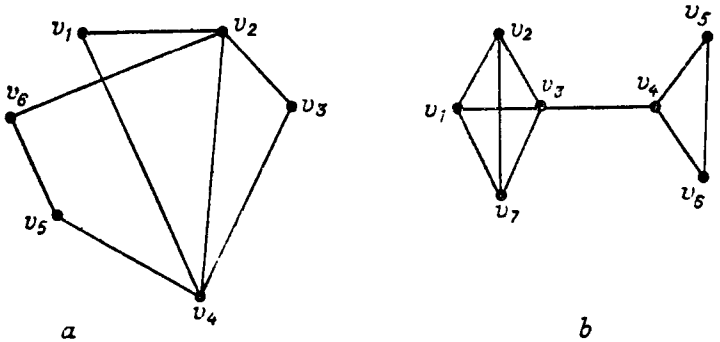



Рис. 7.5

5. Пусть $G=(V, E)$ — граф и $|V|=n$. Какое может быть максимально возможное значение $|E|$?

6. Сколько существует различных графов, имеющих $n$ вершин? Остальные задачи этого параграфа требуют введения некоторых дополнительных понятий.

Определение.

а) Граф $G=(V, E)$ называется *полным*, если для всех $v_1, v_2 \in V$ имеем $[v_1, v_2] \in E$. Полный граф с вершинами обозначается через $K_n$.

б) Граф $G=(V, E)$ называется *двудольным*, если существует разбиение $V=\{V_1, V_2\}$ такое, что никакие две вершины из $V_1$ или из $V_2$ не являются смежными. Двудольный граф называется *полным*, если для любой пары $v_1 \in V_1$ и $v_2 \in V_2$ имеем $[v_1, v_2] \in E$. Если $|V_1|=m$ и $|V_2|=n$, то полный двудольный граф $(V, E)$ обозначается через $K_{m,n}$. //

7. Изобразить граф $K_5$.
8. Построить пример двудольного графа.
9. а) Изобразить граф $K_{3,3}$.
   б) Сколько ребер имеет граф $K_{m,n}$?

## § 2. Маршруты, циклы и связность

Обратим сейчас внимание на понятие маршрута в графе. Значительная часть теории графов и ее приложений занимается вопросами существования и свойств маршрутов. Некоторые важные свойства вытекают из следующих определений.

Определение.

а) Пусть $G=(V, E)$ — граф. *Маршрутом длины* $k$ в графе $G$ из $v$ в $w$ называется последовательность $\langle v_0, v_1, \ldots, v_k\rangle$ вершин (необязательно различных) $v_i \in V$ таких, что $v_0=v$, $v_k=w$, а $[v_{i-1}, v_i]\in E$ для всех $i=1, \ldots, k$. Маршрут называется *замкнутым*, если $v_0=v_k$. Маршрут называется *цепью*, если все его вершины различны. Замкнутая цепь называется *циклом*. Цикл называется *простым циклом*, если только $v_0=v_k$, а остальные $v_i$ различны.

б) Если существует маршрут из $v$ в $w$, $v, w\in V$, то говорят, что $w$ *достижима* из $v$.

в) Граф без циклов называется *ациклическим*.

Циклы и длины циклов были определены для случая подстановок в § 4 гл. 3. Заметим, что понятие замкнутости, в сущности, соответствует своему названию.

Определение.

а) Граф $G=(V, E)$ называется *связным*, если каждая пара различных вершин может быть соединена маршрутом.

б) *Деревом* называется связный ациклический граф.

в) *Корневым деревом* называется дерево с выделенной вершиной, называемой *корнем*.

г) *Остовным деревом* для $G=(V, E)$ называется остовный подграф, являющийся деревом.

В § 1 мы отметили, что вычисления с матрицей смежности обнаруживают важную информацию о природе графа. Следующие результаты являются примерами внутренних связей между алгеброй и топологией в теории графов.

В приведенной ниже теореме и ее следствиях степени $A^k$ вычисляются в $\mathcal{M}(n, \mathrm{Z})$, а не в $\mathcal{M}(n, \mathrm{B})$. Следовательно, в $A^k$ могут возникать числа, большие чем 1.

Теорема. *Пусть $A$ — матрица смежности графа $G=(V, E)$ и $|V|=n$. Тогда $(A^k)_{ij}$ есть число маршрутов длины $k$ от $v_i$ к $v_j$.*

Доказательство. Будем использовать индукцию по $k$. Для $k=1$ маршрут длины 1 как раз является ребром $G$. Следовательно, результат теоремы при $k=1$ вытекает из определения $A$. Пусть

$$(A^{k-1})_{ij}=\alpha_{ij}, \quad A_{ij}=a_{ij},$$

тогда

$$(A^k)_{ij}=(A^{k-1}A)_{ij}=\sum_{q=1}^{n}\alpha_{iq}a_{qj}.$$

Пусть результат имеет место для $k-1$. Тогда, если $\alpha_{iq}$ — элементы матрицы $A^{k-1}$, то $\alpha_{iq}$ — число маршрутов длины $k-1$ от $v_i$ к $v_q$; по определению $a_{qj}$ — число маршрутов длины 1 от $v_q$ к $v_j$. Следовательно, $\alpha_{iq}a_{qj}$ — число маршрутов длины $k$ из $v_i$ к $v_j$, где $v_q$ есть предпоследняя вершина маршрута.

Отсюда следует, что

$$\sum_{q=1}^{n}\alpha_{iq}a_{qj}$$

есть число маршрутов длины $k$ от $v_i$ к $v_j$. Это завершает доказательство. //

Следствие.

а) *Маршрут от $v_i$ к $v_j$ $(i\neq j)$ в $G=(V, E)$ существует тогда и только тогда, когда $(i, j)$-й элемент матрицы порядка $n\times n$ $(n=|V|)$:*

$$A+A^2+\ldots+A^{n-1}$$

*не равен нулю.*

б) *Если не использовать условие $i\neq j$, то требуемая матрица имеет вид*

$$A+A^2+\ldots+A^{n-1}+A^n.$$

Доказательство.

а) Пусть $\langle v_i, v_1, \ldots, v_j\rangle$ — маршрут из $v_i$ в $v_j$ в $G$. Если не существует повторяющихся вершин, то (так как $|V|=n$) маршрут содержит не более $n-1$ ребер, и необходимое утверждение следует из теоремы.

Пусть существует повторяющаяся вершина. Тогда маршрут имеет вид

$$\langle v_{i_1}\ldots, \underbrace{v_{r_1}\ldots, v_{r_1}}_{\text{замкнутый маршрут}}\ldots, v_j\rangle,$$

Если мы удалим все такие замкнутые маршруты, то задача сведется к предыдущему случаю, когда вершины не повторялись. Таким образом, в одну сторону требуемый результат получен. В обратную сторону рассуждения очевидны.

б) Если разрешается $i = j$, то существование маршрута из $v_i$ в $v_j$ влечет то, что существует последовательность $\langle v_i, v_1, \ldots, v_j\rangle$. Если не существует повторяющихся вершин (за исключением, возможно, случая $v_i = v_j$), тогда маршруты состоят из более чем $n+1$ вершин (не более $n$ ребер). Следовательно, $(i, j)$-й элемент матрицы $\sum\limits_{k=1}^{n} A^k$ не равен нулю. Тогда при $|V| = n$ отсюда следует

$$A(R^+(E)) = A(R(E)) \vee A(R^2(E)) \vee \cdots \vee A(R^n(E)) = \bigvee_{k=1}^{n} A(R^k(E)),$$

$$A(R^*(E)) = I \vee A(R(E)) \vee \ldots \vee A(R^{n-1}(E)) = \bigvee_{k=1}^{n-1} A(R^k(E)). \text{ //}$$

Напомним, что для произвольного бинарного отношения $R$ величина $R^+$ определялась как

$$R^+ = \bigcup_{k=1}^{\infty} R^k,$$

и если $R \subseteq V \times V$ при $|V| = n$, то отсюда следует (с учетом § 1 гл. 6), что

$$A(R^+) = A^+(R) = \bigvee_{k=1}^{n} A(R^k).$$

Аналогично

$$A(R^*) = A^*(R) = \bigvee_{k=0}^{n-1} A(R^k).$$

В этом параграфе для упрощения обозначений будем теперь обозначать $A(R^k(E))$ через $A(R^k)$, $A(R^+(E))$ через $A(R^+)$, а $A(R^*(E))$ через $A(R^*)$. Алгоритм Уоршолла требует $4n^3$ операций для определения $A(R^+)$, тогда как при помощи приведенных выше соотношений требуется $4n^4 - 7n^3$ операций. Можно получить и другие,

еще более эффективные алгоритмы для больших значений $n$.

Матрицу $C = A(R^*)$ называют *матрицей связи, связности* или *достижимости* графа $G = (V, E)$. Маршрут из $v_i$ к $v_j$ $(i \neq j)$ существует в $G$ тогда и только тогда, когда $(i, j)$-й элемент из $C$ равен 1. Граф $G$ является связным тогда и только тогда, когда $C_{ij} = 1$ для всех $1 \leqslant i, j \leqslant n$.

Важные свойства отношения $R^*$ могут быть сформулированы следующим образом.

**Предложение.** *$R^*$ — отношение эквивалентности на $V$.*

**Доказательство.** Так как по определению $R^*$ является рефлексивным замыканием $R$, то необходимо только проверить симметричность $R^*$. Выполнение $vR^*w$ влечет существование маршрута $\langle v, v_1, \ldots, v_k, w\rangle$ от $v$ к $w$ в $G$, т. е.

$$[v, v_1] \in E, \quad [v_1, v_2] \in E, \ \ldots, \ [v_k, w] \in E.$$

Следовательно,

$$[w, v_k] \in E, \quad [v_k, v_{k-1}] \in E, \ \ldots, \ [v_2, v_1] \in E, \quad [v_1, v] \in E.$$

Таким образом,

$$\langle w, v_k, v_{k-1}, \ldots, v_2, v_1, v\rangle$$

есть маршрут из $w$ в $v$ в $G$, откуда следует $wR^*v$. //

Отношение $R^*$ определяет важный класс подграфов, который сейчас будет определен. Будут также даны некоторые сопутствующие понятия; они будут важны в дальнейшем, когда будут обсуждаться «пересечения» графа.

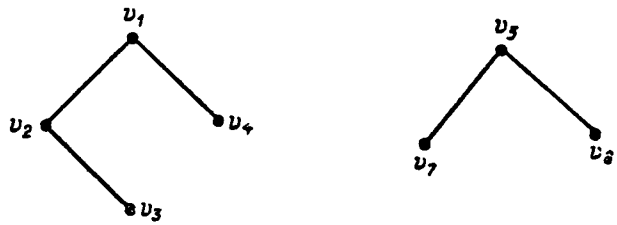
**Определение.** Пусть $\{V_i\colon 1 \leqslant i \leqslant p\}$ — разбиение графа, определяемое отношением $R^*$. Тогда говорят, что $p$ — *число связности* $G$. Подграфы $(V_i, E_i)$, порожденные классами эквивалентности, называют *компонентами связности* графа $G$.

*Лесом* называется граф, в котором каждая связная компонента является деревом. *Остовный лес* для графа $G = (V, E)$ — это совокупность вершин разъединенных деревьев $T_i = (V_i, E_i)$ таких, что $V = \bigcup_i V_i$ и $E_i \subset E$ для всех $i$. (Разъединенность вершин означает, что $V_i \cap V_j = \varnothing$ при $i \neq j$.)

Рисунок 7.6 иллюстрирует вышеупомянутые понятия для графа при $p = 2$.

Упражнение 7.2.

1. Пусть $G=(V, E)$, где $V=\{v_1, v_2, v_3, v_4\}$ и $E=\{[v_1, v_2], [v_1, v_3], [v_2, v_4], [v_3, v_4]\}$.




*Остовный лес для графа*

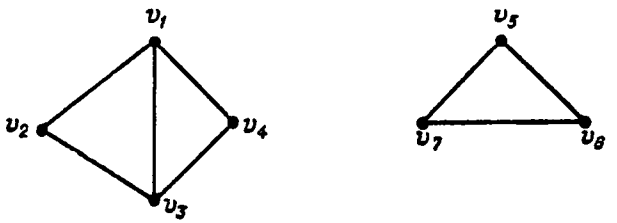



Рис. 7.6

Используя матрицу смежности $G$, определить:

а) число маршрутов длины 2 из $v_3$ в $v_2$;

б) число маршрутов длины 3 из $v_1$ в $v_2$;

в) является ли $G$ связным.

2. Изобразить остовные деревья для графов из упражнения 7.1, 3.

3. Дать матричную характеристику ациклично сти в графе.

## § 3. Планарные графы

Исторически во многих работах по теории графов рассматривают специальный класс графов, который может быть аккуратно представлен рисунком на плоскости $\mathbf{R}^2$. В этом параграфе мы рассмотрим три важных результата, касающихся таких графов.

### 3.1. Теоремы Эйлера и Куратовского.

Определение.

а) Граф $G$ называется *планарным*, если он может быть изображен на плоскости так, что его ребра не

пересекаются*). Такой рисунок называют *картой* $G$.

б) Карта $G$ называется *связной*, если $G$ связен. //

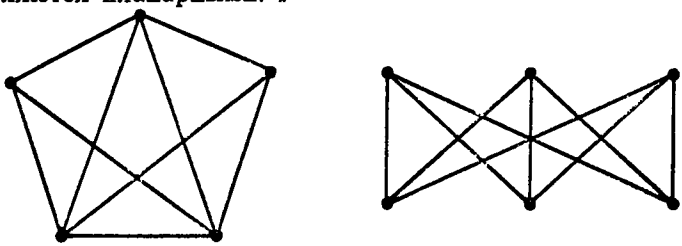
Пример 3.1.

1. $G=(\{v_1, v_2, v_3, v_4\}, \{[v_1, v_2], [v_1, v_3], [v_1, v_4], [v_2, v_3], [v_2, v_4], [v_3, v_4]\})$. Изображение и соответствующая карта $G$ показаны на рис. 7.7; следовательно, $G$ — планарный граф.


*Изображение G* *Карта G*

Рис. 7.7

2. Графы, изображенные на рис. 7.8, *a* и *b* обычно обозначают через $K_5$ и $K_{3,3}$ соответственно. Позднее мы рассмотрим доказательство того факта, что эти графы не являются планарными. //



*a* *b*

Рис. 7.8

Карта делит $\mathbf{R}^2$ на «области»; проиллюстрируем это в следующем примере.

Пример 3.2. Рассмотрим карту, изображенную на рис. 7.9. Она делит $\mathbf{R}^2$ на четыре области; $r_1$, $r_2$ и $r_3$ являются ограниченными областями, а $r_4$ — «неограниченная» область.

В оставшихся параграфах будем обозначать множество областей данной карты через $\mathcal{R}$.

*) Такой граф, изображенный на плоскости, называется *плоским графом*.— *Примеч. ред.*

Теорема Эйлера. *Для произвольной связной карты*

$$|V| - |E| + |\mathscr{R}| = 2.$$

Доказательство. Пусть $|V| = 1$. Тогда очевидно, что $|E| = 0$ и $|\mathscr{R}| = 1$. Следовательно, формула справедлива для $|V| = 1$. Рассмотрим два возможных способа расширения данной карты.

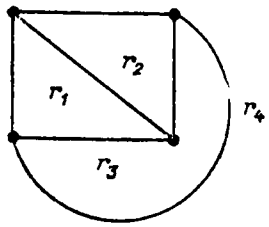



Рис. 7.9

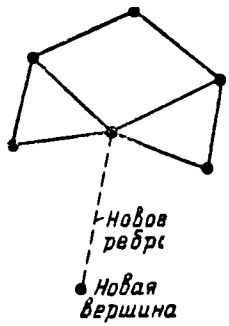



Рис. 7.10

1) Добавим новую вершину и присоединим ее к существующей карте.

2) Соединим две существующие вершины.

Покажем, что значение $|V| - |E| + |\mathscr{R}|$ инвариантно относительно обоих способов расширения.

В первом случае добавим новую вершину так, как, например, это сделано на рис. 7.10. Этот процесс увеличивает $|V|$ на 1 и $|E|$ на 1, однако $|\mathscr{R}|$ остается тем же самым, вследствие чего значение $|V| - |E| + |\mathscr{R}|$ не изменяется.

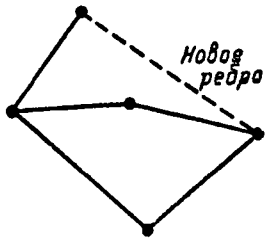



Рис. 7.11

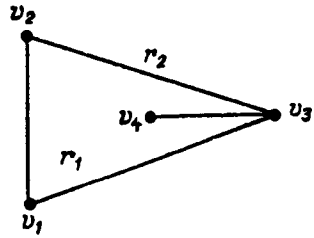



Рис. 7.12

Во втором случае соединим две вершины, как, например, на рис. 7.11. Тогда $|V|$ остается тем же самым, $|E|$

возрастает на 1, а $|\mathcal{R}|$ уменьшается на 1; следовательно, значение $|V| - |E| + |\mathcal{R}|$ опять остается неизменным.

Все карты могут быть получены из случая $|V| = 1$ путем выполнения 1) и (или) 2); если необходимо, то процедура повторяется. Следовательно, так как $|V| - |E| + |\mathcal{R}| = 2$ при $|V| = 1$, а значение $|V| - |E| + |\mathcal{R}|$ остается инвариантным при выполнении 1) и 2), то мы имеем $|V| - |E| + |\mathcal{R}| = 2$ для всех связных карт.

Очевидно, что каждая область карты ограничена замкнутым маршрутом. Ниже приведены два простых результата (один касается ограничивающих замкнутых маршрутов). Этих результатов оказывается достаточно для доказательства того, что $K_5$ и $K_{3,3}$ не являются планарными.

Определение. Пусть $G = (V, E)$ — планарный граф. Для карты $G$ определим *степень* $\Delta_r$ области $r$ как длину замкнутого маршрута, ограничивающего $r$. //

Пример 3.3. Для карты, изображенной на рис. 7.12, имеем $\langle v_1, v_3, v_4, v_3, v_2, v_1\rangle$ — граничный замкнутый маршрут для $r_1$, а $\langle v_1, v_3, v_2, v_1\rangle$ — граничный замкнутый маршрут для $r_2$. Следовательно, $\Delta_{r_1} = 5$, $\Delta_{r_2} = 3$. //

Предложение. *Пусть* $G = (V, E)$ — *планарный граф. Тогда*

а) $\sum\limits_{r \in R} \Delta_r = 2|E|$ *для любой карты* $G$;

б) *если* $|V| \geqslant 3$, *то* $|E| \leqslant 3|V| - 6$.

Доказательство оставляем читателю в качестве упражнения.

Предложение. *Графы* $K_5$ *и* $K_{3,3}$ *не являются планарными.*

Доказательство.

Докажем, что $K_5$ не является планарным. Если $K_5$ — планарный граф, то из предыдущего результата следует, что $|E| \leqslant 3|V| - 6$. Тогда для графа $K_5$ с $|E| = 10$ и $|V| = 5$ имеем $10 \leqslant 15 - 6 = 9$, что неверно. Таким образом, предположение, что граф $K_5$ планарный, неверно.

Докажем теперь, что $K_{3,3}$ не является планарным. Предполагая противное, имеем $\sum\limits_{r \in \mathcal{R}} \Delta_r = 2|E|$. Однако в $K_{3,3}$ никакие три вершины не связаны одна с другой; следовательно, $\Delta_r \geqslant 4$ для всех $r \in \mathcal{R}$. Из формулы Эйлера $|\mathcal{R}| = 2 + |E| - |V| = 2 + 9 - 6 = 5$; следовательно, $\sum\limits_{r \in \mathcal{R}} \Delta_r \geqslant 5 * 4 = 20$. Таким образом, $2|E| \geqslant 20$, откуда

$|E| \geqslant 10$. Поскольку последнее неравенство неверно, то граф $K_{3,3}$ не является планарным. //

Графы $K_5$ и $K_{3,3}$ интересны тем, что они являются существенно «единственными» непланарными графами. Все другие непланарные графы имеют подграфы «подобные» или $K_5$, или $K_{3,3}$. Перед тем как уточнить наше утверждение, необходимо ввести два определения.

Определение.

а) Пусть $G=(V, E)$ — граф. *Элементарное стягивание* $G$ образуется путем удаления ребра $[v_i, v_j]$ из $E$, замены каждого $v_i$ и $v_j$ в $E$ новым символом $w$, удаления $v_i$ и $v_j$ из $V$ и добавления $w$ к $V$. Графически элементарное стягивание $G$ получают путем слияния двух смежных вершин после удаления ребра между ними и обозначения «составной» вершины через $w$.

б) Граф $G$ называется *стягиваемым* к графу $G'$, если $G'$ может быть получен из $G$ путем последовательности элементарных стягиваний.

Пример 3.4. На рис. 7.13 изображены графы $G$ и $G'$, при этом $G$ стягивается к $G'$. //

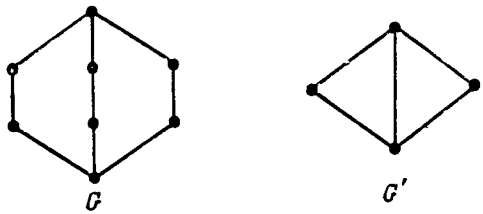



Рис. 7.13

Теорема (Куратовский). *Граф является планарным тогда и только тогда, когда он не содержит подграфов, стягиваемых к* $K_5$ *или* $K_{3,3}$.

Доказательство этой теоремы лежит за пределами целей книги и поэтому опущено. //

Из теоремы Куратовского заключаем, что $K_5$ и $K_{3,3}$ являются существенно единственными не планарными графами. Алгоритмы, основанные на этой теореме, были придуманы для того, чтобы определить, является ли данный граф планарным или нет.

**3.2. Раскраска карт и графов.**

Определение.

а) *Раскраской* $G=(V, E)$ называется задание цвета вершинам $G$ так, что если $[v, w] \in E$, то $v$ и $w$ имеют различные цвета.

б) Хроматическим числом $\chi(G)$ графа $G$ называется минимальное число цветов, требующееся для раскраски $G$. //

Теорема. $\chi(G) \leqslant 4$ *для всех планарных графов* $G$. //

Эта теорема была впервые «доказана» в 1976 г. проверкой на компьютере всех возможных случаев*).

Определение. Пусть $M$ — карта. Определим карту $M'$, называемую *двойственной* к $M$, следующим образом: выберем внутреннюю точку в каждой области $M$; если две области имеют общее ребро, то проведем дугу, связывающую выбранные внутренние точки в этих областях. Этот процесс определяет $M'$. //

Пример 3.5. На рис. 7.14 изображена карта $M$ вместе с двойственной картой $M'$. //

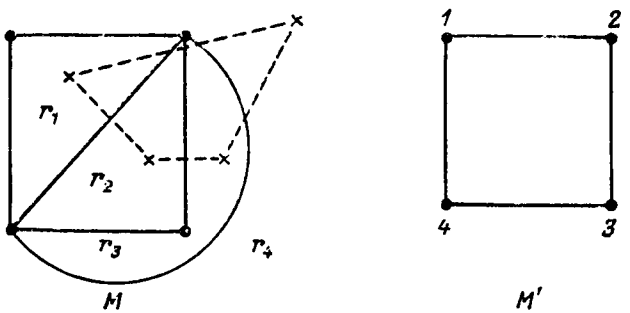



Рис. 7.14

Раскраска $M'$ соответствует раскраске областей $M$ так, что области, имеющие общее ребро, имеют разные цвета. Следовательно, мы можем переформулировать теорему о раскрашивании в четыре цвета следующим образом.

Теорема. *Если области карты* $M$ *необходимо раскрасить таким образом, чтобы смежные области имели различные цвета, то для этого требуется не более четырех цветов.* //

Упражнение 7.3.

1. Пусть $T = (V, E)$ — дерево с $|V| = n$. Доказать, что $|E| = n - 1$.

2. Проверить справедливость формулы Эйлера для графов из упражнения 7.1, 1.

---

*) До сих пор нет уверенности в том, что эта теорема на самом деле доказана.— *Примеч. ред.*

3. Пусть $G=(V, E)$ — планарный граф и $\Delta_r$ обозначает степень области $r$ в карте $G$. Доказать, что

$$\sum_{r \in \mathcal{R}} \Delta_r = 2|E|.$$

4. Пусть $G=(V, E)$ — связный планарный граф и $|V| \geqslant 3$. Доказать, что

$$|E| \leqslant 3|V| - 6.$$

5. Привести примеры, показывающие, что приведенное в задаче 4 этого упражнения неравенство неверно при $|V| < 3$.

6. Определить граф $G$, для которого $\chi(G)=4$.

7. Пусть $T$ — дерево. Что является значением $\chi(T)$?

## § 4. Структуры данных для представления графа

Матрица смежности предполагает очевидный метод представления графа в машине — на языке высокого уровня мы можем использовать массив, чтобы хранить элементы матрицы. Симметричность и нулевая диагональ сокращают необходимый объем памяти; для графа с $n$ вершинами требуется $\frac{1}{2} n(n-1)$ ячеек. Обычно многие элементы матрицы равны нулю и поэтому большая часть используемой памяти является лишней, однако, несмотря на это, иногда матрица смежности является наиболее удобным представлением графа. Однако для многих задач предпочтительным является представление в виде списка смежности. В этом случае мы свяжем список связей $L_v$ с каждой вершиной $v \in V$; $L_v$ является списком вершин, смежных с $v$.

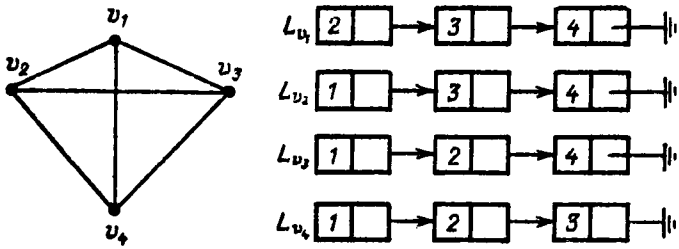



Рис. 7.15

Пример 4.1. На рис. 7.15 даны граф и списки, которые могут быть использованы для его представления. //

Один из путей применения этих списков — это использование массивов

$$E(j, k),\ 1 \leqslant k \leqslant 2, \quad \text{и} \quad P(i),\ 1 \leqslant i \leqslant n,$$

где $E(j, 1)$ хранит номера вершин, $E(j, 2)$ хранит связь, а $P(i)$ — точки начала списка $L_{v_i}$ в $E$.

Пример 4.2. Для графа из примера 4.1 мы можем иметь ситуацию, изображенную на табл. 7.1.

Таблица 7.1

| $i$ | $E(i, 1)$ | $E(i, 2)$ | $P(i)$ | $i$ | $E(i, 1)$ | $E(i, 2)$ | $P(i)$ |
|---|---|---|---|---|---|---|---|
| 1 | | | 5 | 13 | 4 | 0 | |
| 2 | 1 | 6 | 2 | . | . | . | |
| 3 | | | 11 | . | . | . | |
| 4 | | | 35 | . | . | . | |
| 5 | 2 | 9 | | 19 | 2 | 12 | |
| 6 | 3 | 10 | | . | . | . | |
| 7 | | | | . | . | . | |
| 8 | | | | . | . | . | |
| 9 | 3 | 13 | | 35 | 1 | 36 | |
| 10 | 4 | 0 | | 36 | 2 | 37 | |
| 11 | 1 | 19 | | 37 | 3 | 0 | |
| 12 | 4 | 0 | | | | | |

Если $v_i$ не имеет смежных вершин, то установим $P(i) = 0$, в противном случае $E(P(i), 1)$ является смежной с $v_i$ и $[i, E(P(i), 1)]$ является ребром графа. Аналогично, если $E(P(i), 2) \neq 0$, тогда $E(E(P(i), 2), 1)$ является смежной с $v_i$. Фрагмент программы, которая дает список смежных вершин к каждой вершине, выглядит следующим образом:

```
for 1 ⩽ i ⩽ n do
    begin
      write 'вершины смежные с вершиной' , i
      k ← P(i)
        while k ≠ 0 do
            begin
                write E(k, 1)
                k ← E(k, 2)
            end
    end
```

Аналогичная программа может быть использована для создания матрицы смежности $M$ графа из представления

с помощью списка связей $E$, $P$ следующим образом:

```
for 1 ≤ i, j ≤ n do M(i, j) ← 0
for 1 ≤ i ≤ n do
    begin
      k ← P(i)
        while k ≠ 0 do
          begin
            M(i, E(k, 1)) ← 1
            k ← E(k, 2)
          end
    end
```

Чтобы создать структуры $E$ и $P$ из $M$, можно использовать следующую программу:

```
for 1 ≤ i ≤ n do P(i) ← i
for 1 ≤ j ≤ n do E(i, j) ← 0
for 1 ≤ i ≤ n do
    begin
      ifree ← i
      for 1 ≤ j ≤ n do
        begin
          if M(i, j) = 1 then do
              begin
                E (ifree, 1) ← j
                if j ≠ n do
                    begin
                      E (ifree, 2) ← ifree + n
                      ifree ← ifree + n
                    end
                end
        end
    end
```

Возможно другое представление графов с помощью списка связей. Выбор представления во многом зависит от используемых алгоритмов.

Представление $L_v$ вершин, смежных с $v$, в виде списка связей определяет «порядок» ребер, выходящих из $v$. Рассмотрим это последнее утверждение и списки связей, данные в начале этого параграфа, по отношению к графу. Ребра упорядочены, как показано на рис. 7.16. Граф с упорядочиванием ребер такого сорта называют *упорядоченным графом*. Дадим формальное определение,

Определение. Множество $V=\{v_1, \ldots, v_n\}$ вершин вместе с множеством $\{L_{v_1}, L_{v_2}, \ldots, L_{v_n}\}$ упорядоченных списков упорядоченных пар вершин называют *упорядоченным графом.*

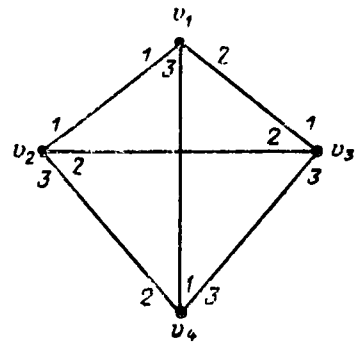


Рис. 7.16

Необходимо наложить некоторые условия на списки $L_v$, чтобы рассматриваемые структуры были графами. Эти условия следующие:

а) $(v, v) \notin L_v$ для любого $v \in V$;

б) $(w, u) \in L_w \Rightarrow (u, w) \in L_u$. //

Граф, изображенный на рис. 7.16, может быть записан в терминах упорядоченного графа следующим образом:

$$(\{v_1, v_2, v_3, v_4\}, \{((v_1, v_2), (v_1, v_3), (v_1, v_4)),$$
$$((v_2, v_1), (v_2, v_3), (v_2, v_4)), ((v_3, v_1), (v_3, v_2), (v_3, v_4)),$$
$$((v_4, v_1), (v_4, v_2), (v_4, v_3))\}).$$

Упорядоченный граф определяет единственный неупорядоченный граф. Обратное утверждение неверно, поскольку в общем случае возможно много способов упорядочения графа. Некоторые из этих упорядочений рассматриваются в соответствии со следующим определением.

Определение. Два упорядоченных графа $G_1$ и $G_2$ называются *эквивалентными*, если существует биекция $f: V_1 \to V_2$ между множествами вершин и биекция сохраняет списковую структуру. Другими словами, если

$$L_v = ((v, w_1), \ldots, (v, w_k))$$

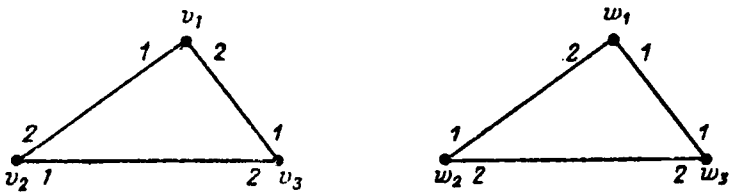
есть список ребер $G_1$, то

$$L_{f(v)} = ((f(v), f(w_1)), \ldots, (f(v), f(w_k)))$$

есть список ребер $G_2$. //

Пример 4.3. Графы, изображенные на рис. 7.17, эквивалентны, однако они не являются эквивалентными как упорядоченные графы. //

Все понятия § 2 (маршруты, замкнутые маршруты, связность, дерево) переносятся очевидным образом на упорядоченные графы.




Ри . 7.17

Упражнение 7.4.

1. Записать алгоритмы данного параграфа на каком-либо языке программирования и проверить их работу на некоторых наборах данных.

## § 5. Обход графа

5.1. Введение. Во многих задачах, включающих графы, требуется обойти граф, т. е. чтобы каждая вершина графа «посещалась» или «обрабатывалась» только один раз. Таким образом, обход графа может быть представлен последовательностью вершин, соответствующих порядку, в котором они обрабатываются.

Если $G=(\{v_1, v_2, \ldots, v_n\}, E)$ и $\sigma$: $\mathbf{N}_n \to \mathbf{N}_n$ — перестановка, то последовательность

$$t = v_{\sigma(1)}, v_{\sigma(2)}, \ldots, v_{\sigma(n)}$$

определяет обход $G$. Так как существует $n!$ различных перестановок $\mathbf{N}_n$, то должно быть $n!$ различных способов обхода графа с $n$ вершинами. Другими словами, существует $n!$ способов полного упорядочения вершин. Вершину $v_{\sigma(1)}$ называют начальной вершиной обхода, определяемого подстановкой $\sigma$.

Мы опишем здесь только два метода обхода графов. Они могут быть полезны в приложениях. Оба метода применяются к упорядоченным графам и позволяют определить перестановку (или полное упорядочение вершин).

5.2. Обход графа по глубине. Пусть $G = (\{v_1, \ldots, v_n\}, \{L_{v_1}, \ldots, L_{v_n}\})$ — упорядоченный граф. Выберем некоторую начальную вершину $v_s$ $(1 \leqslant s \leqslant n)$ и положим $\sigma(1) = s$. Далее вершины последовательности $t$ определяются

следующим образом: $v_{\sigma(2)}$ — первая вершина (смежная с $v_{\sigma(1)}$) из списка смежности $L_{v\sigma(1)}$, $v_{\sigma(3)}$ — первая вершина из $L_{v_{\sigma(2)}}$, которой нет еще в $t$, и т. д., $v_{\sigma(k)}$ — первая вершина из $L_{v_{\sigma(k-1)}}$, которой нет еще в $t$ ($k \geqslant 4$). При этом, если встречается вершина $u$ такая, что все вершины из $L_u$ уже содержатся в $t$, то процесс повторяется из вершины $w \in t$, где $w$ — последняя вершина в $t$ такая, что $L_w$ содержит вершины, не входящие в $t$. Обход заканчивается, когда никакая вершина из $V \backslash t$ не может быть достигнута из вершин последовательности $t$.

Если граф $G$ связный, то описанный выше процесс определяет обход $G$, в противном случае — только одну из компонент графа $G$ (содержащую $v_{\sigma(1)}$). Если граф $G$ не является связным, то для получения полного обхода $G$ необходимо начинать процесс в каждой связной компоненте графа $G$. С помощью этого метода можно определить число связных компонент графа. Для каждого выбора начальной вершины в связном графе получен единственный обход, так что возможны все $n$ обходов по глубине упорядоченного связного графа. Если $G$ имеет связные компоненты $V_i$ ($1 \leqslant i \leqslant p$), где $|V_i| = n_i$, то определены $n_1 * n_2 * \ldots * n_p$ обходов по глубине.

С помощью следующей рекурсивной процедуры можно найти обход по глубине. Здесь $t$ — массив длины $n = |V|$, все значения которого вначале равны 0; $t(v_i)$ устанавливается равным 1 для обозначения того, что вершина $v_i$ обработана.

```
procedure dft(v)
    t(v) ← 1
    process vertex v
    for each w ∈ L_v do if t(w) = 0 then dft(w)
end proc
```

Пример 5.1. Рассмотрим упорядоченный граф, изображенный на рис. 7.18. Первый обход по глубине из начальной вершины $v_1$ определяется следующим образом: $v_1, v_2, v_4, v_8, v_5, v_6, v_3, v_7$. //

**5.3. Обход по ширине.** Пусть $G = (\{v_1, \ldots, v_n\}, \{L_{v_1}, \ldots, L_{v_n}\})$ — упорядоченный граф. Выберем начальную вершину $v_s$ и предположим, что $L_{v_s} = ((v_s, w_1), (v_s, w_2), \ldots, (v_s, w_k))$.

Первые $k+1$ членов $t$ определяются следующим образом: $v_{\sigma(1)} = v_s$, $v_{\sigma(2)} = w_1$, ..., $v_{\sigma(k+1)} = w_k$, а $v_{\sigma(k+1+i)}$

является $t$-й вершиной из $L_{w_1}$, не входящей в $t$. Это исчерпывает $L_{w_1}$, и процесс начинается над $L_{w_2}$ и т. д. Обход прекращается, когда все вершины, достижимые

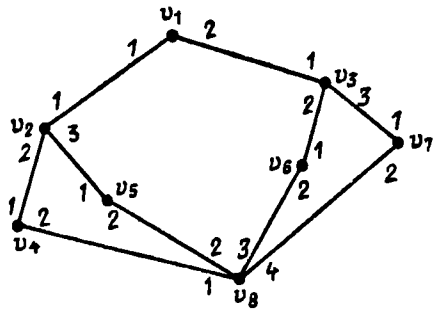



Рис. 7.18

из $v_{\sigma(1)}$, содержатся в $t$. Замечания из п. 5.2 о единственности, связности и числе возможных обходов также применимы к этому обходу. Обход графа по ширине можно найти с помощью следующей процедуры: $t$ играет ту же роль, что и прежде, а $q$ — оставшаяся часть в процедурах сложения и удаления (**add** $q$ и **delete** $q$ соответственно)

```
procedure bft(v)
  t(v) ← 1
  process vertex v
  initialize q with v
    while q ≠ 0 do
      begin
        delete q(v, q)
        for w ∈ L_v do
          begin
            if t(w) = 0 then do
              begin
                add q(w, q)
                t(w) ← 1
                process vertex w
              end
          end
      end
  endproc
```

Пример 5.2. Первый обход по ширине графа в примере 5.1 с начальной вершины $v_1$ задается следующим

образом:

$$v_1,\ v_2,\ v_3,\ v_4,\ v_5,\ v_6,\ v_7,\ v_8.\ //$$

**5.4. Остовные леса обходов по глубине и ширине.** Пусть $G=(\{v_1, \ldots, v_n\}, E)$ — граф, а $t=v_{\sigma(1)}, \ldots, v_{\sigma(n)}$ — обход $G$. Тогда $t$ определяет подмножество $E^t$ из $E$ следующим образом: $[v, w]\in E^t$ тогда и только тогда, когда $[v, w]$ используется при построении обхода. Так как для упорядоченных графов обход $t$ определяет аналогичным путем подсписок $L_v^t$ каждого списка $L_v$, то $L_v^t$ получают из $L_v$ удалением всех пар $(v, w)$, которые не использованы в сечении.

Предложение. *Пусть* $G=(\{v_1, \ldots, v_n\}, \{L_{v_1}, \ldots, L_{v_n}\})$ *— упорядоченный связный граф, а* $t$ *— обход по глубине или ширине графа* $G$. *Тогда*

$$G^t=\left(\{v_1, \ldots, v_n\}, \left\{L_{v_1}^t, \ldots, L_{v_n}^t\right\}\right)$$

*есть упорядоченное остовное дерево для* $G$.

Доказательство. Так как $G$ — связный граф, то подграф $G^t$ также связен и является остовным для $G$. Если $G^t$ содержит замкнутый маршрут, тогда некоторые вершины появляются более одного раза в $t$, но так как $t$ — обход, то это невозможно, и $G^t$ является ациклическим графом. Следовательно, $G^t$ — дерево. //

Следствие. *Каждый связный граф имеет остовное дерево.* //

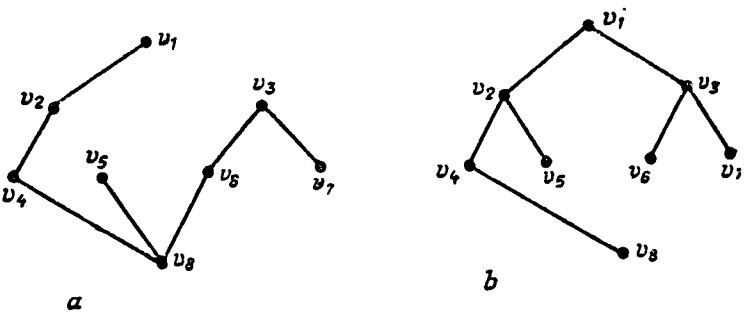



Рис. 7.19

Пример 5.3. Для графа из примера 5.1 остовными деревьями, определенными первичными обходами по глубине и ширине с начальной вершиной $v_1$, будут деревья, изображенные соответственно на рис. 7.19, *a* и 7.19, *b*.

Для графов, не являющихся связными, полные обходы по глубине или ширине определяют остовный лес.

Упражнение 7.5.

1. Пусть $G = (\{v_1, \ldots, v_5\}, \{L_{v_1}, \ldots, L_{v_5}\})$ — упорядоченный граф, определяемый следующими списками:

$L_{v_1} = ((v_1, v_2))$, $L_{v_2} = ((v_2, v_5), (v_2, v_4), (v_2, v_3), (v_2, v_1))$,

$L_{v_3} = ((v_3, v_2))$, $L_{v_4} = ((v_4, v_2), (v_4, v_5))$, $L_{v_5} = ((v_5, v_4), (v_5, v_2))$.

Определить:

а) обход по глубине с начальной вершиной $v_2$;

б) обход по ширине с начальной вершиной $v_4$.

2. Нарисовать остовные деревья, соответствующие обходам упражнения 7.5.1.

3. Пусть матрица смежности графа $G$ имеет блочную структуру

$$\begin{bmatrix} A_1 & & 0 \\ & A_2 & \\ & \ddots & \\ & \ddots & \\ 0 & & A_p \end{bmatrix},$$

где каждое $A_i$ является квадратной матрицей с булевыми элементами, а все остальные элементы равны нулю. Что можно сказать о свойствах $G$?

4. Написать процедуры на каком-нибудь языке программирования для определения обходов по глубине и ширине.

## § 6. Ориентированные графы

**6.1. Введение.** Во многих приложениях теории графов требуется, чтобы ребра графа имели направление. Например, поток данных проходит через программу.

Определение. *Ориентированный граф* (*орграф*) $G$ есть пара $G = (V, E)$, где $V$ — конечное множество вершин, а $E$ — произвольное подмножество $V \times V$. //

Предложение.

а) *Ориентированный граф* $G = (V, E)$ *определяет отношение на* $V$.

б) *Пусть* $V$ *— конечное множество. Тогда отношение на* $V$ *определяет ориентированный граф, у которого множество вершин* — $V$.

Доказательство.

а) Как и в § 1, определим $R(E)$ следующим образом: $vR(E)w$ тогда и только тогда, когда $(v, w) \in E$. Очевидно, что $R(E)$ — отношение.

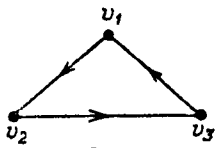
б) Если $R$ — отношение на $V$, то ориентированный граф $G = (V, E)$, определяемый $R$, имеет множество ребер $E$, где $(v, w) \in E$, тогда и только тогда, когда $vRw$. //

Направление ребра обозначают порядком в $V \times V$; например, если $(v, w) \in E$, то говорят, что ребро *выходит* из $v$ и входит в $w$. На диаграмме в этом случае для указания направления используют стрелки.

Пример 6.1. Пусть $V = \{v_1, v_2, v_3\}$, а $E_1 = \{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}$. Тогда матрица смежности и изображение орграфа $G_1 = (V, E_1)$ будут такими, как на рис. 7.20.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

*Матрица смежности*




*Изображение*

Рис. 7.20

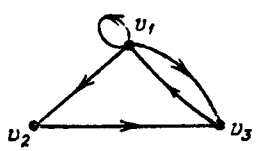
Аналогично на рис. 7.21 приведена матрица смежности и изображение графа $G_2 = (V, E_2)$, где

$$E_2 = \{(v_1, v_1), (v_1, v_2), (v_1, v_3), (v_2, v_3), (v_3, v_1)\}. \text{ //}$$

Поскольку реберное отношение для орграфа не обязательно симметрично или нерефлексивно, то, вообще

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

*Матрица смежности*




*Изображение*

Рис. 7.21

говоря, не обязательно, чтобы $A = A^T$ или $A_{ii} = 0$. Ребра типа $(v, v)$ называют *петлей*. Степень $\delta(v)$ вершины $v \in V$ может быть записана в виде суммы $\delta(v) = \delta^-(v) + \delta^+(v)$, где $\delta^-(v)$ — число ребер, входящих в $v$, а $\delta^+(v)$ — число ребер, выходящих из $v$. Множества

$\{w\colon (w, v)\in E\}$ и $\{w\colon (v, w)\in E\}$ называют соответственно *входящим узлом* и *выходящим узлом* вершины $v\in V$. Понятия эквивалентности и пометки обобщаются на орграфы естественным образом.

6.2. Маршруты и связность в орграфах.

Определение. *Маршрутом длины* $k$ из $v$ в $w$ в орграфе $G=(V, E)$ называется последовательность ребер вида

$$(v, w_1), (w_1, w_2), (w_2, w_3), \ldots, (w_{k-2}, w_{k-1}), (w_{k-1}, w),$$

т. е. вторая вершина каждого ребра совпадает с первой вершиной следующего ребра. //

Часто удобно представлять маршрут последовательностью вершин

$$v, w_1, w_2, \ldots, w_{k-2}, w_{k-1}, w,$$

которые его определяют. Если $v=w$, то маршрут называют *замкнутым маршрутом* или *циклом*. Орграф без циклов называется *ациклическим*.

Теоремы § 2 также справедливы с аналогичными доказательствами для орграфов. Определим связность или матрицу достижимости тем же самым способом. Заметим, однако, что для орграфов отношение $R^*$ не является отношением эквивалентности на $V$ и, следовательно, не осуществляет разбиения $V$.

Пусть $\mathscr{D}$ обозначает множество всех орграфов, а $\mathscr{G}$ — множество всех графов. Мы можем определить отображение $\mathscr{F}\colon \mathscr{D}\to\mathscr{G}$ следующим образом.

Определение. Пусть $G=(V, E)\in\mathscr{D}$. Тогда множество вершин $F(G)\in\mathscr{G}$ совпадает с $V$, а множество ребер $F(G)$ определяется применением следующих операций на $E$:

а) удаляются все петли из $E$;

б) $(v, w)$ заменяются на $[v, w]$ для всех $(v, w)\in E$.

Тогда $F(G)$ является графом, *связанным* с орграфом $G$. //

Для орграфов понятие связности является более содержательным, чем для графов, и имеет отношение к проблеме обхода. Сейчас мы определим три важных типа связности орграфа.

Определение. Если $G=(V, E)$ — орграф, то будем говорить, что:

а) $G$ *слабо связный*, если граф $F(G)$ связный;

б) $G$ *односторонне связный*, если для каждой пары различных вершин $v$, $w\in V$ существует маршрут из $v$ в $w$ или обратно,

в) $G$ *сильно связный*, если для каждой пары различных вершин $v$, $w \in V$ существует маршрут из $v$ в $w$ и обратно. //

Очевидно, что $G$ сильно связный $\Rightarrow G$ односторонне связный $\Rightarrow G$ слабо связный.

Пример 6.2. Из рис. 7.22 мы видим, что орграф:

а) только слабо связный (рис. 7.22, $a$);

б) односторонне связный, но не сильно связный (рис. 7.22, $b$);

в) сильно связный (рис. 7.22, $c$).

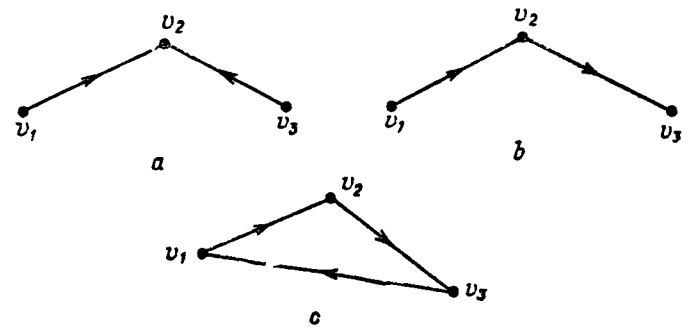



Рис. 7.22

В терминах связности матрицы $C = A(R^*)$ орграф $G$ сильно связный тогда и только тогда, когда $C_{ij} = 1$ для всех $1 \leqslant i, j \leqslant n$; $G$ односторонне связный тогда и только тогда, когда $C_{ij} \vee C_{ji} = 1$ для всех $1 \leqslant i, j \leqslant n$.

Пример 6.3. Рассмотрим орграф, представленный диаграммой на рис. 7.23. Для этого орграфа

$$A(R) = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad A(R^2) = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix},$$

$$A(R^3) = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad A(R^4) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix};$$

поэтому для

$$C = \bigvee_{k=0}^{4} A(R^k) = I \vee A(R) \vee A(R^2) \vee A(R^3) \vee A(R^4)$$

имеем $C_{ij} = 1$ для всех $1 \leqslant i, j \leqslant 5$ и, следовательно, граф является сильно связным. Для более эффективного вычисления $C$ можно использовать алгоритм Уоршолла. //

Если $G = (V, E)$ — орграф, то можно разбить $V$ путем определения отношения эквивалентности $\rho$ следующим

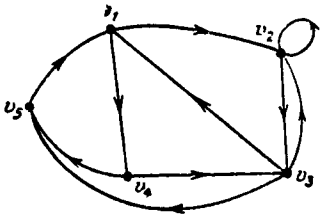



Рис. 7.23

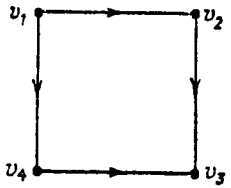



Рис. 7.24

образом: $v \rho w$, если $v = w$ *или* существуют маршруты из $v$ в $w$ и обратно. Если $\{V_i\colon 1 \leqslant i \leqslant p\}$ — разбиение $V$ и $\{E_i\colon 1 \leqslant i \leqslant p$, а $E_i = (V_i \times V_i) \cap E\}$ являются соответствующими множествами ребер, то подграфы $G_i = (V_i, E_i)$ $(1 \leqslant i \leqslant p)$ называются *сильно связными компонентами* $G$.

Очевидно, что $\rho \subseteq R^*$ и $A(\rho)$ может быть определено из $A(R^*)$ как $A(\rho)_{ij} = A(R^*)_{ij} \wedge A(R^*)_{ji}$; граф $G$ сильно связный тогда и только тогда, когда $G$ имеет только одну сильно связную компоненту, т. е. если $p = 1$.

Пример 6.4. Для орграфа на рис. 7.24 имеем

$$A(R^*) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}; \quad A(p) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Таким образом, $G_i = (\{v_i\}, \varnothing)$ $(1 \leqslant i \leqslant 4)$ являются сильно связными компонентами графа. //

Пусть $G = (V, E)$ — ациклический орграф. Вершину $v \in V$ называют *листом*, если $\delta^+(v) = 0$. Если $(v, w) \in E$, то $v$ является *непосредственным предком* $w$, а $w$ — *непосредственным потомком* $v$. Если существует маршрут из $v$ в $w$, то говорят, что $v$ является *предком* $w$, а $w$ — *потомком* $v$.

Эти понятия не имеют смысла для орграфов, имеющих циклы, так как для таких графов вершина может исходить сама из себя.

Пример 6.5. Для ациклического орграфа, изображенного на рис. 7.25, из вершин $v_2$, $v_4$ и $v_5$ ребра не

выходят, $v_1$ является предком $v_5$, $v_5$ является прямым потомком $v_3$ и т. д.

Существует тесная связь между ациклическими орграфами и частично упорядоченными отношениями. В частности, имеет место следующий результат, доказательство которого мы оставляем в качестве упражнения. Заметим, что для сокращения некоторых доказательств,

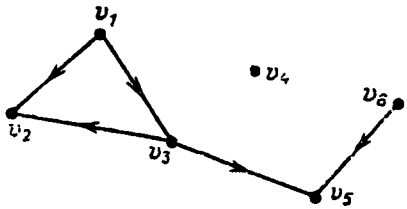



Рис. 7.25

приведенных ниже, частичные порядки будут основаны скорее на отношении $<$, чем на отношении $\leqslant$, и, следовательно, являются транзитивными и нерефлексивными.

Предложение.

а) *Пусть отношение $<$ является частичным отношением порядка на конечном множестве $V$. Тогда, если*

$$E = \{(v, w): v < w\},$$

*то пара $G = (V, E)$ является ациклическим графом.*

б) *Пусть $G = (V, E)$ — ациклический орграф и отношение $<$ определяется следующим образом: $v < w$, если $v$ является предком $w$. Тогда отношение $<$ является частичным отношением порядка на $V$.* ∎

В терминах орграфов можно дать точное определение структурам данных, известным как *ориентированное дерево.*

*Ориентированное дерево* $T = (V, E)$ — это ациклический орграф, в котором одна вершина $v_r \in V$ не имеет предков, а каждая другая вершина имеет только одного непосредственного предка; $v_r$ называется *корнем* дерева. *Бинарное дерево* — это ориентированное дерево, в котором каждая вершина имеет не более двух непосредственных потомков, т. е. $\delta^+(v) \leqslant 2$ для всех $v \in V$. Говорят, что бинарное дерево является *полным*, если каждая вершина, не являющаяся листом, имеет ровно два непосредственных потомка.

Предложение. *Следующие утверждения эквивалентны по отношению к орграфу* $G=(V, E)$:

а) *$G$ является деревом.*

б) *Граф $F(G)$ связный, и существует вершина $v_r$, которая не имеет предков, а все другие вершины имеют только по одному непосредственному предку.*

в) *$G$ имеет вершину $v_r$, которая соединяется с любой другой вершиной единственным маршрутом.*

г) *$G$ имеет вершину $v_r$, которая не имеет предков; все другие вершины имеют только одного непосредственного предка; существует маршрут к каждой вершине из $v_r$.*

Доказательство оставляем в качестве упражнения. //

**6.3. Упорядоченные орграфы и обходы.** Списки смежности являются альтернативной по отношению к матрице смежности формой представления орграфов. Заданное списком смежности представление определяет порядок ребер, выходящих из каждой вершины.

Определение. *Упорядоченным орграфом* называется пара $G=(V, E)$, где $V$ — конечное множество вершин, а $E$ — множество упорядоченных списков ориентированных ребер. Элементы $E$ имеют вид

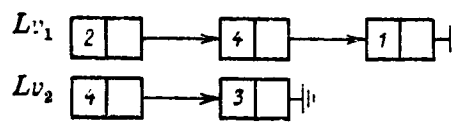
$$L_v=((v, w_1), \ldots, (v, w_k)),$$

где $v, w_i \in V$. //

Пример 6.6. Упорядоченный орграф

$$G=(\{v_1, v_2, v_3, v_4\}, \{((v_1, v_2), (v_1, v_4), (v_1, v_1)), ((v_2, v_4), (v_2, v_3))\})$$

может быть представлен списками смежности




и может быть изображен диаграммой (рис. 7.26).

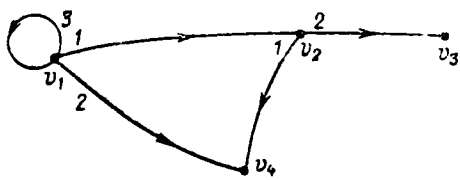



Рис. 7.26

Упорядоченный орграф $G$ определяет единственный неупорядоченный орграф; мы только заменяем каждый список $((v, w_1), \ldots, (v, w_k))$ множеством $\{(v, w_1), \ldots, (v, w_k)\}$. Орграф, определенный таким образом, называют орграфом, *подчиненным* $G$. *Упорядоченный* ациклический орграф является упорядоченным графом, чьим подчиненным орграфом является ациклический орграф. *Упорядоченное ориентированное дерево* является упорядоченным орграфом, чей подчиненный орграф является ориентированным деревом.

Пример 6.7.

$$T = (\{v_1, \ldots, v_6\}, \{((v_1, v_2), (v_1, v_3)), ((v_3, v_4), (v_3, v_5), (v_3, v_6))\})$$

является упорядоченным ориентированным деревом, где $v_1$ — корень. Оно может быть изображено, как показано на рис. 7.27. //

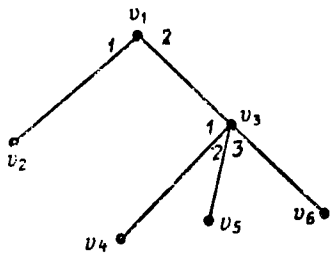


Рис. 7.27

Упорядоченные ориентированные деревья будем изображать спуском вершин слева направо (рис. 7.27). Если принять такое соглашение, то номера ребер можно опускать.

Определение.

1) Пусть $S_V$ и $S_E$ — множества. *Пометкой* упорядоченного орграфа $G = (V, E)$ называется пара отображений $(f, g)$, где

$$f\colon V \to S_V \text{ — пометка вершин,}$$

$$g\colon E \to \bigcup_{k=1}^{\infty} S_E^k \text{ — пометка ребер.}$$

Отображение $g$ имеет вид

$$g(((v, w_1), \ldots, (v, w_k))) = (\alpha_1, \ldots, \alpha_k) \in S_E^k.$$

2) Говорят, что два помеченных графа $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ с функциями пометок $(f_1, g_1)$ и $(f_2, g_2)$ соответственно эквивалентны, если существует биекция $h\colon V_1 \to V_2$ такая, что

а) $((v, w_1), \ldots, (v, w_k)) \in E_1$ тогда и только тогда,

когда

$$((h(v), h(w_1)), \ldots, (h(v), h(w_k)))\in E_2$$

(эквивалентны как упорядоченные графы);

б) $f_1(v)=f_2(h(v))$ для всех $v\in V$ (метки вершин совпадают);

в) для всех $((v, w_1), (v, w_2), \ldots, (v, w_k))\in E_1$ имеем

$$g_1(((v, w_1), \ldots, (v, w_k)))=$$
$$=g_2(((h(v), h(w_1)), \ldots, (h(v), h(w_k))))$$

(метки ребер совпадают). //

Следуя § 5, определим обход орграфа как перестановку или полное упорядочивание вершин. Для упорядоченных орграфов все делается точно так же. Для упорядоченных ориентированных деревьев часто полезны другие обходы. Некоторые из них будут описаны ниже.

Определение. Пусть $T=(\{v_1, \ldots, v_n\}, E)$ — упорядоченное ориентированное дерево и $L_v=((v, w_1), \ldots, (v, w_k))\in E$. Определим *отношение* $<$ на множестве $\{w_1, \ldots, w_k\}$ следующим образом: $w_i<w_j$ тогда и только тогда, когда $i<j$. Определим таким образом отношение $<$ для каждого списка $E$. //

Предложение. *Отношение* $<$ *является отношением частичного порядка на* $V$.

Доказательство.

Из соотношения $v<v$ следует список вида

$$((q, w_1), \ldots, (q, v), \ldots, (q, v), \ldots, (q, w_k)),$$

который невозможен, так как в дереве не существует циклов. Следовательно, $v\not<v$ для любого $v\in V$.

Из соотношений $v<w$ и $w<u$ следует, что существуют $x, y\in V$ такие, что

$$L_x=((x, w_1), \ldots, (x, v), \ldots, (x, w), \ldots, (x, w_k)),$$
$$L_y=((y, u_1), \ldots, (y, w), \ldots, (y, u), \ldots, (y, u_l)).$$

Если $x\neq y$, то $\delta^-(w)=2$, что невозможно, так как $T$ — дерево. Следовательно, $x=y$ и

$$L_x=((x, w_1), \ldots, (x, v), \ldots, (x, w), \ldots$$
$$\ldots, (x, u), \ldots, (x, w_k)),$$

т. е. $v<u$. Поэтому отношение $<$ есть частично упорядоченное отношение на $V$. //

Отношение $<$ сравнивает только вершины, выходящие из одной вершины,

Пример 6.8. Для упорядоченного ориентированного дерева на рис. 7.28 имеем

$< = \{(v_2, v_3), (v_4, v_5), (v_4, v_6), (v_5, v_6)\}$ или

$v_2 < v_3$, $v_4 < v_5$, $v_4 < v_6$ и $v_5 < v_6$. //

Замечание. Обозначим множество всех спусков из вершины $v \in V$ через $\Gamma^+(v)$; аналогично через $\Gamma^-(v)$ обозначим множество входов в $v$.

Определение. Отношение $<$ называют *трансверсальным порядком* вершин упорядоченного направленного дерева $T = (V, E)$. //

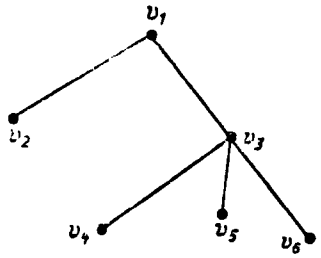


Рис. 7.28

Наша цель в оставшейся части главы — вывести различные полезные методы обхода для дерева с использованием симметрии путем расширения трансверсального порядка. Перед точным определением обходов необходимы два результата. Пусть $<$ определяет отношение $<_*$ на $V$ следующим образом: если $w_i < w_j$, то $w_i' <_* w_j'$ для всех $w_i \in \Gamma^+(w_i) \cup \{w_j\}$ и для всех $w_j' \in \in \Gamma^+(w_j) \cup \{w_j\}$.

Предложение.

1) $< \subseteq <_*$;

2) $<_*$ — *частично упорядоченное отношение на* $V$.

Доказательство.

1) Утверждение очевидно.

2) а) Из $v <_* v$ следует, что или $v < v$, или существуют $x$, $y \in V$ такие, что $x < y$ и $v \in \Gamma^+(x) \cup \{x\}$ и $v \in \in \Gamma^+(y) \cup \{y\}$. Однако $v \not< v$, так как $<$ — отношение частичного порядка. Аналогично из $x < y$ следует $x \neq y$; следовательно, $v \in \Gamma^+(x) \cup \{x\}$ и $v \in \Gamma^+(y) \cup \{y\}$. Однако это означает, что $\delta^-(v) = 2$ или $\delta^-(w) = 2$ для некоторого $w \in \Gamma^-(v)$. Это невозможно, потому что $T = (V, E)$ — дерево. Таким образом, $v <_* v$ для всех $v \in V$.

б) $v <_* w$ означает, что $v < w$ или что существуют $x$, $y \in V$ такие, что $x < y$ и $v \in \Gamma^+(x) \cup \{x\}$ и $w \in \Gamma^+(y) \cup \cup \{y\}$; $w <_* u$ означает, что $w < u$ или что существуют $r$, $s \in V$ такие, что $r < s$ и $w \in \Gamma^+(r) \cup \{r\}$ и $u \in \Gamma^+(s) \cup \cup \{s\}$; $w \in \Gamma^+(r) \cup \{r\}$ и $w \in \Gamma^+(y) \cup \{y\}$ дают, что или $r \in \Gamma^+(y)$, или $y \in \Gamma^+(r)$.

Следовательно, дерево имеет одну из форм, изображенных на рис. 7.29. Если $r \in \Gamma^+(y)$, то $s \in \Gamma^+(y)$ и

$x < y$ дает $x <_e s$. Однако $u \in \Gamma^+(s)$, следовательно, $x <_e u$, и так как $v \in \Gamma^+(x)$, то, следовательно, $v <_e u$. Если $y \in \Gamma^+(r)$, то $x \in \Gamma^+(r)$ и $r < s$ дает $x <_e s$. Однако

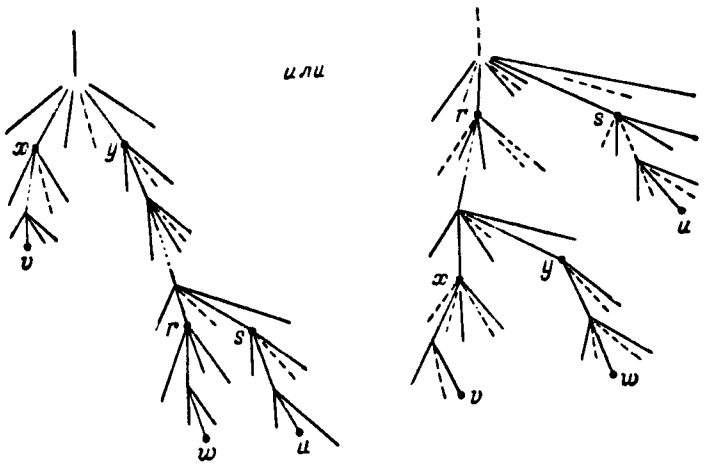



Рис. 7.29

$u \in \Gamma^+(s) \cup \{s\}$; следовательно, $x <_e u$, а $v \in \Gamma^+(x)$ дает $v <_e u$.

Следовательно, $<_e$ является отношением частичного порядка на $V$. //

Иногда $v <_e w$ читают как «$v$ находится слева от $w$».

Предложение. *Пусть $T = (V, E)$ — упорядоченное ориентированное дерево. Тогда для $v_i, v_j \in V$ $(i \neq j)$ или $v_i <_e v_j$, или $v_j <_e v_i$, или же $v_i$ и $v_j$ находятся на маршруте.*

Доказательство. Пусть $v_i, v_j \in V$ и $v_i \neq v_j$. Тогда существует вершина $v_a$, такая, что $v_i \in \Gamma^+(v_a) \cup \{v_a\}$ и $v_j \in \in \Gamma^+(v_a) \cup \{v_a\}$. Если $v_i = v_a$ или $v_j = v_a$, то $v_i$ и $v_j$ находятся на маршруте; в противном случае рассмотрим прямые спуски $w_1, \ldots, w_k$ из $v_a$. Тогда или

$$v_i \in \Gamma^+(w_l) \text{ и } v_j \in \Gamma^+(w_m) \text{ для } w_l \neq w_m,$$

или

$$v_i, v_j \in \Gamma^+(w_k).$$

В первом случае имеем или $v_i <_e v_j$, или $v_j <_e v_i$ в зависимости от того, $w_l < w_m$ или $w_m < w_l$. Во втором случае повторяем процесс из $w_k$, пока не будут выполнены условия первого случая, или получаем $v_i = v_a$ или $v_j = v_a$; в этом случае $v_i$ и $v_j$ находятся на маршруте. //

Много полезных обходов дерева определяют посредством расширения отношения $<_e$ до полного упорядочивания $V$. Используя приведенный выше результат, надо только расширить $<_e$, чтобы сравнивать вершины, которые находятся на маршруте, для определения полного порядка на $V$, для которого $<_e$ является подпорядком.

Определение. Пусть $T=(V, E)$ — упорядоченное направленное дерево. Определим полную упорядоченность $<_1$ на $V$ следующим образом: если $v_j$ спускается из $v_i$, то $v_i<_1 v_j$; в противном случае $v_i<_1 v_j$, если $v_i<_e v_j$. Отношение $<_1$ называют *предпорядком* на $V$. //

Очевидно, что $\leq_e \subseteq <_1$.

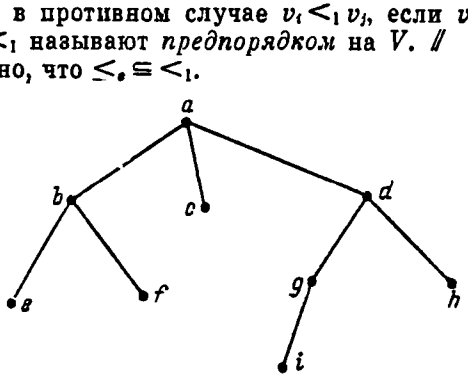



Рис. 7.30

Пример 6.9. Пусть $T=(V, E)$ — упорядоченное направленное дерево, изображенное на рис. 7.30. Тогда предпорядок на $V=\{a, b, c, d, e, f, g, h, i\}$ можно записать как

$$a<_1 b<_1 e<_1 f<_1 c<_1 d<_1 g<_1 i<_1 h.$$

Соответствующий обход вершины имеет вид

$$a, b, e, f, c, d, g, i, h.//$$

Определение. Пусть $T=(V, E)$ — упорядоченное ориентированное дерево. Определим отношение $<_2$ полного порядка на $V$ следующим образом: если $v_i$ спускается из $v_j$, то $v_i<_2 v_j$; в противном случае $v_i<_2 v_j$, если $v_i<_e v_j$. Отношение $<_2$ называют *постпорядком* на $V$. //

Очевидно, что $<_e \subseteq <_2$.

Пример 6.10. Пусть $T$ и $V$ такие же, как и в предыдущем примере. Тогда постпорядок на $V$ будет иметь вид

$$e<_2 f<_2 b<_2 c<_2 i<_2 g<_2 h<_2 d<_2 a$$

с обходом

$$e, f, b, c, i, g, h, d, a. \ //$$

Определение. Пусть $T=(V, E)$ — полное бинарное дерево. Определим *симметричный порядок* $<_s$ на $V$ следующим образом: для каждой вершины, не являющейся листом и имеющей прямые спуски $w_1$ и $w_2$ $(w_1 < w_2)$, положим

$$w_1' <_s v \quad \text{для всех } w_1' \in \Gamma^+(w_1) \cup \{w_1\},$$

$$v <_s w_2' \text{ для всех } w_2' \in \Gamma^+(w_2) \cup \{w_2\}. \ //$$

Порядок $<_s$ также является расширением $<_e$.

Пример 6.11. Пусть $T$ — дерево, изображенное на рис. 7.31. Тогда $<_s$ определено, как показано на рис 7.31.

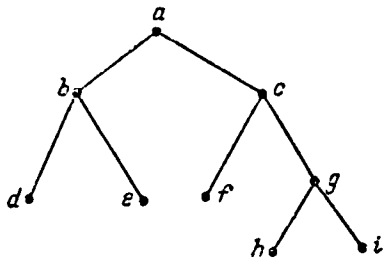



Рис. 7.31

Обход по предпорядку упорядоченного ориентированного дерева соответствует обходу по глубине с корнем в качестве начальной вершины.

Если $T=(V, E)$ — упорядоченное ориентированное дерево, то вершины $T$ могут быть расположены в предпорядок с применением следующего алгоритма, начинающегося с корня:

```
procedure pre(v)
process v
if L_v ≠ ∅ do pre(w) for each w ∈ L_v
endproc
```

Соответствующий алгоритм для постпорядка имеет вид

```
procedure post(v)
if L_v = ∅ then process v
          else begin
               post(w) for each w ∈ L_v
               process v
               end
```

Упражнение 7.6.

1. Пусть $A \in \mathcal{M}(n, \mathrm{B})$ — матрица смежности орграфа. Выписать выражения для функций $\delta^+$ и $\delta^-$ в терминах $A$.

2. Сколько различных орграфов может существовать на вершинах?

3. Пусть $G = (V, E)$ — орграф. Какое максимально возможное значение величины $|E|$?

4. Пусть $G = (V, E)$ — орграф, где $V = \{v_1, v_2, v_3, v_4, v_5\}$ и $E = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_3)\}$. Определить матрицы $R(E^+)$ и $R(E^*)$.

5. Дать матричные характеристики слабой, односторонней и сильной связности в орграфе.

6. Пусть $G = (V, E)$ — ациклический орграф. Какое максимально возможное значение величины $|E|$?

7. Доказать два последних предложения в п. 6.2.

8. Показать, что если $T = (V, E)$ — полное бинарное дерево и $V^l$ обозначает множество уходов из $T$, то $|V^l| = |V \backslash V^l| + 1$.

9. *Поддеревом* $T' = (V', E')$ ориентированного дерева $T = (V, E)$ называется такое ориентированное дерево, что:

а) $\varnothing \neq V' \subseteq V$;

б) $E' = V' \times V' \cap E$;

в) ни одна из вершин $V \backslash V'$ не является спуском вершины в $V'$. Нарисовать все поддеревья

$$T = (\{v_1, v_2, \ldots, v_6\}, \{(v_1, v_2), (v_1, v_3), (v_3, v_4), (v_3, v_5), (v_3, v_6)\}).$$

10. Определение. Если $T$ — ориентированное дерево, то *уровень* вершины определяют как максимальную длину маршрута от этой вершины до листа. *Глубина* вершины — это длина пути от корня до этой вершины. *Глубиной* $T$ называют длину самого длинного маршрута в $T$. *Высотой вершины* $T$ называют глубину $T$ за вычетом глубины вершины. *Высота* $T$ является высотой корня. //

Пусть $T$ — ориентированное дерево

$$(\{v_1, \ldots, v_9\}, \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_3, v_5), (v_3, v_6), (v_3, v_7), (v_5, v_8), (v_5, v_9)\}),$$

а) нарисовать $T$ со значениями уровней в качестве меток вершины;

б) нарисовать $T$ со значениями глубин в качестве меток вершины;

в) нарисовать $T$ со значениями высот в качестве меток вершин;

г) чему равна глубина $T$?

д) чему равна высота $T$?

11. Пусть $T$ — ориентированное дерево. *Разрезом* $C$ дерева $T$ называется подмножество вершин $T$ таких, что

а) не существует двух вершин $C$ на маршруте в $T$;

б) ни одна вершина не может быть добавлена к $C$ без нарушения а).

Определить все разрезы ориентированного дерева, изображенного на рис. 7.32.

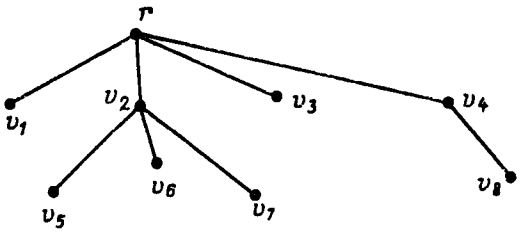



Рис. 7.32

12. Пусть $T=(V, E)$ — полное бинарное дерево и $|V|=n$. Показать, что существует $n!/2^{(n-1)/2}$ полных порядков на $V$, которые расширяют трансверсальный порядок $<$.

13. Проинтерпретировать $<_1$, $<_2$ и $<_3$ для бинарных деревьев, связанных со структурой арифметических выражений, как в § 6 гл. 3.

ГЛАВА 8

# ЯЗЫКИ И ГРАММАТИКИ

Все средства общения включают язык. Обычно мы «общаемся» с компьютером при помощи языка, который каким-либо образом записывается (перфокарты, телетайп, экран дисплея и т. п.), и, следовательно, предложения языка состоят из строк символов. Действительно, весь вычислительный процесс может рассматриваться как преобразование одного множества строк в другое. Такие процессы ведут себя совершенно определенным образом, и, следовательно, с ними можно обращаться как с математическими объектами — по крайней мере строки могут рассматриваться как элементы моноида. В этой главе мы будем подходить к изучению языка скорее с математической точки зрения, чем с литературной. В § 1 будет введено понятие строки и будут рассмотрены некоторые проблемы, относящиеся к этому вопросу; в § 2 будут введены языковые структуры. Далее будут более детально исследованы некоторые важные классы языков и рассмотрено введение в грамматический разбор.

## § 1. Основные понятия

**1.1. Строки.** *Буква* (или *символ*) — это простой неделимый знак, или символ; множество букв образует *алфавит*.

Пример 1.1.

$$A = \{a, b, c\}, B = \{0, 1\},$$

$$C = \{\text{PERFORM, ADD, GIVING, TO, } \ldots\},$$

$$E = \{a, b, c, \ldots, x, y, z\}.$$

Здесь мы можем рассматривать $B$ как бинарный алфавит, $C$ — как алфавит языка Кобол (в котором слова типа PERFORM не могут быть разделены), а $E$ — как английский алфавит.

Алфавиты являются множествами, и поэтому к ним можно применять теоретико-множественные обозначения. В частности, если $A$ и $B$ такие алфавиты, что $A \subseteq B$, то будем говорить, что $A$ является *подалфавитом* $B$, или что $B$ является *расширением* $A$.

*Строки* являются упорядоченными совокупностями букв алфавита (например, алфавита $A$) и, следовательно, выглядят подобно элементам $A^n = A \times A \times \ldots \times A$. Однако будет более естественным записывать их в виде $a_1a_2\ldots a_n$, а не $(a_1, a_2, \ldots, a_n)$. Буквы сами по себе также являются строками для случая $n = 1$. Мы будем допускать случай, когда строка не имеет букв (*пустая строка*), и обозначать эту строку через $\Lambda$. Заметим, что $\Lambda$ не является символом, т. е. $\Lambda \notin A$ для любого алфавита $A$.

По аналогии с лингвистикой будем строки также называть *словами*. Множество всех строк (слов) над алфавитом $A$ называют *замыканием* $A$ и обозначают $A^*$, так что

$$A^* = A^0 \cup A^1 \cup A^2 \cup \ldots = \bigcup_{n=0}^{\infty} A^n, \quad \text{где} \quad A^0 = \{\Lambda\}.$$

Для удобства определим также множество непустых строк над $A$ следующим образом:

$$A^+ = A^* \setminus \{\Lambda\} = \bigcup_{n=1}^{\infty} A^n.$$

Как уже упоминалось в примере 2.1 гл. 5, основная операция над строками называется конкатенацией; формально она может быть определена как бинарная операция $\odot$ на $A^*$ следующим образом:

$$\odot\colon (\alpha, \beta) \mapsto \alpha\beta.$$

Аналогично эта же операция определяется для $A^+$. Операция $\odot$ ассоциативна, но не коммутативна; $\Lambda$ является единицей в $A^*$ по отношению к $\odot$. Сформулируем ниже основные свойства операции $\odot$ на $A^*$ и $A^+$.

Предложение. *По отношению к операции* $\odot$

а) $A^*$ *является моноидом*;

б) $A^+$ — *полугруппа*. //

Результат слияния строк $\alpha$ и $\beta$, т. е. $\alpha\beta$, заключается в следующем: строка $\beta$ записывается сразу же за строкой $\alpha$. Другой способ определения строки — рекурсивный: $\alpha$ является строкой над алфавитом $A$, если или $\alpha = \Lambda$, или $\alpha = a\beta$, где $a \in A$, а $\beta$ — строка над $A$. Здесь $a\beta$ означает, что буква $a$ стоит непосредственно перед строкой $\beta$.

Все слова в $A^n$ ($n \in \{0\} \cup \mathbf{N}$) состоят точно из $n$ букв; в этом случае говорят, что строка имеет *длину* $n$. Длина $\alpha \in A^*$ обозначается как $|\alpha|$ и $|\alpha| = n$ тогда и только тогда, когда $\alpha \in A^n$.

Очевидно, что $|\Lambda| = 0$ и $|a| = 1$ тогда и только тогда, когда $a \in A$.

Если строки состоят из повторяющихся букв, то обычно принимают сокращенные обозначения, чтобы показать, что строку следует рассматривать как произведение (по отношению к операции конкатенации). Поэтому для $a \in A$ будем писать

$$\Lambda = a^0, \qquad aa = a^2, \qquad aa^{n-1} = a^n, \qquad n \in \mathbf{N}.$$

Будем использовать следующее обозначение для повторяющихся строк: строку $ababa$ будем записывать как $(ab)^2a$ или же как $a(ba)^2$.

Этот пример иллюстрирует одну из основных трудностей при рассмотрении строк. Мы используем строки для описания строк, и, следовательно, мы должны иметь возможность различать используемые алфавиты. Если приведенные выше выражения рассматривались над алфавитом $A$, а символов «(» и «)» в $A$ нет, тогда смысл понятен; с другой стороны, если круглые скобки есть в $A$, то выражение $(ab)^2a$ может быть понято как $(ab))a$. При условии что мы осознаем возможность таких проблем и знаем, какие алфавиты используются, этих ошибок можно избежать, используя различные алфавиты и биекцию между двумя алфавитами. В некоторых случаях может быть более удобно построить мономорфизм между множествами строк, так что одно множество рассматривается над существенно более простым алфавитом.

Пример 1.2. Пусть $B = \{0, 1\}$ и $A = \{a, b, c\}$. Тогда $\varphi$: $A^* \to B^*$, определенное соотношением

$$\varphi(xy) = \varphi(x)\varphi(y),$$

где $x, y \in A^*$ и

$$\varphi(a) = 0, \qquad \varphi(b) = 10, \qquad \varphi(c) = 110,$$

является гомоморфизмом моноида (нам надо сохранять комбинации) из $A^*$ в $\{0, 10, 110\}^* \subseteq B^*$. Например,

$$\varphi\colon abbca \to 010101100, \quad \varphi^{-1}\colon 01011010110 \to abcbc.$$

В действительности этот метод построения может использоваться для отображения произвольного конечного алфавита в $\{0, 1\}^*$. //

Из определения длины строки следует, что если $\alpha, \beta \in A^*$, то

$$|\alpha\beta| = |\alpha| + |\beta|, \qquad |\alpha^n| = n|\alpha|.$$

Более того, если $a \in A$, то

$$|a^n| = n.$$

При преобразовании одной строки в другую нежелательно, чтобы вся входная строка изменялась под действием одной операции; в противном случае процесс можно было бы определить только с помощью множества входных-выходных пар. В дальнейшем нам понадобится понятие подстроки.

Пусть заданы строки $\alpha$ и $\beta$ над алфавитом $A$. Строка $\beta$ называется *подстрокой* $\alpha$, если

$$\alpha = \gamma\beta\delta, \qquad \gamma, \delta \in A^*.$$

Пример 1.3. Пусть $A = \{a, b, c\}$ и $\alpha = abac$. Тогда подстроками $\alpha$ являются следующие строки:

$$\Lambda, a, b, c, ab, ba, ac, aba, bac, abac.$$

В частности, заметим, что $\alpha$ является подстрокой самой себя, а $\Lambda$ — подстрокой $\alpha$ (и любой другой строки), поскольку

$$\alpha = \Lambda abac = a\Lambda bac = aa\Lambda bac = aa\Lambda ba\Lambda c =$$
$$= aa\Lambda ba\Lambda c\Lambda = a\Lambda bac = a\Lambda\Lambda bac = \ldots \text{ //}$$

Выделение подстроки естественно приводит к замене подстроки другой строкой. Однако пока еще мы не достигли уровня, достаточного для корректного выполнения этой операции. Рассмотрим, что случится, если мы имеем «функцию» $f$, которая замещает строку $xy$ строкой $yxx$ при условии, что первая строка является подстрокой в операнде. Тогда

$$f(pqxy) = pqyxx.$$

Однако неясно, что́ является результатом $f(xyxxy)$. Возможны два случая — $yxxxy$ или $xyyxx$. Аналогично получается, если мы применяем $f$ несколько раз. Тогда $f^2(xyxxy) = yxxyxx$, однако, применяя $f$ к $xxyy$, получаем

$$f(xxyy) = xyxxy, \qquad f^2(xxyy) = yxxxxy \text{ или } xyxyxx.$$

Таким образом, «функция» является не полностью определенной, и мы не можем поправить дело, потребовав,

чтобы операция изменяла все подстроки, поскольку они могут частично перекрываться.

Ситуация еще более усложняется, если применяются несколько замещающих функций. Необходимы средства выбора определенной подстроки всякий раз, когда возникает такой выбор; в частности, мы будем рассматривать подстроку, встречающуюся первой при чтении слева направо.

Для формализации рассуждений будем использовать порядковые свойства целых чисел и сами целые числа, соответствующие длинам строк. Предположим, что $\alpha$ и $\beta$ — строки над $A$, $|\alpha| \leqslant |\beta|$ и $\alpha$ является подстрокой $\beta$. Предположим, что $\alpha$ в $\beta$ встречается $m$ различными способами и что $|\beta| - |\alpha| = n$. Тогда мы можем записать $\beta m$ различными способами:

$$\beta = \gamma_1\alpha\delta_1 = \gamma_2\alpha\delta_2 = \ldots = \gamma_m\alpha\delta_m,$$

где $\gamma_i, \delta_i \in A^*$, $1 \leqslant i \leqslant m$ и $m \leqslant n+1$ (если $\alpha = \beta$, то $n = 0$; поэтому существует только одна возможность). Будем говорить, что $\gamma_1, \ldots, \gamma_m$ *специфицируют* различные вхождения $\alpha$ в $\beta$ и что $\gamma_1$ дает *первое вхождение*, а вхождение $\alpha$ непосредственно за $\gamma_1$ является *первым вхождением*.

Пример 1.4. Пусть $g$ — функция $A^* \to A^*$ такая, что $\{x, y, p, q\} \subseteq A$, и $g$ заменяет первое вхождение $xy$ в строке на $yxx$. Тогда

$$g(pqxy) = pqyxx, \qquad g(xypxy) = yxxpxy,$$
$$g^2(xypxy) = yxxpyxx.$$

Заметим также, что

$$g^6(x^2y^2) = g^5(xyx^2y) = g^4(yx^4y) = g^3(yx^3yx^2) =$$
$$= g^2(yx^2yx^4) = g(yxyx^6) = y^2x^8,$$
$$g(y^2x^8) = y^2x^8;$$

поэтому

$$g^6(x^2y^2) = g^7(x^2y^2) = \ldots \text{ //}$$

Перед тем как продолжить изложение, отметим, что существует альтернативный набор терминов для буквы, алфавита и слова; это — *слово*, *словарь* и *предложение* соответственно. В некоторых контекстах эти термины более разумны, однако в этом случае необходимо проявлять особую тщательность в использовании термина «слово», поскольку оно имеет два смысла.

**1.2. Языки.** Совокупность строк (или предложений) называется языком. Формально *язык* $L$ над алфавитом

$A$ — это множество строк в $A^*$; поэтому $L \subseteq A^*$. Следовательно, операции над строками индуцируют операции на языках. Отсюда получаем $L^+$ (*транзитивное замыкание* $L$) и $L^*$ (*рефлексивное замыкание* $L$) следующим образом:

а) $L^0 = \{\Lambda\}$;

б) если $L_i$ и $L_j$ — языки, то $L_iL_j = \{xy\colon\ x \in L_i,\ y \in L_j\}$;

в) $L^n = L^{n-1}L$, $n \in \mathbf{N}$;

г) $L^+ = \cup_{n>1} L^n$;

д) $L^* = \cup_{n>0} L^n$.

Сейчас обратим внимание на то, как слова могут составляться в предложения, а множество всех предложений, имеющих смысл, образует язык. Нас будут в основном интересовать искусственные языки, такие как языки программирования или языки, описывающие правильные математические выражения, однако вначале будет полезно рассмотреть случай английского языка. Это даст возможность сформулировать некоторые определения таким образом, что мы сможем сделать первые шаги в теорию языка. Возьмем предложение

«The dog bit me».

Это предложение можно рассматривать двумя способами. Во-первых, изучать его как простую совокупность слов, каждое из которых является упорядоченной совокупностью букв; в этом случае предложение рассматривается *синтаксически*. Во-вторых, интерпретировать предложение, считая, что мы понимаем значения слов и их внутренние связи; тогда мы получаем *семантику* — значение предложения. В дополнение заметим, что если мы произносим предложение, то оно влияет на нас своим воздействием — прагматизмом. В совокупности эти три области образуют *семиотику* языка.

Пример 1.5. В языках программирования Фортран и Кобол утверждения

$A = B + C$, ADD $B$ TO $C$ GIVING $A$

имеют одинаковый семантический смысл понятий сложения и присваивания, однако у них разный синтаксис. Прагматически они могут быть представлены на некоторой машине как результат выполнения кода

LOAD $B$
ADD $C$
STORE $A$. //

Основным объектом нашего рассмотрения будет область синтаксиса. Чтобы проиллюстрировать класс структур, которые мы будем изучать, рассмотрим диаграмму на рис. 8.1.

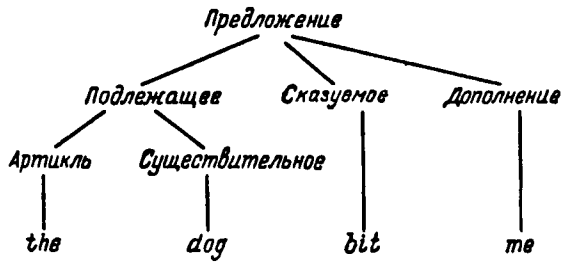



Рис. 8.1

На самом деле эта диаграмма означает, что ⟨предложение⟩ может быть построено путем слияния ⟨подлежащего⟩, ⟨сказуемого⟩ и ⟨дополнения⟩, хотя это требует формального определения. Подлежащее состоит из ⟨артикля⟩ с ⟨существительным⟩, и окончательно получаем ⟨артикль⟩ «the», ⟨существительное⟩ «dog», ..., что дает нам предложение «the dog bit me».

Перед тем как ввести терминологию и обозначения, необходимые для уточнения общих понятий в конкретной ситуации, изображенной на рис. 8.1, мы установим основные цели теории языков и сделаем обзор оставшейся части главы.

Напомним, что для заданного алфавита $A$ язык $L$ является произвольным подмножеством множества $A^*$, однако произвольные подмножества представляют очень незначительный интерес. Мы хотим сосредоточить внимание на специальных языках, содержащих строки, которые благодаря внешней информации об их семантике считаются осмысленными или хорошо сконструированными.

Наиболее интересные языки бесконечны и, следовательно, не могут быть выписаны явно. В этих случаях надо придумать способы порождения языка; грамматика $G$ может рассматриваться как такая порождающая система. Сформулируем две основные задачи формальной теории языков:

а) Как по заданной грамматике $G$ (и связанным с ней языком $L$) порождать предложения $\alpha$: $\alpha \in L$?

б) Как по заданным $L \leqslant A^*$ и $\alpha \in A^*$ устанавливать, принадлежит ли $\alpha \in L$?

Для того чтобы проверить, входят ли эти строки в $L$, надо знать, как $L$ порождается грамматикой $G$. В § 2 и 3 мы опишем общие принципы грамматик с фразовой структурой, а затем подробнее рассмотрим некоторый подкласс, имеющий большое практическое значение.

Обозначим через $L(G)$ язык, порожденный грамматикой $G$. Тогда алгоритм проверки вхождения $\alpha \in L(G)$ называется грамматическим разбором; он использует $\alpha$ и $G$. Часто первоначальная грамматика не подходит для определенной техники разбора, однако она может быть преобразована к эквивалентному, более подходящему виду. Вопросы, относящиеся к грамматическому разбору или модификации грамматики, будут рассматриваться в § 4.

Проводя далее идею манипулирования грамматикой $G$, в некоторых весьма специальных (но обычных) ситуациях можно перенести почти все (а иногда и все) трудности грамматического разбора в анализ грамматики и, следовательно, намного упростить анализ конкретных строк. Спецификация ограничений, которым должны удовлетворять эти грамматики, является более сложной. Изучению этих проблем посвящен § 5.

Упражнение 8.1.1. Предположим, что $A$ и $B$ — непустые алфавиты такие, что

$$|A| = p, \quad |B| = q,$$

и $\varphi$: $A \to \mathbf{N}_p$, $\psi$: $B \to \mathbf{N}_q$ — биекции. Пусть $\chi_1$: $A^* \to \mathbf{N}$ определено как

$$\chi_1\colon\ a_1 \dots a_k \mapsto \sum_{i=1}^{k} \varphi(a_i) * p^{(i-1)},$$

а $\chi_2$: $B^* \to \mathbf{N}$ определено как

$$\chi_2\colon\ b_1 \dots b_k \mapsto \sum_{i=1}^{k} \psi(b_i) * q^{(i-1)}.$$

Доказать, что $\chi_2^{-1} \circ \chi_1$ является биекцией строк, и показать, как применяется это отображение. Для этого надо найти прямые и обратные образы строк $x^2yxz$ в $A^*$ и 145332 в $B^*$, где

$$A = \{x, y, z\}, \quad B = \{1, 2, 3, 4, 5\}.$$

## § 2. Грамматики с фразовой структурой

### 2.1. Основные определения.

Определение. *Грамматикой с фразовой структурой* (ГФС) $G$ называется алгебраическая структура, состоящая из упорядоченной четверки $(N, T, P, S)$, где:

а) $N$ и $T$ — непустые конечные алфавиты *нетерминальных* и *терминальных символов* *) соответственно таких, что $N \cap T = \varnothing$;

б) $P$ — конечное множество *продукций*, $P \subseteq V^+ \times V^*$, где $V = N \cup T$ называется *словарем* $G$;

в) $S \in N$ называется *начальным символом* или *источником*. //

Предполагая, что символ $\rightarrow$ не содержится в $V$, соотношение $(\alpha, \beta) \in P$ обычно записывают в виде $\alpha \rightarrow \beta$.

Понятие продукции, которую также называют *правилом преобразования*, должно давать возможность заменять одну строку символов другой. Терминальные символы обычно рассматриваются как неизменяемые символы. Поэтому, возможно, определение продукции в ГФС является чрезмерно общим. На практике соответствующие ограничения будут вводиться так, чтобы не нарушать постоянства терминальных символов, однако сейчас этого определения достаточно.

В качестве первого шага рассмотрим рис. 8.1 и попытаемся понять, как он связан со следующими примерами.

Пример 2.1. Предложение на английском языке, приведенное ранее в качестве иллюстрации, может быть определено в грамматике $G = (N, T, P, S)$, где $N = \{$⟨предложение⟩, ⟨подлежащее⟩, ⟨артикль⟩, ⟨существительное⟩, ⟨сказуемое⟩, ⟨дополнение⟩$\}$; $T = \{$the, dog, bit, me$\}$; $P = \{$(⟨предложение⟩, ⟨существительное⟩, ⟨сказуемое⟩, ⟨дополнение⟩), (⟨существительное⟩, ⟨артикль⟩, ⟨подлежащее⟩), (⟨артикль⟩ the), (⟨подлежащее⟩ dog), (⟨сказуемое⟩ bit), (⟨дополнение⟩ me)$\}$; $S =$ ⟨предложение⟩.

Эта частная система порождает только одно предложение «the dog bit me» и, следовательно, может быть заменена на

$$N = \{\text{⟨предложение⟩}\},$$
$$P = \{(\text{⟨предложение⟩ the dog bit me})\}$$

или даже на

$$L = \{\text{the dog bit me}\}.$$

Однако если мы в данном случае захотим расширить язык, чтобы включить в него все предложения, начинающиеся со слов, скажем, «the lion», «the rat», «the tiger»,

*) Эти символы будут называться также нетерминалами и терминалами соответственно.— *Примеч. ред.*

со сказуемыми «ate» и «attacked» и дополнениями «you» и «Napoleon» (тогда $L$ будет иметь более 35 элементов), то это может быть сделано добавлением только семи дополнительных элементов к каждому из множеств $T$ и $P$. В этом примере размер языка составляет $4 * 3 * 3$, в то время как размер множества $P$ примерно равен $4 + 3 + 3$. Еще большее значение имеет тот факт, что мы можем включить все предложения вида «the dog bit (the son of)$^n$ Napoleon» (их бесконечное множество), добавляя к $T$ и $P$ незначительное число элементов. //

Перед тем как описать механизм порождения предложений, мы должны упомянуть нотацию, введенную Бэкусом (нормальная форма Бэкуса или форма Бэкуса-Наура, БНФ). Она особенно полезна, когда мы хотим использовать элементы из $N$, которые можно спутать с элементами из $T$ такими, как ⟨предложение⟩ и «предложение». Эта нотация использует четыре символа:

::= (мета-присвоить), ⟨ (мета-открыть),

⟩ (мета-закрыть), | (мета-или).

Понятия «мета-открыть» и «мета-закрыть» используются для того, чтобы выделять строки в качестве элементов $N$, «мета-присвоить» заменяет символ $\rightarrow$, и если $(\alpha, \beta) \in P$, $(\alpha, \gamma) \in P$, то это может быть записано в виде $\alpha ::= \beta \mid \gamma$, что читается как «$\alpha$ есть $\beta$ или $\gamma$».

БНФ впервые использовалась для определения синтаксиса Алгол-60. В случае, если у читателя имеются какие-либо сомнения в том, что БНФ способна определить что-нибудь серьезное, рекомендуем прочитать сообщение про Алгол-60. В работах по формальным языкам обычно избегают длинных строк в $N$ и, следовательно, нотация Бэкуса не используется, за исключением символа мета-или. Обычно прописные буквы используют для обозначения элементов $N$, а строчные — для элементов $T$.

Пример 2.2. Рассмотрим $G = (N, T, P, S)$, где

$$N = \{S, T\}, \qquad T = \{a, b, c, d\},$$
$$P = \{S \rightarrow aTd, \quad T \rightarrow bT \mid b \mid cT \mid c\}.$$

Заметим, что двойное использование $T$ в этом примере не вызывает никаких затруднений.

Грамматика будет порождать все строки $a\{b, c\}^+d$, однако мы все еще не показали, как этого можно до-

стичь. Будем использовать продукции следующим образом.

Пусть $\alpha, \beta \in V^*$; тогда $\beta$ *прямо выводится* из $\alpha$, если $\alpha = \gamma\sigma\beta$ и $\beta = \gamma\rho\delta$, где $\gamma, \delta, \rho \in V^*$, $\sigma \in V^+$ и $\sigma \to \rho \in P$. Этот факт будем записывать в виде $\alpha \Rightarrow \beta$; он может неформально рассматриваться как преобразования строки $\alpha$ в строку $\beta$ замещением подстроки $\sigma$ в $\alpha$ на $\rho$. (Заметим, что не обязательно заменять конкретное вхождение $\sigma$ в $\alpha$ или использовать конкретную продукцию с левой частью $\sigma$. Возможны любые вариации.)

Пусть теперь $\alpha$ и $\beta$ — слова над $V$ и существует конечная последовательность $\alpha_0, \alpha_1, \ldots, \alpha_r$, где $\alpha_0 = \alpha$, $\alpha_r = \beta$ и $\alpha_{i-1} \Rightarrow \alpha_i$ $(i = 1, \ldots, r)$. Тогда будем говорить, что $\alpha$ *порождает* $\beta$ (записывается $\alpha \overset{*}{\Rightarrow} \beta$) и что вывод $\beta$ из $\alpha$ реализуется следующим образом: $\alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots$ $\ldots \Rightarrow \alpha_{r-1} \Rightarrow \beta$. Аналогично $\alpha \overset{+}{\Rightarrow} \beta$, если вывод использует непустую последовательность прямых выводов. Если $\alpha \in V^*$ такое, что $S \overset{*}{\Rightarrow} \alpha$, то $\alpha$ называют *сентенциальной формой*. Более того, если $\alpha \in T^*$ и $S \overset{*}{\Rightarrow} \alpha$, то $\alpha$ является *предложением*, порожденным $G$. Таким образом, *язык* $L(G)$, порожденный $G$, есть $\{\alpha\colon \alpha \in T^* \text{ и } S^* \Rightarrow \alpha\}$. Там, где $G$ подразумевается, можно определить $L(X) = \{\alpha\colon \alpha \in T^*, X \in N \text{ и } X \overset{*}{\Rightarrow} \alpha\}$. Поскольку, применяя продукции к сентенциальным формам, можно действовать достаточно произвольно, то возможно существование нескольких допустимых выводных последовательностей для данного предложения в $L(G)$, где $G$ — конкретная грамматика. Среди этих последовательностей мы выбираем ту, которая на каждом этапе оперирует с самой левой из возможных подстрок, в которой элементы заменяются на элементы из $P$. Такая последовательность называется (левой) *канонической выводной последовательностью* для предложения.

Пример 2.3. Пусть

$$G = (\{B\}, \{(\ ,\ )\}, P, B),$$

где

$$P = \{B \to (B) \mid BB \mid (\ )\}.$$

Тогда предложение ( )(( )( )) может быть выведено многими способами.

Приведем пять из них:

1) $B \Rightarrow BB \Rightarrow$
$\Rightarrow (\ )B \Rightarrow$
$\Rightarrow (\ )(B) \Rightarrow$
$\Rightarrow (\ )(BB) \Rightarrow$
$\Rightarrow (\ )((\ )B) \Rightarrow$
$\Rightarrow (\ )((\ )(\ ))$;

2) $B \Rightarrow BB \Rightarrow$
$\Rightarrow (\ )B \Rightarrow$
$\Rightarrow (\ )(B) \Rightarrow$
$\Rightarrow (\ )(BB) \Rightarrow$
$\Rightarrow (\ )(B(\ )) \Rightarrow$
$\Rightarrow (\ )((\ )(\ ))$;

3) $B \Rightarrow BB \Rightarrow$
$\Rightarrow B(B) \Rightarrow$
$\Rightarrow (\ )(B) \Rightarrow$
$\Rightarrow (\ )(BB) \Rightarrow$
$\Rightarrow (\ )((\ )B) \Rightarrow$
$\Rightarrow (\ )((\ )(\ ))$;

4) $B \Rightarrow BB \Rightarrow$
$\Rightarrow B(B) \Rightarrow$
$\Rightarrow (\ )(B) \Rightarrow$
$\Rightarrow (\ )(BB) \Rightarrow$
$\Rightarrow (\ )(B(\ )) \Rightarrow$
$\Rightarrow (\ )((\ )(\ ))$;

5) $B \Rightarrow BB \Rightarrow$
$\Rightarrow B(B) \Rightarrow$
$\Rightarrow B(BB) \Rightarrow$
$\Rightarrow B((\ )B) \Rightarrow$
$\Rightarrow (\ )((\ )B) \Rightarrow$
$\Rightarrow (\ )((\ )(\ ))$.

Первый из этих выводов является каноническим.

**2.2. Иерархия Хомского.** Обсуждавшаяся до сих пор система — сильное описательное средство, однако при создавшемся положении вещей она является слишком общей. Тем не менее, если наложить ограничения, мы получим более интересный, хотя все еще достаточно мощный математический объект. Начальные ограничения, которые мы будем накладывать на структуру грамматики, определяют элементы $P$.

Определение (иерархия Хомского). Пусть $G = (N, T, P, S)$ является ГФС, описанной в п. 2.1. Такую грамматику называют *грамматикой Хомского типа* 0. Если все элементы $P$ получаются из формы $\alpha \to \beta$, где $\alpha = \gamma_1 x \gamma_2$, а $\beta = \gamma_1 \delta \gamma_2$, $\gamma_1, \gamma_2 \in V^*$, $x \in N$, $\delta \in V^+$, то говорят, что $G$ является контекстно-зависимой грамматикой, или *грамматикой Хомского типа* 1 (КЗГ). (В этом определении строки $\gamma_1$ и $\gamma_2$ могут рассматриваться как контекст, в котором $x$ может заменяться посредством $\delta$.)

Другим (альтернативным) ограничением для грамматики Хомского типа 1 является то, что в каждой продукции $\alpha$ и $\beta$ должны быть такими, что $1 \leqslant |\alpha| \leqslant |\beta|$. (Эквивалентность этих двух определений неочевидна и доказывается ниже.) Если подстановки могут быть выполнены

без рассмотрения контекстов, тогда мы можем заменить «контексты» $\gamma_1$ и $\gamma_2$ пустой строкой $\Lambda$ и получить более слабое ограничение: если $x \to \delta \in P$, то $x \in N$ и $\delta \in V^+$. Этому ограничению удовлетворяют *грамматики Хомского типа* 2. Наконец, если $P$ состоит только из продукций вида $x \to \delta$, где $x \in N$ и $\delta \in T \cup TN$ (так, что правая часть является или единичным терминалом, или единичным терминалом, за которым следует единичный нетерминал), то говорят, что $G$ является *грамматикой Хомского типа* 3. //

Часто бывает полезно использовать более общие формы внутри множества продукций, хотя формально это и не разрешается. Хотелось бы быть в состоянии включить пустую строку $\Lambda$ в качестве правой части любой продукции. Однако, как увидим позднее, это вызывает трудности. Такие $\Lambda$-продукции крайне необходимы с общей точки зрения, если только $\Lambda \in L$. В этом случае мы можем добавить $S \to \Lambda$ к $P$ при условии, что $S$ не встречается в правой части любой продукции. Однако в некоторых случаях необходимо разрешать также и более общие $\Lambda$-продукции. Чтобы различать грамматики Хомского и те грамматики, в которых разрешаются $\Lambda$-продукции, введем расширенные версии грамматик Хомского типа 2 и 3 — *контекстно-свободные* и *регулярные* грамматики соответственно.

Языки, порожденные каким-либо из этих типов грамматик, имеют аналогичные названия. Так, структурная грамматика порождает структурный язык, структурная грамматика Хомского типа 1 — язык Хомского типа 1, контекстно-свободная грамматика — *контекстно-свободный язык*, а регулярная грамматика порождает *регулярный язык* (или *регулярное множество*). Большинство примеров этой главы будет касаться контекстно-свободных языков, а в гл. 9 мы сконцентрируем внимание на регулярных языках. Однако большинство практических языков являются некоторыми расширениями контекстно-зависимых языков. Чтобы указать на ограничения контекстно-свободной грамматики, рассмотрим следующий важный пример.

Пример 2.4. $\{x^n y^n z^n : n \in \mathbf{N}\}$ является контекстно-зависимым языком. Предположим, что $G = (N, T, P, S)$, где

$$N = \{S, X, Y, Z\}, \quad T = \{x, y, z\}, \quad P = \{P_1, \ldots, P_7\},$$
$$P_1 = S \to xSYZ, \quad P_2 = S \to xYZ,$$

$$P_3 = xY \to xy, \qquad P_4 = yY \to yy, \quad P_5 = yZ \to yz,$$
$$P_6 = ZY \to YZ, \quad P_7 = zZ \to zz.$$

Вначале заметим, что для любого $n \in \mathbf{N}$ мы можем получить

$$S \overset{*}{\Rightarrow} x^{n-1}S(YZ)^{n-1} \Rightarrow \qquad (P_1)$$
$$\Rightarrow x^n(YZ)^n \overset{*}{\Rightarrow} \qquad (P_2)$$
$$\overset{*}{\Rightarrow} x^nY^nZ^n \Rightarrow \qquad (P_6)$$
$$\Rightarrow x^nyY^{n-1}Z^n \overset{*}{\Rightarrow} \qquad (P_3)$$
$$\overset{*}{\Rightarrow} x^ny^nZ^n \Rightarrow \qquad (P_4)$$
$$\Rightarrow x^ny^nzZ^{n-1} \overset{*}{\Rightarrow} \qquad (P_5)$$
$$\overset{*}{\Rightarrow} x^ny^nz^n; \qquad (P_7)$$

поэтому

$$\{x^ny^nz^n\colon\ n \in \mathbf{N}\} :\subseteq L(G).$$

Теперь мы должны показать, что никакие другие строки не могут быть порождены $G$. Хотя возможны некоторые изменения в порядке применения правил $(P_1)$, $(P_2)$ и $(P_6)$, любое предложение должно выводиться посредством сентенциальной формы такой, как $x^nYZ\alpha$, где $\alpha$ состоит из $n-1$ символов $Y$ и $Z$. Для того чтобы получить строку над $T$, мы должны в конце концов использовать правила $(P_4)$, $(P_5)$ и $(P_7)$, однако $(P_7)$ может преобразовать $Z$ в $z$ только в контексте $zZ$, а $(P_5)$ осуществляет такую же замену в контексте $yZ$. Аналогично для замены $Y$ на $y$ при помощи правил $(P_4)$ и $(P_8)$ требуются контексты $yY$ и $xY$ соответственно. На этой стадии подстрока $x^n$ состоит только из терминалов, поэтому на следующем шаге строка должна иметь вид $x^nyZ\alpha$ и получаться при помощи $(P_3)$. Однако мы знаем, что правильное предложение должно порождаться преобразованием из $Z\alpha$ в $Y^{n-1}Z^n$ посредством $(P_6)$. Действительно, только таким образом можно успешно получить строку.

Предположим, что мы имеем промежуточную подстроку вида $yY^mZ^pY\beta$, где $\beta$ состоит из оставшихся элементов $Y$ и $Z$. Из рассуждений, аналогичных приведенным выше, следует, что для получения подстроки $y^{m+1}Z^pY\beta$ нуж-

но $m$ раз применить $(P_4)$. Однако, если сейчас мы используем $(P_5)$ для получения $y^{m+1}zZ^{p-1}Y\beta$, то никаким правилом нельзя заменить элемент $Y$ на $y$ (или любой другой терминал). Единственный способ выйти из этого положения — это $p$ раз применить $(P_6)$, чтобы переместить $Y$ влево и, следовательно, получить $x^n y^n z^n$. //

Это пример контекстно-зависимого языка, который, как будет показано, не является контекстно-свободным. Аналогично существуют контекстно-свободные языки, которые не являются регулярными (см. гл. 9). Вернемся теперь к доказательству эквивалентности альтернативных определений контекстно-зависимых грамматик.

Определение. Грамматики $G_1$ и $G_2$ *эквивалентны*, если

$$L(G_1) = L(G_2). \text{ //}$$

Предложение. *$L$ является контекстно-зависимым языком тогда и только тогда, когда он может быть порожден грамматикой, у которой продукции $\sigma \to \mu$ удовлетворяют условию $1 \leqslant |\sigma| \leqslant |\mu|$.*

Доказательство. Если $L$ — контекстно-зависимый язык, то существует грамматика $G$ с продукциями вида $\alpha A \beta \to \alpha\gamma\beta$, где $A \in N$, $\gamma \in V^+$ и $\alpha, \beta \in V^*$ такие, что $L = L(G)$. Однако

$$|\alpha A \beta| = |\alpha| + |A| + |\beta| = |\alpha| + 1 + |\beta| \geqslant 1,$$

$$|\alpha\gamma\beta| = |\alpha| + |\gamma| + |\beta| \geqslant |\alpha| + 1 + |\beta| = |\alpha A \beta|.$$

Следовательно, $1 \leqslant |\alpha A \beta| \leqslant |\alpha\gamma\beta|$, что и требовалось доказать.

Пусть $G = (N, T, P, S)$ — грамматика, у которой продукции $\sigma \to \mu$ удовлетворяют соотношению $1 \leqslant |\sigma| \leqslant |\mu|$. Мы должны создать грамматику $G'$, эквивалентную $G$, с продукциями вида $\alpha A \beta \to \alpha\gamma\beta$.

Продукции из $G$ имеют вид

1) $A \to \gamma_1 \ldots \gamma_p$ или же

2) $\alpha_1 \ldots \alpha_n \to \beta_1 \ldots \beta_q$, где $n \leqslant q$ и $A \in N$, $\alpha_i, \beta_i, \gamma_i \in V$.

Во всех продукциях заменим каждый встречающийся элемент $a_i \in T$ новым нетерминальным элементом $A_i$ и включим продукции $A_i \to a_i$ в $G'$. Продукции типа 1) теперь имеют правильную форму и включены в $G'$. Однако продукции типа 2) необходимо модифицировать. Сейчас они имеют вид

$$W_1 \ldots W_n \to Y_1 \ldots Y_q, \quad n \leqslant q,$$

где $W_i$ и $Y_i$ являются нетерминальными символами новой грамматики. Для каждой такой продукции введем новые элементы $\widehat{Y}_1, \ldots, \widehat{Y}_q$, не являющиеся терминалами, и $n+q$ новых продукций: $n$ продукций

$$W_1 \ldots \qquad\qquad W_n \to \widehat{Y}_1 W_2 \ldots W_n,$$
$$\widehat{Y}_1 W_2 \ldots \qquad\qquad W_n \to \widehat{Y}_1 \widehat{Y}_2 W_3 \ldots W_n,$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\widehat{Y}_1 \ldots \widehat{Y}_{n-2} W_{n-1} W_n \to \widehat{Y}_1 \ldots \widehat{Y}_{n-2} \widehat{Y}_{n-1} W_n,$$
$$\widehat{Y}_1 \ldots \widehat{Y}_{n-2} \widehat{Y}_{n-1} W_n \to \widehat{Y}_1 \ldots \widehat{Y}_{n-2} \widehat{Y}_{n-1} \widehat{Y}_n \widehat{Y}_{n+1} \ldots \widehat{Y}_q$$

и $q$ продукций

$$\widehat{Y}_1 \widehat{Y}_2 \ldots \widehat{Y}_q \to Y_1 \widehat{Y}_2 \ldots \widehat{Y}_q,$$
$$Y_1 \widehat{Y}_2 \ldots \widehat{Y}_q \to Y_1 Y_2 \widehat{Y}_3 \ldots \widehat{Y}_q,$$
$$\cdots\cdots\cdots\cdots\cdots$$
$$Y_1 Y_2 \ldots Y_{q-1} \widehat{Y}_q \to Y_1 Y_2 \ldots Y_{q-1} Y_q.$$

Все эти продукции имеют вид $\alpha A \beta \to \alpha \gamma \beta$.

Новые нетерминалы $\widehat{Y}_1, \ldots, \widehat{Y}_q$ вынуждают применять эти продукции в написанном порядке так, чтобы никакие из предложений, не входящих в исходный язык, не могли быть созданы. //

В заключение этого параграфа обсудим понятие неоднозначности. Классическим примером неоднозначного предложения является предложение

«They are flying planes».

Мы имеем две интерпретации этого предложения, зависящие от того, рассматриваем мы «are flying» как сказуемое или же «flying planes» как дополнение. Это приводит нас непосредственно к точному определению неоднозначности. Язык называется *неоднозначным*, если он содержит неоднозначное предложение. Предложение является *синтаксически неоднозначным*, если оно имеет более одного канонического вывода, и *семантически неоднозначным*, если для заданного канонического вывода оно имеет более одной интерпретации. (Выводы относятся не непосредственно к языку, а к грамматике, порождающей его. Следовательно, мы должны ссылаться на *неоднозначную* грамматику; однако существуют существенно неоднозначные языки, которые могут порождаться только неоднозначными грамматиками.) Для более подробного

изучения семантических неоднозначностей рекомендуем обратиться к специальной литературе о языках программирования, а сейчас проиллюстрируем синтаксические неоднозначности двумя примерами.

Пример 2.5.

1. Пусть $G=(\{E\}, \{1, -\}, \{E \to E-E \mid 1\}, E)$. Тогда

а) $E \Rightarrow E-E \Rightarrow$
$\Rightarrow 1-E \Rightarrow$
$\Rightarrow 1-E-E \Rightarrow$
$\Rightarrow 1-1-E \Rightarrow$
$\Rightarrow 1-1-1;$

б) $E \Rightarrow E-E \Rightarrow$
$\Rightarrow E-E-E \Rightarrow$
$\Rightarrow 1-E-E \Rightarrow$
$\Rightarrow 1-1-E \Rightarrow$
$\Rightarrow 1-1-1.$

Из этих последовательностей следует, что два указанных вывода являются различными, и, следовательно, хотелось бы придать им различные значения. В примере а)

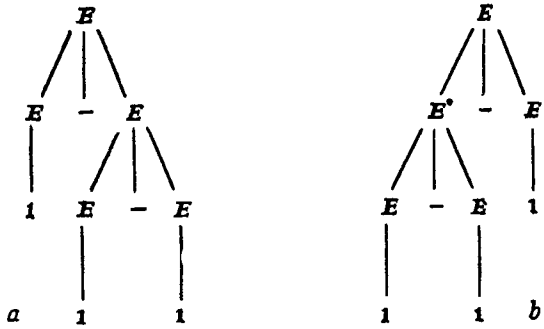



Рис. 8.2

второй знак «минус», вычисляемый вначале, дает 1; в примере б) первый знак «минус», выполняемый первым, дает −1. (Диаграммы на рис. 8.2 иллюстрируют различные структуры.

2. Рассмотрим хорошо известный пример из первой спецификации языка Алгол-60. Сужая грамматику до относящегося к делу подъязыка, будем иметь продукции

$$S \to \text{if } B \text{ then } S \text{ else } S \mid \text{if } B \text{ then } S \mid U,$$

где $S$ — утверждение, $B$ — булево выражение, $U$ — без-

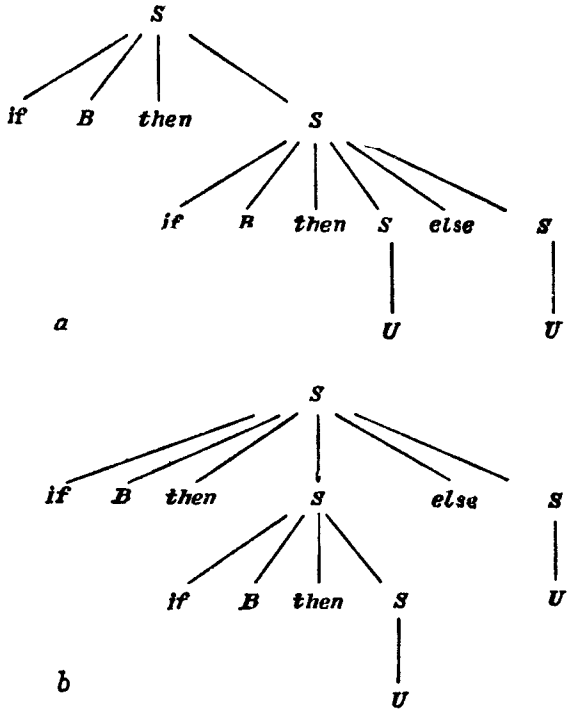



Рис. 8.3

условное утверждение. Теперь рассмотрим выражение

$$\text{if } B_1 \text{ then if } B_2 \text{ then } U_1 \text{ else } U_2.$$

Мы не знаем, принадлежит ли else $U_2$ к if $B_1$ или к if $B_2$. Формально мы можем вывести это предложение, рассматривая $B$ и $U$ как терминалы, следующим образом (рис. 8.3, *a*, *b* соответственно):

а) $S \Rightarrow \text{if } B \text{ then } S \Rightarrow$

$\Rightarrow \text{if } B \text{ then if } B \text{ then } S \text{ else } S \Rightarrow$

$\Rightarrow$ if $B$ then if $B$ then $U$ else $S \Rightarrow$

$\Rightarrow$ if $B$ then if $B$ then $U$ else $U$;

б) $S \Rightarrow$ if $B$ then $S$ else $S \Rightarrow$

$\Rightarrow$ if $B$ then if $B$ then $S$ else $S \Rightarrow$

$\Rightarrow$ if $B$ then if $B$ then $U$ else $S \Rightarrow$

$\Rightarrow$ if $B$ then if $B$ then $U$ else $U$. //

Упражнение 2.2.

1. Выразить явно языки, определенные следующими грамматиками:

а) $G = (\{\langle\text{число}\rangle, \{0, 1, 2, \ldots, 9\}, P, \langle\text{число}\rangle)$, где $P = \{\langle\text{число}\rangle \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\}$;

б) $G = (\{\langle P\rangle, \langle L\rangle, \langle D\rangle\}, \{0, 1, 2, \ldots, 9\}, P, \langle P\rangle)$, где

$$P = \{\langle P\rangle \to \langle L\rangle\langle D\rangle \mid \langle L\rangle,$$
$$\langle L\rangle \to 1 \mid 2 \mid 3 \ldots 8 \mid 9, \quad \langle D\rangle \to \langle L\rangle, \quad \langle D\rangle \to 0\}.$$

2. Определить грамматику $G' = (N', T', P', S')$, эквивалентную

$$G = (\{A, B, C, S\}, \{x, y, z\}, P, S),$$

где

$$P = \{S \to AB^2C,\ AB \to BAz,\ zB \to A^2Bx,\ A \to x,$$
$$B \to y,\ C \to z\},$$

с продукциями вида $\alpha Q \beta \to \alpha\gamma\beta$ для

$$Q \in N',\ \gamma \in (N' \cup T')^+,\ \alpha, \beta \in (N' \cup T')^*.$$

3. Определить класс Хомского грамматики, определенной следующим образом:

$$G = (\{A, B, T, S\}, \{x, y, z\}, P, S),$$

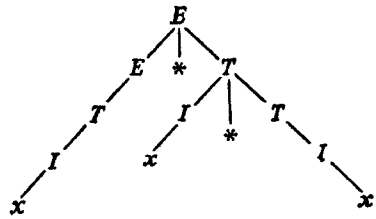
где

$$P = \{S \to xTB \mid xB,\ T \to xTA \mid xA,$$
$$B \to yz,\ Ay \to yA,\ Az \to yzz\}.$$

Используя свойство класса, к которому принадлежит $G$, установить, принадлежат или нет $L(G)$ следующие строки:

$$x^2yxz,\ x^2y^2z^2,\ xyxz.$$

4. Определить последовательность разрезов представленного здесь производящего дерева, соответствующую са́мому правому выводу предложения $x + x * x$.




5. Определить порядок в множестве продукций $P$ таким образом, чтобы была возможность канонического вывода, определяющего последовательность целых чисел в N. Продемонстрировать две такие последовательности для предложения «*aza*» в $L(G_1)$, где $G_1$ дано ниже. Вывести также строку над N, описывающую все выводы в языке $L(G_2)$, т. е. показать, что $A \overset{+}{\Rightarrow} A$ подразумевает неоднозначность. Здесь

$$G_1 = (N, T, P, E), \quad G_2 = (N, T, P, A),$$

где $N = \{A, B, C, E, R\}$, $T = \{a, d, e, x, z\}$ и

$P = \{A \to B \mid Cd,\ B \to Bx \mid eC \mid C,$

$C \to A \mid xR,\ E \to aE \mid Ea \mid R,\ R \to z\}$.

6. Выяснить, являются ли следующие грамматики неоднозначными:

а) $G = (\{A, B, S\}, \{a, b, c\}, P, S)$, где $P = \{S \to AB,\ A \to a \mid ab,\ B \to c \mid bc\}$;

б) $G = (\{\langle$целое без знака$\rangle, \langle$число$\rangle\}, D, P, \langle$целое без знака$\rangle)$, где $D = \{0, 1, \ldots, 9\}$ и

$P = \{\langle$целое без знака$\rangle \to \langle$число$\rangle,$

$\langle$число$\rangle \to \langle$число$\rangle\ \langle$число$\rangle, \langle$число$\rangle \to 0 \mid 1 \mid 2 \ldots \mid 9\}$.

## § 3. Контекстно-свободные языки

**3.1. Основные определения.** Контекстно-свободные грамматики (КСГ) и контекстно-свободные языки (КСЯ) важны для практических вычислений, так как, хотя большинство языков является некоторым расширением контекстно зависимых языков, их легче изучать как контекстно свободные языки, а затем по другим (семанти-

ческим) критериям отбросить некоторые из предложений. В этих случаях на КЗГ можно ссылаться как на грамматики, специфицированные *связанным* синтаксисом, а на КСГ — как специфицированные *несвязанным* синтаксисом. КСГ также дают возможность прояснить вопросы, содержащие (синтаксическую) неоднозначность.

Последовательность вывода $\alpha \in L(G)$ может быть изображена как упорядоченное дерево (см. ниже). Корень дерева обозначен через $S$, и если $S \overset{*}{\Rightarrow} \alpha \in T^*$ и $\alpha = a_1 \ldots a_n$, то выходы помечены по порядку $a_1, \ldots, a_n$. Предположим, что $S \overset{*}{\Rightarrow} \beta \Rightarrow \gamma \overset{*}{\Rightarrow} \alpha$ и что $\beta \Rightarrow \gamma$ достигается в результате применения продукции $C \to \gamma_1 \ldots \gamma_m$, где $\gamma_i \in V$. В дереве это представляется пометкой вершины $C$ и $m$ ее преемников (точек, из которых $C$ достигается за один шаг) $\gamma_1, \ldots, \gamma_m$. Поэтому метки могут быть одинаковыми.

Пример 3.1. Рассмотрим грамматику с продукциями

$$E \to T \mid E + T,\ T \to I \mid I * T,\ I \to (E) \mid x.$$

Обычно в случае контекстно-свободных грамматик мы будем опускать другие элементы грамматики; первое правило специфицирует источник, а нетерминалами являются только символы в левой стороне продукций. В этом случае вывод предложения $x + x * x$ может быть изображен так, как это сделано на рис. 8.4. //

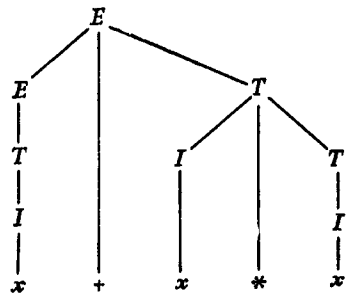


Рис. 8.4

С конструктивной точки врения это изображение называется деревом вывода. (Когда это дерево используют для того, чтобы проанализировать, могут или не могут строки содержаться в $L(G)$, оно называется *деревом грамматического разбора.*) Легко видеть, что предложение является неоднозначным, если оно имеет два неизоморфных дерева вывода, и что для контекстно-свободных грамматик канонический грамматический разбор изоморфен любой другой схеме обхода дерева.

Мы уже установили тот очевидный факт, что КСГ являются более ограниченными, чем КЗГ, но тем не ме-

нее они все же обладают весьма широкими изобразительными возможностями.

Пример 3.2. Используя контекстно-свободные правила, можно породить:

1) все последовательности из символов $A$:

$$S = AS \mid \Lambda;$$

2) все непустые списки из символов $A$, отделенные друг от друга символами $B$:

$$S \to A \mid ABS;$$

3) те же списки, что и в примере 3.2,2, однако допускается возможность пустого списка:

$$S \to T \mid \Lambda, \quad T \to A \mid ABT;$$

4) все строки, начинающиеся с последовательности символов $A$ или $B$ и оканчивающиеся символами $C$ или $D$ соответственно:

$$S \to ASC \mid BSD \mid X;$$

например,

$$S \to [S] \mid (S) \mid X.$$

В этих примерах $A$, $B$, $C$, $D$ и $X$ могут быть определены дополнительно. //

**3.2. Характеристические свойства.** Особенностью рассмотренных выше примеров, о которых вскоре мы сможем сказать несколько больше, является свойство рекурсии (один шаг рекурсии при каждом $n \in \mathbf{N}$). Трудности возникают тогда, когда требуется наложить некоторые ограничения на глубину рекурсии, не придумывая новых правил для каждой допустимой глубины рекурсии. Поскольку $N$ и $P$ конечны, то очевидно, что если не разрешать никаких рекурсий, то $L(G)$ также будет конечен, и этот случай не очень интересен.

Прежде чем идти дальше, введем необходимую терминологию. Говорят, что грамматика $G$:

а) *леворекурсивная*, если в ней имеются выводы вида

$$X \overset{*}{\Rightarrow} X\alpha, \quad \text{где} \quad X \in N, \alpha \in V^+;$$

б) *праворекурсивная*, если в ней имеются выводы вида

$$X \overset{*}{\Rightarrow} \alpha X, \quad \text{где} \quad X \text{ и } \alpha \text{ такие же, как и выше;}$$

в) *самовключающая*, если она имеет выводы вида

$$X \overset{*}{\Rightarrow} \alpha X \beta, \quad \text{где} \quad X \in N \quad \text{и} \quad \alpha, \beta \in V^+.$$

Говорят, что КСГ *рекурсивна*, если имеется один из случаев а)—в). Из сделанных выше замечаний ясно, что желательно бы иметь в грамматике «петли», однако не произвольного типа. К этому вопросу мы вернемся в § 4.

Сформулируем результат о возможностях КСГ. В теории КСЯ это, вероятно, наиболее известный результат. Его доказательство использует рекурсивные свойства КСГ и структуру деревьев. Этот результат известен как *лемма о разрастании* для КСЯ или же как *uvwxy* теорема.

Теорема. *Если $L$ — контекстно-свободный язык, то существует $n \in \mathbf{N}$ такое, что если $x \in L$ и $|z| \geqslant n$, то $z$ может быть записано в виде $uvwxy$, где $u$, $v$, $w$, $x$, $y \in T^*$, $vx \neq \Lambda$ и для любого $i \in \mathbf{N}$ выполняется условие*

$$uv^iwx^iy \in L.$$

Доказательство. Поскольку $L$ — контекстно-свободный язык, то он может быть порожден некоторой грамматикой $G = (N, T, P, S)$ и не имеет продукций, за исключением, возможно, $S \to \Lambda$ (если $\Lambda \in L(G)$), уменьшающих длину сентенциальных форм. (Если $\Lambda \in L$, то $S$ также исключают из правых частей продукций для того, чтобы не уменьшалась длина сентенциальных форм. В § 4 показано, что такая грамматика $G$ может быть найдена.)

Если $G$ не рекурсивна, то, поскольку $N$ и $P$ конечны, $L(G)$ также конечен, и, следовательно, теорема справедлива, если взять $n$ большим, чем длина самой длинной строки в $L(G)$. С другой стороны, если $G$ рекурсивна, то существует дерево вывода, в котором некоторый нетерминал, например $A$, встречается дважды на пути от корня к листу. Эта ситуация изображена на рис. 8.5. (Сюда включены лишь необходимые нам свойства.)

Более того, поскольку $G$ рекурсивна, то мы можем добиться выполнения соотношения $|uvwxy| \geqslant n$, где $n$ больше длины самого длинного предложения, полученного путем нерекурсивного вывода ($\leqslant k^m$, где $k$ — длина самой длинной продукции $P$, а $m = |N|$). Таким образом, если $z \in L$ и $|z| \geqslant n$, то $z$ должно иметь требуемый вид для некоторых пяти строк. Тогда $A \overset{+}{\Rightarrow} vAx$ ($vx \neq \Lambda$, по-

этому $|vAx| > |A|$) и $A \overset{*}{\Rightarrow} w$. Следовательно,

$$A \overset{*}{\Rightarrow} v^i A x^i \overset{*}{\Rightarrow} v^i w x^i$$

для любого $i \in \mathbf{N}$. Отсюда, так как $S \overset{*}{\Rightarrow} uAy$, имеем

$$S \overset{*}{\Rightarrow} uv^i w x^i y$$

для любого $i \in \mathbf{N}$ и, таким образом,

$$uv^i w x^i y \in L$$

для любого $i \in \mathbf{N}$. //

Этот результат может быть использован для проверки того, что некоторые конструкции в языках программирования не могут быть определены с помощью КСГ. Дадим более реальный пример, который не требует знания конкретного языка.

Пример 3.3. Грамматика из примера 2.4 порождает язык $\{x^n y^n z^n\colon n \in \mathbf{N}\}$. Сейчас мы можем показать, что

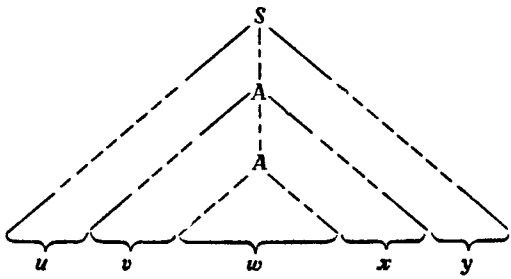



Рис. 8.5

этот язык является контекстно-зависимым и не может быть порожден КСГ. Из теоремы следует, что существует некоторое достаточно большое $n$, при котором $x^n y^n z^n$ может быть записано в виде $abcdf$ (требуется очевидная замена символов) для некоторых строк $a, \ldots, f$. Поскольку $x$ и $z$ в строке разделены, то очевидно, что строки $a$ и $b$ не могут содержать все символы $x$, $y$ и $z$; аналогично и для всех других пар из $\{a, b, c, d, f\}$. В частности, по крайней мере один из символов $x$, $y$ и $z$ не может быть одновременно в строках $b$ и $d$; таким образом, строка $ab^2cd^2f$, которая по теореме содержится в $L$, содержит не все символы $x$, $y$ и $z$ (они также могут быть распо-

ложены в другом порядке, однако в дальнейшем мы не будем рассматривать эту возможность). Следовательно, мы не получаем тот же самый язык, из чего и следует требуемый результат. //

Подобные противоречия могут быть получены во многих ситуациях, когда информация, содержащаяся в более ранней части строки, влияет на требуемую структуру последующей подстроки. Следующий пример является типичным в этом отношении.

Пример 3.4. Язык $L = \{1^p\colon p \in \mathbf{N},\ p$ — простое число$\}$ не является контекстно-свободным:

$$L = \{11,\ 111,\ 11111,\ \ldots\}.$$

Предположим, что $L$ является КСЯ. Так как существует бесконечное множество простых чисел, то имеется простое число $q$ такое, что

$$1^q = uvwxy, \qquad vx = \Lambda,$$

$$uv^i wx^i y \in L \quad \text{для всех } i \in \mathbf{N}$$

(это следует из приведенной выше леммы). Таким образом, существуют $a$, $b$, $c$, $d$, $e$ такие, что

$$1^q = 1^a 1^b 1^c 1^d 1^e$$

при $b + d > 0$ и

$$1^{q_i} = a^a \left(1^b\right)^i 1^c \left(1^d\right)^i 1^e \in L$$

для всех $i \in \mathbf{N}$, так что $q = a + b + c + d + e =$ простое число и $q > 1$, а $q_i = a + c + e + (b + d)i$ — простое число для всех $i \in \mathbf{N}$. В частности, $q_i$ — простое число при $i = a + b + c + d + e + 1$, и в этом случае

$$q_i = (a + c + e) + (b + d)(a + b + c + d + e + 1) =$$

$$= (a + c + e) + (b + d)((a + c + e) + (b + d) + 1) =$$

$$= ((a + c + e) + (b + d))(1 + (b + d)) = q(1 + b + d).$$

Однако $q > 1$ и $b + d + 1 > 1$; следовательно, $q_i$ не является простым числом для всех $i \in \mathbf{N}$, и мы получаем противоречие. Поэтому $L = \{1^p\colon p \in \mathbf{N},\ p$ — простое число$\}$ не является КСЯ. //

Этот пример также демонстрирует практическую важность связанного и несвязанного синтаксисов. Можно придумать жесткий фиксированный синтаксис, который

включает правильную семантику. Однако, где это возможно, часто гораздо удобнее и эффективнее разрешить использование более широкого языка, порожденного (обычно контекстно-свободной) грамматикой, а затем, если необходимо, сузить множество путем дальнейшей семантической проверки.

В примере 3.4 мы могли бы использовать правило

$$S \to 1S|11,$$

чтобы породить все строки $1^q$: $q > 1$, и после этого проверить, что «$q$ — простое число», с помощью подходящего арифметического алгоритма.

Короче говоря, контекстно-зависимые грамматики являются сложными и не изучены с достаточной полнотой. С другой стороны, контекстно-свободным грамматикам уделяется достаточно много внимания, и они составляют основу почти всех практических компьютерных трансляционных систем.

Упражнение 8.3.

1. Вывести КСГ, которая порождает множество всех строк над $\{a, b\}$, имеющих равное количество $a$ и $b$.

2. Построить грамматики, порождающие следующие языки:

а) $\{a^{3n}\colon n \geqslant 1\}$;

б) $\{a^n b^{2m-1}\colon n, m \geqslant 1\}$;

в) $\{a^n b^n\colon n \geqslant 1\}$, $n, m \in \mathbf{N}$.

3. Используя лемму о разрастании, показать, что язык

$$L = \left\{a^{n^2}\colon n \in \mathbf{N}\right\}$$

не является контекстно-свободным.

4. Показать, что если $L_1$ и $L_2$ являются КСЯ, то таким же является язык $L_1 \cup L_2$.

5. Доказать, что множества

$$\{x^n y^n z^m\colon n \geqslant 1,\ m \geqslant 1\},\ \{x^m y^n z^n\colon n \geqslant 1,\ m \geqslant 1\}$$

являются КСЯ; показать, что если языки $L_1$ и $L_2$ являются контекстно-свободными, то отсюда не следует, что язык $L_1 \cap L_2$ является контекстно-свободным.

## § 4. Понятия грамматического разбора и грамматических модификаций

Наиболее непосредственный и очевидный контакт, который средний пользователь имеет с процессами перевода (трансляцией с одного языка на другой),— это использование различного рода компиляторов для таких языков высокого уровня, как Паскаль, Фортран, Кобол, Алгол и др. При использовании такого языка программа, которую мы написали, транслируется в *эквивалентную* программу в машинном коде (*объектную программу*), которая может быть расшифрована и выполнена компьютером. Общая схема компиляции изображена на рис. 8.6.


Рис. 8.6

В общем случае стадии процесса компиляции могут рассматриваться связанными последовательно, как это изображено на диаграмме; однако на практике они часто выполняются одновременно. Генерация кода требует знания семантических интерпретаций, которые связаны с

каждой синтаксической структурой внутри программы. Для оптимизации машинного кода необходимо знать тонкости строения машины. Мы не будем рассматривать эти стадии, а ограничимся лишь обсуждением трансляции ключевой программы в дерево грамматического разбора.

Ключевая (исходная) программа является просто строкой символов. Внутри этой строки часто встречаются некоторые комбинации символов, в которых отдельные символы не имеют смысла, однако комбинация символов передает смысл. (См. пример 2.1; «dog» имеет значение, а буква «о» внутри «dog», очевидно, отдельно не несет смысловой нагрузки.) Такие составные символы, называемые также *лексемами*, не являются абсолютно необходимыми и могут не использоваться в некоторых языковых трансляторах, однако обычно они существуют и кодируются одним символом (для каждой комбинации свой символ), чтобы сократить длину исходной программы (на данный момент в ее лексической форме) и избежать необходимости рассматривать ненужные детали на следующих этапах. Типичными лексемами являются:

а) ключевые слова, т. е. слова с постоянным значением в языке; например,

| | | | |
|---|---|---|---|
| begin<br>end<br>while | Паскаль, | GOTO<br>DO<br>.OR. | Фортран, |

+, −, *, / в большинстве языков;

б) числа 52, 31.65 и т. п.;

в) строки или последовательности символов;

г) идентификаторы, введенные программистом.

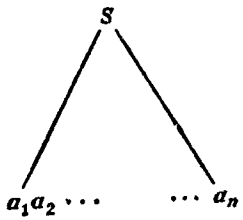


Рис. 8.7

Лексемы обычно описываются регулярными грамматиками. Следовательно, мы свели исходную проблему к грамматическому разбору строки лексем. Графически это означает — заполнить треугольник на рис. 8.7 таким образом, чтобы он был совместим с продукцией правил грамматики.

**4.1. Процедуры приведения.** В общем случае нам не разрешается изменять строку $\alpha = a_1a_2\ldots a_n$; поэтому вся деятельность до проведения процесса грамматического разбора должна быть направлена на грамматику. Потенциально нам

будет необходимо осуществить достаточно сложные преобразования грамматики, чтобы проверить, что все нетерминалы действительно можно использовать в грамматическом разборе. Существует два варианта, в которых нетерминалы могут не подходить для проведения произвольного грамматического разбора; опишем их формально.

Определение. Пусть $G=(N, T, P, S)$ есть КСГ. Тогда говорят, что нетерминальный символ $X \in N$ является:

а) *недоступным*, если $X \neq S$ и не существует вывода вида

$$S \overset{+}{\Rightarrow} \alpha X \beta \quad \text{для} \quad \alpha, \beta \in V^*;$$

б) *непродуктивным*, если не существует строки $\gamma \in T^*$ такой, что $X \overset{+}{\Rightarrow} \gamma$;

в) *бесполезным*, если он недоступен или непродуктивен.

Грамматика, не имеющая бесполезных нетерминалов, называется *редуцированной*. //

Ясно, что бесполезные символы не играют никакой роли в построении предложений. Хотя хотелось бы не включать в грамматику бесполезность символов, они могут быть введены алгоритмами, предназначенными для модификации грамматики с целью соответствия некоторым требованиям (см. ниже). Бесполезные символы не обязательно увеличивают размер грамматического разбора, и сейчас мы опишем процесс их удаления.

Пусть $G=(N, T, P, S)$ есть КСГ. Определим множество $N'$ как

$$N' = N \cup \{\tau\},$$

где $\tau$ — новый символ ($\tau \notin V$), и отношение $\rho$ на $N'$ следующим образом: $(A, B) \in \rho$, если $A \to \alpha B \beta \in P$ при $A, B \in N$, $\alpha, \beta \in V^*$; $(A, \tau) \in \rho$, если $A \to \gamma \in P$ при некотором $\gamma \in T^*$.

Предложение.

а) *$A$ доступно тогда и только тогда, когда $A = S$ или $(S, A) \in \rho^+$;*

б) *$A$ является продуктивным тогда и только тогда, когда $(A, \tau) \in \rho^+$.*

Доказательство.

а) $A$ доступно тогда и только тогда, когда существует вывод вида $S \overset{*}{\Rightarrow} \alpha A \beta$ для $\alpha, \beta \in V^*$ или, что эквивалент-

но, тогда и только тогда, когда существует $i \geqslant 0$ такое, что

$$\overbrace{S \Rightarrow \ldots \Rightarrow}^{i} \alpha A \beta.$$

Когда $S \neq A$, это имеет место лишь в случае $S \rho^+ A$; поэтому $(S, A) \in \rho^+$.

б) $A$ продуктивно тогда и только тогда, когда $A \overset{i}{\Rightarrow} \gamma$ для некоторого $i > 0$ и $\gamma \in T^*$, т. е. тогда и только тогда, когда существует последовательность сентенциальных форм $\alpha_0, \alpha_1, \ldots, \alpha_i$ таких, что

$$A \Rightarrow \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_i = \gamma,$$

т. е. когда существует последовательность $A = A_0, A_1, \ldots, A_{i-1} \in N$ такая, что $A_i$ является подстрокой $\alpha_i$, и, следовательно,

$$A_0 \rho A_1, A_1 \rho A_2, \ldots, A_{i-2} \rho A_{i-1}, A_{i-1} \rightarrow \beta,$$

где $\beta$ — подстрока $\gamma$, т. е. $A_0 \rho A_1, A_1 \rho A_2, \ldots, A_{i-1} \rho \tau$. Поэтому $A \rho^+ \tau$. //

На практике $\rho^+$ можно вычислять, используя алгоритм Уоршолла. Пусть $N_u \subset N$ — множество бесполезных символов $G$ и $N' = N \backslash N_u$, $P' = P \backslash P_u$, где $P_u$ — множество продукций, содержащих элементы $N_u$. Тогда $G' = (N', T', P', S)$, где $T'$ — множество терминальных символов, появляющихся в продукциях $P'$, эквивалентно КСГ без бесполезных символов.

Алгоритм. Удаление бесполезных символов.

Вход: КСГ $G = (N, T, P, S)$.

Выход: эквивалентная КСГ $G' = (N', T', P', S)$ без бесполезных символов.

Метод: построить $N'$, $T'$, $P'$, как указано выше. //

Пример 4.1. Рассмотрим грамматику

$$G = (\{A, B, C, D\}, \{x, y, p, q, w, a\}, P, A),$$

где

$$P = \{A \rightarrow x | yDC | D,\ B \rightarrow q | Bx,\ C \rightarrow Cx | yC,\ D \rightarrow Da | Cw | p\}.$$

Используем отношение $\rho$, определенное выше, и его представление в матричной форме:

$$M(\rho) = \begin{array}{c} \\ A \\ B \\ C \\ D \\ \tau \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & \tau \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

В этом примере имеем $M(\rho^+) = M(\rho) = M$. Таким образом, $M_{AB} = M_{C_1} = 0$, и поэтому $B$ недоступно, а $C$ непродуктивно. Следовательно, грамматика сводится к

$$G' = (\{A, D\}, \{x, a, p\}, P', A),$$

где

$$P' = \{A \to x | D, D \to Da | p\}. \ //$$

После удаления бесполезных символов каждый оставшийся нетерминальный символ $X$ встречается по крайней мере в одном дереве вывода (рис. 8.8) с $X$, связанным вверх с $S$ и вниз с некоторыми терминальными строками $a_1 \ldots a_n$.

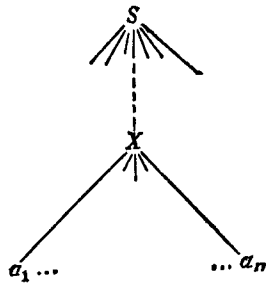


Рис. 8.8

Один «очевидный» путь грамматического разбора строки — это вывести все строки, отметить их соответствующие канонические последовательности, а затем проверить предложение, сравнивая его с каждой строкой. При совпадении использовать выводящую последовательность, чтобы определить дерево грамматического разбора. Конечно, в большинстве примеров величина $|L|$ бесконечна, и поэтому этот процесс невозможен; однако если грамматика не имеет неудачных продукций, то это приближение обеспечивает основу техники разбора, которая, по крайней мере в локальном контексте, может использоваться на практике.

Если длина сентенциальных форм не может уменьшаться при применении $G$, то при проверке $\alpha \in L(G)$, $|\alpha| = n$ можно отбросить все формы (перед получением строк над $T$), чья длина превосходит $n$. Предложения, длина которых не превосходит $n$, могут сравниваться с $\alpha$ обычным путем; в разумной грамматике все такие возможные строки должны порождаться за конечное число шагов. Сейчас мы займемся приведением грамматики и построением ее эквивалентной версии, которая обладает «более легким» грамматическим разбором.

Для того чтобы процесс порождения, связанный с данным предложением, был конечен, необходимо гарантировать, что все последовательности вывода действительно могут быть получены. Следовательно, появление таких ситуаций, как $X \to X$ и $X \to \Lambda$, представляет интерес. Записывая их непосредственно как продукции, это

легко обнаружить; однако более общие ситуации $X \overset{+}{\Rightarrow} X$ и $X \overset{+}{\Rightarrow} \Lambda$ труднее локализовать. Более того, два типа выводов связаны таким образом, как это продемонстрировано в следующем примере.

Пример 4.2. Предположим, что включенные продукции некоторой грамматики имеют вид

$$X \to Y, \quad Y \to W, \quad Z \to V, \quad W \to Z, \quad V \to X.$$

Следуя возможной последовательностью вывода из $X$, получаем, что $X \overset{+}{\Rightarrow} X$, и, следовательно, как только $X$ встречается в сентенциальной форме, мы могли бы вставить прогрессию $X \Rightarrow Y \Rightarrow W \Rightarrow Z \Rightarrow V \Rightarrow X$ (снова и снова), не получая, таким образом, ничего, кроме неоднозначности.

Как нетрудно видеть, это «петля» из нетерминалов внутри $N$. Заметим, однако, что мы могли бы иметь практически такую же ситуацию в завуалированном виде, если вместо $X \to Y$ и $Y \to W$ имели бы, например, $X \to AY$, $Y \to AWA$, $A \to \Lambda$. //

Определение. *Λ-продукцией* является продукция вида

$$X \to \Lambda, \quad X \in N.$$

КСГ $G = (N, T, P, S)$ называется *Λ-свободной*, если

а) или $P$ не имеет Λ-продукций,

б) или существует только одна Λ-продукция $S \to \Lambda$ и $S$ не появляется в правой части произвольной продукции из $P$.

Продукция является *одиночной*, если она имеет вид

$$X \to Y, \quad \text{где} \quad X, Y \in N.$$

Продукция вида $X \to X$ при $X \in N$ называется *тривиальной*. КСГ $G = (N, T, P, S)$ называется *циклически свободной*, если не существует выводов вида $X \overset{+}{\Rightarrow} X$ для любого $X \in N$. //

Как уже отмечалось, обнаружение и удаление циклов и Λ-выводов тесно связаны. Мы начнем с места расположения всех нетерминалов, из которых может быть достигнуто Λ.

Замечание. Для заданной грамматики $G = (N, T, P, S)$ через $N_\Lambda$ будем обозначать множество $\left\{X: X \overset{+}{\Rightarrow} \Lambda\right\} \subseteq N$. //

Алгоритм. Вычисление $N_A$.
Вход: произвольная КСГ $G=(N, T, P, S)$.
Выход: $N_A$.
Метод: пусть $P=\{P_1, \ldots, P_{|P|}\}$, где каждое $P_i$ имеет вид

$$\alpha_i \to \beta_i\colon\ \alpha_i \in N,\ \beta_i \in V^*;$$

тогда, рассматривая $N_A$ как «переменную» типа множества, имеем

$N_A \leftarrow \varnothing$,
$i \leftarrow |P|$,
**repeat** (**if** $(\alpha_i \notin N_A)$ **and** $(\beta_i \in N_A^*)$
**then** $(N_A \leftarrow N_A \cup \{\alpha_i\},\ i \leftarrow |P|)$
**else** $i \leftarrow i-1$)
**until** $i=0$. //

Алгоритм. Переход к Λ-свободной грамматике.
Вход: произвольная КСГ $G=(N, T, P, S)$.
Выход: эквивалентная Λ-свободная КСГ $G'=(N', T, P', S')$.
Метод:
1) определяем $N_A$;
2) строим $P'$ следующим образом:

а) Пусть $A \to \alpha_0 B_1 \alpha_1 B_2 \alpha_2 \ldots B_k \alpha_k \in P$,

где $k \geqslant 0$ и при $1 \leqslant i \leqslant k$ каждое $B_i$ есть в $N_A$, но ни один символ в $\alpha_j \in V^*$ $(0 \leqslant j \leqslant k)$ не находится в $N_A$. Тогда добавим к $P'$ продукции вида

$$A \to \alpha_0 X_1 \alpha_1 X_2 \alpha_2 \ldots X_k \alpha_k,$$

где $X_i$ есть или $B_i$ или Λ, без добавления $A \to \Lambda$ к $P'$ (это могло бы иметь место, если бы все $\alpha_i$ совпали с Λ).

б) Пусть $S \in N_A$. Тогда добавим к $P'$ продукции

$$S' \to \Lambda | S,$$

где $S'$ — новый символ, и тогда $N'=N \cup \{S'\}$; в противном случае $N'=N$ и $S'=S$. //

Сейчас мы можем рассмотреть удаление циклов из КСГ. Месторасположение циклов может быть легко найдено выделением отношения $\rho=\{(A, B)\colon A \to B \in P\}$ и формированием замыкания $\rho^+$. Тогда ясно, что произвольное $X$: $X\rho^+X$ должно быть в цикле. Объединим это вместе со схемой «обратной замены», которая удаляет

все нетерминалы внутри произвольного цикла. (Она также удаляет любую тривиальную продукцию.)

Алгоритм. Переход от КСГ к эквивалентной циклически свободной грамматике.

Вход: Λ-свободная КСГ $G=(N, T, P, S)$.

Выход: эквивалентная циклически свободная грамматика

$$G'=(N', T', P', S').$$

(Нетерминалы $G$ переименованы: $A_1$ на $A_n$, где $n=|N|$, $S$ на $A_1$, а каждую продукцию $P_i$ выражают через $\alpha_i \to \beta_i$.) Дополнительно мы используем множество INCYCLES, а $n$ нетерминалов обозначаем через REPLACE$i$, где $1 \leqslant i \leqslant n$. Алгоритм будет иметь следующий вид:

1) определяем $\rho$ над $\mathbf{N}_n$ так, что $i\rho j$ тогда и только тогда, когда $A_i \to A_j \in P$;

2) пусть $\sigma = \rho^+$;

3) INCYCLES $\leftarrow \varnothing$;

4) **for** $i$ **from** 1 **to** $n-1$
**do** (**for** $j$ **from** $i+1$ **to** $n$
**do if** ($j \notin$ INCYCLES **and**
$i\sigma j$ **and**
$j\sigma i$ )
**then** (INCYCLES $\leftarrow$ INCYCLES $\cup \{j\}$,
REPLACE$j \leftarrow A_i$))

5) $j \leftarrow 0$
**for** $i$ **from** 1 **to** $|P|$
**do** (**for all** $k \in$ INCYCLES
in $P_i$ replace $A_k$ by REPLACE$k$
giving new $P_i$
if (new $P_i \notin \{P'_1, \ldots, P'_j\}$
and $\alpha_i \neq \beta_i$ in new $P_i$)
then ($j \leftarrow j+1$, $P'_j \leftarrow$ new $P_i$))

6) $G'=(N|\text{INCYCLES}, T, P', S)$. //

Говорят, что КСГ является приведенной, если она Λ-свободна, циклически свободна и редуцирована. Полу-

чив собственную КСГ $G$, мы можем использовать ее для проверки условия $\alpha \in L(G)$ для данного $\alpha \in T^*$.

**4.2. Модификации грамматического разбора.** Как было установлено во введении к § 4, задача грамматического разбора строки состоит в заполнении треугольника вывода (рис. 8:7) с соответствующим деревом. Конечно, в большинстве случаев это заполнение нельзя разумно выполнить за один шаг, и обычно оно получается при помощи последовательности поддеревьев. Эти последовательности поддеревьев могут быть получены многими способами; три наиболее используемых способа изображены на рис. 8.9.

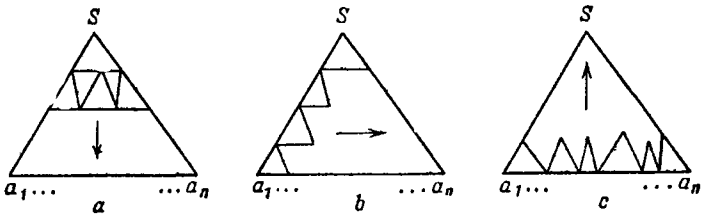



Рис. 8.9

Стратегия грамматического разбора, изображенная на рис. 8.9, *a*, называется грамматическим разбором сверху вниз. В нем применяют продукции (в некотором выбранном порядке) к сентенциальным формам, пытаясь расширить $S$ внутрь строки $a_1 \ldots a_n$. При таком разборе естественно использовать в качестве гида часть строки $(a_1 \ldots a_n)$, чтобы управлять выводом. Для практических рассмотрений, согласующихся с чтением $a_1 \ldots a_n$ слева направо (рис. 8.9, *b*) и желанием начать разбор перед включением полной строки, необходимо использовать начало строки. Следовательно, в поиске сентенциальных форм, которые начинаются с $a_1 \in T$ (и соответственно исследуя терминальные строки как начальные подстроки последовательных частей входной строки), мы должны отвергнуть возможность выводов вида $X \overset{+}{\Rightarrow} X\beta$ $(X \in N, \beta \in V^+)$. Таким образом, чтобы начать разбор сверху вниз, мы должны удалить левую рекурсию.

В общем случае это может быть сделано с использованием процесса, аналогичного решению системы линейных алгебраических уравнений. Часто возможно рассматривать одну рекурсию за один раз и удалять ее, ис-

пользуя довольно простое тождество. Оправданием такого подхода служит порождаемый язык. Рассмотрим $X \overset{+}{\Rightarrow} X\alpha$. Путем «обратной подстановки» правых частей продукций мы можем получить прямую рекурсию как элемент $P$; таким образом, имеем $X \to X\alpha \mid \beta$. Рассматривая это как полную грамматику над $\{\alpha, \beta\}$ ($\beta$ представляет все другие нелеворекурсивные возможности для $X$), очевидным образом получаем, что

$$L(X) = \{\beta\alpha^n\colon\ n = 0 \text{ или } n \in \mathbf{N}\}.$$

Это множество может порождаться также продукциями

$$X \to \beta Y, \quad Y \to \alpha Y \mid \Lambda,$$

которые не являются леворекурсивными (они праворекурсивны). Чтобы закончить преобразование, продукция $X \to \beta Y$ должна быть расширена при необходимости до правильного числа термов.

Пример 4.3. Рассмотрим грамматику с продукциями

$$A \to Bc \mid dC,\ B \to xA \mid Ce,\ C \to Ab \mid w.$$

Она леворекурсивна, поскольку

$$A \Rightarrow Bc \Rightarrow Cec \Rightarrow Abec.$$

Путем обратной подстановки для $C$ в $B$ и $B$ в $A$ получаем

$$B \to xA \mid (Ab \mid w)e \equiv B \to xA \mid Abe \mid we,$$

$$A \to BC \mid dC \equiv A \to (xA \mid Abe \mid we)c \mid dC \equiv$$

$$\equiv A \to xAc \mid Abec \mid wec \mid dC \equiv A \to A\underbrace{bec}_{\alpha} \mid \underbrace{xAc \mid wec \mid dC}_{\beta}.$$

Поэтому, используя преобразование

$$A \to \beta Y, \quad Y \to \alpha Y \mid \Lambda,$$

получаем

$$A \to (xAc \mid wec \mid dC)Y \equiv A \to xAcY \mid wecY \mid dCY,$$

$$Y \to bec\,Y \mid \Lambda,\ B \to xA \mid Ce,$$

$$C \to Ab \mid w. \ //$$

Заметим, что в этом примере $B$ может быть «вырезано» и может не принимать никакого участия в произвольном предложении, порожденном из корня $A$ (следовательно, надо удалить все следы $B$ из грамматики).

Отметим также, что мы ввели Λ-продукцию. Эта продукция, созданная как побочный эффект «преобразования грамматического разбора», вызвала бы меньше проблем, чем аналогичная продукция, встречающаяся естественным образом. Символ $B$ может быть удален при помощи алгоритма для удаления бесполезного символа.

Хотя обычно будет достаточно частичного удаления отдельных леворекурсивных цепей внутри $N$, мы должны использовать матричное обобщение описанного процесса для того, чтобы справиться с удалением внутренних левых рекурсий.

Напомним, что в обычном случае мы заменяем $X \to X\alpha \mid \beta$ при помощи $X \to \beta Y$, $Y \to \alpha Y \mid \Lambda$. Если мы имеем $n$ нетерминалов $X_1, \ldots, X_n$, которые являются взаимно леворекурсивными, так что не сводятся путем последовательности обратных замен к простому случаю, тогда мы можем представить соответствующие продукции схематически следующим образом:

$$X_1 \to X_1A_{11} \mid X_2A_{21} \mid \ldots \mid X_nA_{n1} \mid B_1,$$
$$X_2 \to X_1A_{12} \mid X_2A_{22} \mid \ldots \mid X_nA_{n2} \mid B_2,$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$X_n \to X_1A_{1n} \mid X_2A_{2n} \mid \ldots \mid X_nA_{nn} \mid B_n,$$

где каждое $A_{ij}$ представляет собой остаток всех операций, которые могут быть выведены из $X_j$ и которые начинаются с $X_i$, и аналогично каждое $B_j$ представляет все альтернативы для нетерминалов $X_j$, которые не начинаются с элементов $\{X_1, \ldots, X_n\}$.

Теперь, поскольку $A_{ij}$ и $B_j$ являются множествами строк, отсюда следует, что:

1) если $X_j \to X_i$, то $\Lambda \in A_{ij}$;

2) если $X_j \to \alpha$ и $\alpha \neq X_i\beta$ для $\beta \in V^*$, то $\alpha \in B_j$;

3) если $X_j \not\to X_i\gamma$ для любого $\gamma \in V^*$, то $A_{ij} = \varnothing$.

Следовательно, над алгебраической системой $(V^*, \odot, |)$ мы можем свести эти продукции к матричной схеме

$$\mathbf{X} = \mathbf{XA} \mid \mathbf{B} \text{ в } \mathcal{M}(n, (V^*, \odot, |)),$$

или, записывая альтернативный оператор | как +, к схеме

$$\mathbf{X} = \mathbf{XA} + \mathbf{B} \text{ в } \mathcal{M}(n, (V^*, \odot, +)).$$

По аналогии с простым (нематричным) случаем, о кото-

ром будет более подробно сказано в гл. 9, скажем, что

$$\mathbf{X} = \mathbf{BY},$$

где $\mathbf{Y} = \mathbf{AY} + \mathbf{I}$, а $\mathbf{I}$ определяется на $(V^*, \odot, +)$ как

$$\mathbf{I}_{ij} = \begin{cases} \{\Lambda\}, & \text{если } i = j, \\ \varnothing, & \text{если } i \neq j. \end{cases}$$

Пример 4.4. Предположим, что $S = D$ и $G$ имеет следующие продукции:

$$D \to Dx \mid Ey \mid Fz, \quad E \to Da \mid Fc,$$
$$F \to Dp \mid Eq \mid Fr \mid w.$$

Таким образом, используя общую схему, получаем

| | $X_j \to X_1 \; A_{1j}$ | $X_2 \; A_{2j}$ | $X_3 \; A_{3j}$ | $B_j$ |
|---|---|---|---|---|
| $X_1 =$ | $D \to Dx$ | $Ey$ | $Fz$ | |
| $X_2 =$ | $E \to Da$ | | $Fc$ | |
| $X_3 =$ | $F \to Dp$ | $Eq$ | $Fr$ | $w$. |

Итак, $X_i = \sum\limits_k B_k Y_{ki}$. Поэтому

$$X_1 = D \to \sum_k B_k Y_{k1} = B_3 Y_{31} = w Y_{31},$$

$$E \to w Y_{32}, \quad F \to w Y_{33}, \quad Y_{ij} = \sum_k A_{ik} Y_{kj} + I_{ij};$$

следовательно,

$$Y_{11} \to xY_{11} \mid aY_{21} \mid pY_{31} \mid \Lambda,$$
$$Y_{12} \to xY_{12} \mid aY_{22} \mid pY_{32},$$
$$Y_{13} \to xY_{13} \mid aY_{23} \mid pY_{33},$$
$$Y_{21} \to yY_{11} \mid qY_{31},$$
$$Y_{22} \to yY_{12} \mid qY_{32} \mid \Lambda,$$
$$Y_{23} \to yY_{13} \mid qY_{33},$$
$$Y_{31} \to zY_{11} \mid cY_{21} \mid rY_{31},$$
$$Y_{32} \to zY_{12} \mid cY_{22} \mid rY_{32},$$
$$Y_{33} \to zY_{13} \mid cY_{23} \mid rY_{33} \mid \Lambda.$$

В этом примере преобразования производят ненужные нетерминалы; удаляя их, получаем

$$D \to wY_{31},$$

$$Y_{31} \to zY_{11} \mid cY_{21} \mid rY_{31},$$
$$Y_{11} \to xY_{11} \mid aY_{21} \mid pY_{31} \mid \Lambda,$$
$$Y_{21} \to yY_{11} \mid qY_{31}.$$

Изменяя соответствующим образом имена, получаем

$$D \to wJ,$$
$$J \to zK \mid cL \mid rJ,$$
$$K \to xK \mid aL \mid pJ \mid \Lambda,$$
$$L \to yK \mid qJ.$$

Чтобы нагляднее показать степень трансформации, приведем здесь дерево грамматического разбора строки «*wcqzayx*» для первоначальной грамматики (рис. 8.10, *а*) и модифицированной (рис. 8.10, *b*) грамматик.

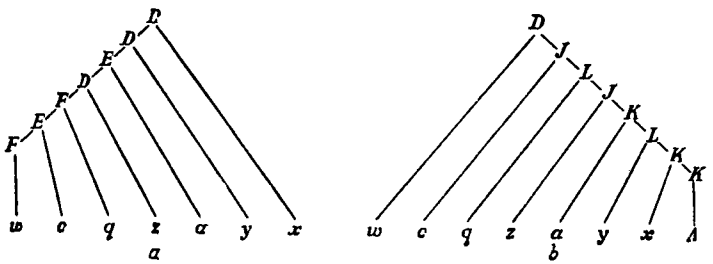



Рис. 8.10

Удаление левых рекурсий обеспечивает, если это возможно, вывод строки, начинающейся с требуемого терминального символа; однако это не гарантирует, что мы получим только одну такую строку, и, следовательно, может случиться, что мы пойдем неправильным путем.

Пытаясь избежать неправильных последовательностей грамматического разбора, мы можем явно использовать манипуляции, уже встречавшиеся при предыдущих преобразованиях. Этот процесс называют *левой факторизацией*. Процесс требует, чтобы были проведены продукции, включающие в левую часть данный нетерминал. Затем расширяют все правые части и те, которые начинаются с общей подстроки над $T$, собирают вместе.

Пример 4.5.

$$A \to xyB \mid xyBC \mid yPQ \mid yxV \equiv$$
$$\equiv A \to xyB(\Lambda \mid C) \mid y(PQ \mid xV) \equiv A \to xyBA_1 \mid yA_2,$$

где $A_1 \to \Lambda \mid C$, обычно записываемое как $C \mid \Lambda$, и $A_2 \to \to PQ \mid xV$. //

Заметим, что мы вновь можем ввести Λ-продукции. Однако если в настоящий момент нет Λ-продукций, то в заключительных левофакторизованных продукциях и грамматике не будет левых рекурсий. В этом случае следующий символ входной строки можно использовать для того, чтобы непосредственно определить, какие альтернативы надо использовать для расширения нетерминалов в сентенциальную форму. Если Λ-продукции встречаются в явном виде, то это вызывает затруднения.

В силу того что Λ является ведущей подстрокой каждой строки над произвольным алфавитом, она всегда совпадает с началом строки задания, и, следовательно, никакие последующие альтернативы никогда не будут рассматриваться. Здесь нет возможности входить в полный анализ проблемы, однако заметим, что если $G = (N, T, P, S)$ и мы определяем

а) $\text{FIRST}(\alpha) = \{x\colon\ \alpha \overset{*}{\Rightarrow} x\beta,\ x \in T, \beta \in V^*\}$

и

б) $\text{FOLLOW}(\alpha) = \{x\colon\ S \overset{*}{\Rightarrow} \gamma\alpha x\delta,\ x \in T,\ \gamma\delta \in V^*\}$,

и если для каждой продукции $X \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_n$

а) $\text{FIRST}(\alpha_i) \cap \text{FIRST}(\alpha_j) = \varnothing,\ i \neq j$

и

б) $X \overset{*}{\Rightarrow} \Lambda$,

то

$$\text{FIRST}(X) \cap \text{FOLLOW}(X) = \varnothing.$$

В этом случае $G$ можно использовать для предсказывающего анализа (см. рис. 8.9, *б*). В таком анализе можно проверять альтернативы в произвольном порядке, не применяя Λ-выводов, пока все другие возможности не исчезли. Следующий пример иллюстрирует этот процесс.

Пример 4.6. Предположим, что единственной продукцией в грамматике является

$$C \to xCx \mid \Lambda.$$

Попытаемся провести грамматический разбор строки «*xx*». Строка отбрасывается, хотя она и законна, потому что мы вынуждены применить первую продукцию дваж-

ды, порождая таким образом неправильное продвижение, поскольку $x \in \mathrm{FIRST}(C) \cap \mathrm{FOLLOW}(G)$ и $C \to \Lambda$. Это графически изображено на рис. 8.11.

Использование грамматики $C \to xxC \mid \Lambda$ (рис. 8.12) не вызывает никаких трудностей в грамматическом разборе, потому что сейчас $x \notin \mathrm{FOLLOW}(C)$. //

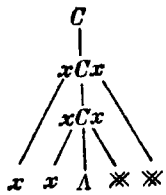



Рис. 8.11

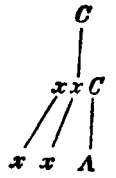



Рис. 8.12

Перед тем как завершить параграф, упомянем другой основной метод грамматического разбора — *снизу вверх*, в котором продукции применяют *назад*, пытаясь свести строку задания к $S$ (см. рис. 8.9, *с*). Этот метод применяют более широко по сравнению с методом сверху вниз; используя некоторые модификации этого метода, можно повысить эффективность грамматического разбора.

Упражнение 8.4.

1. Модифицировать грамматику, продукции которой даны ниже, таким образом, чтобы она не была леворекурсивной:

$$A \to Bx \mid Cz \mid w, \quad B \to Ab \mid Bc, \quad C \to Ax \mid By \mid Cp.$$

2. Пусть $G = (N, T, P, S)$ является КСГ и символ $A \in N$ не является бесполезным символом $G$. Показать, что существование одного из следующих выводов в $G$ влечет за собой неоднозначность $G$:

а) $A \overset{+}{\Rightarrow} A\gamma A$;

б) $A \overset{+}{\Rightarrow} \alpha A \mid A\beta$;

в) $A \overset{+}{\Rightarrow} \alpha A \mid \alpha A \beta A$;

г) $A \overset{+}{\Rightarrow} A$

при $\gamma \in (N \cup T)^*$ и $\alpha, \beta \in (N \cup T)^+$.

3. Определить Λ-свободную КСГ, эквивалентную КСГ и определенную как

$$G = (\{S\}, \{a, b\}, P, S),$$

где

$$P = \{S \to aSbS \mid bSaS \mid \Lambda\}.$$

4. Пусть

$$G = (\{A, B, C, D, E, S\}, \{a, b, c\}, P, S),$$

где

$$P = \{S \to A \mid B,\ A \to C \mid D,\ B \to D \mid E,\ C \to S \mid a \mid \Lambda,$$
$$D \to S \mid b,\ E \to S \mid c \mid \Lambda\}.$$

Найти приведенную грамматику, эквивалентную $G$.

## § 5. Грамматики операторного предшествования

Важное подмножество КСГ содержит в себе так называемые *операторные грамматики*. Это грамматики, в которых все продукции такие, что никакие два нетерминала не являются смежными в любой правой части, и, следовательно, лежащий между ними терминал можно представить как оператор (хотя не обязательно в арифметическом смысле). Попытаемся определить отношения предшествования на множестве $T \cup \{\vdash, \dashv\}$, где $\vdash$ и $\dashv$ суть новые символы, которых нет в $V$ и которые ограничивают «предложение». Правила определим следующим образом:

1. $a \doteq b$, если $A \to \alpha a \beta b \gamma \in P$; здесь $\alpha$, $\gamma \in V^*$ и $\beta \in N \cup \{\Lambda\}$.

2. $a \lessdot b$, если $A \to \alpha a B \beta \in P$; здесь $B \overset{+}{\Rightarrow} \gamma b \delta$, $\gamma \in N \cup \{\Lambda\}$ и $\alpha$, $\beta$, $\delta \in V^*$.

3. $a \gtrdot b$, если $A \to \alpha B b \beta \in P$; здесь $B \overset{+}{\Rightarrow} \gamma a \delta$, $\delta \in N \cup \{\Lambda\}$ и $\alpha$, $\beta$, $\gamma \in V^*$.

4. $\vdash \lessdot a$, если $S \overset{+}{\Rightarrow} \alpha a \beta$, $\alpha \in N \cup \{\Lambda\}$, $\beta \in \Lambda^*$.

5. $a \gtrdot \dashv$, если $S \overset{+}{\Rightarrow} \alpha a \beta$, $\beta \in N \cup \{\Lambda\}$, $\alpha \in V^*$.

Символы $\lessdot$, $\doteq$ и $\gtrdot$ обозначают отношения предшествования (читается как «имеет меньшее старшинство, чем», «имеет такое же старшинство, как», «имеет большее старшинство, чем»); при условии что не более од-

ного такого отношения справедливо между двумя произвольными операторами из $T \cup \{\vdash, \dashv\}$, соответствующую операторную грамматику называют *грамматикой операторного предшествования*.

Хотя она и является гораздо более сложной, чем другие виды грамматик, встречавшихся до сих пор, понятие предшествования может быть введено так, что будет совпадать с обычным старшинством арифметических операторов и будет расширено до операторов, действия которых важны с точки зрения вычислений, однако обычно считаются само собой разумеющимися при вычислениях «на бумаге».

Пример 5.1.

$$E \to E+T \mid T, \quad T \to T*P \mid P, \quad P \to (E) \mid x.$$

Для этой грамматики отношения предшествования приведены в виде таблицы на рис. 8.13. //

| | $+$ | $*$ | $($ | $)$ | $x$ | $\dashv$ |
|---|---|---|---|---|---|---|
| $\vdash$ | $\lessdot$ | $\lessdot$ | $\lessdot$ | | $\lessdot$ | |
| $+$ | $\gtrdot$ | $\lessdot$ | $\lessdot$ | $\gtrdot$ | $\lessdot$ | $\gtrdot$ |
| $*$ | $\gtrdot$ | $\gtrdot$ | $\lessdot$ | $\gtrdot$ | $\lessdot$ | $\gtrdot$ |
| $($ | $\lessdot$ | $\lessdot$ | $\lessdot$ | $\doteq$ | $\lessdot$ | |
| $)$ | $\gtrdot$ | $\gtrdot$ | | $\gtrdot$ | | $\gtrdot$ |
| $x$ | $\gtrdot$ | $\gtrdot$ | | $\gtrdot$ | | $\gtrdot$ |

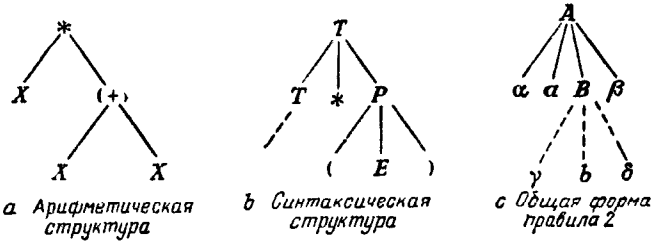
Рис. 8.13

Для того чтобы увидеть, что происходит в действительности, рассмотрим этап внутри вывода предложения $\vdash x*(x+x)\dashv$. Из правила 2 определений предшествования видно, что для символов $*$ и $($ имеем

$$T \to T*P, \quad P \overset{+}{\Rightarrow} (E).$$

Таким образом, выполняется отношение $* \lessdot \cdot ($, и поэтому поддерево $P$ должно быть вычислено перед вычислением $T*P$; следовательно, действие, связанное с «(», которым является удаление этой и парной к ней закрывающей скобки, выполняется перед действием, обозначенным «*». (Графически ситуацию можно представить так, как это сделано на рис. 8.14. Здесь для правила 2 имеем $A \equiv T$, $\alpha \equiv T$, $a \equiv *$, $B \equiv P$, $\beta \equiv \Lambda$, $\gamma \equiv \Lambda$, $b \equiv ($ и $\delta \equiv E$. Отсюда видно, что основная структура грамматик

операторного предшествования является простой и естественной, однако выглядит сложной при записи из-за общности правил.) Заменяя $x$ целыми числами 2, 3 и 4,




*a* Арифметическая структура *b* Синтаксическая структура *c* Общая форма правила 2

Рис. 8.14

получаем $\vdash 2*(3+4)\dashv$. Записывая отношения предшествования под этим выражением, видим, как определяется порядок вычислений

$$\begin{array}{ll}\vdash & 2 \quad *;\\ & <\cdot \ \cdot>:\end{array}$$

а) выберем число 2 и сохраним его в стеке:

$$\begin{array}{ll}\vdash & * \quad ( \quad 3 \quad +,\\ & <\cdot \ <\cdot \ <\cdot \ \cdot>;\end{array}$$

б) аналогично удалим 3 из выражения и поместим в стек:

$$\begin{array}{ll}\vdash & * \quad ( \quad + \quad 4 \quad ),\\ & <\cdot \ <\cdot \ <\cdot \ \ <\cdot \ \cdot>;\end{array}$$

в) с 4 поступим подобным образом:

$$\begin{array}{ll}\vdash & * \quad ( \quad + \quad ),\\ & <\cdot \ <\cdot \ <\cdot \ \ \cdot>;\end{array}$$

г) выполним сложение двух верхних элементов в стеке и результат оставим там же; удалим символ +:

$$\begin{array}{ll}\vdash & * \quad ( \ ),\\ & <\cdot \ <\cdot \doteq;\end{array}$$

д) отбросим скобки:

$$\begin{array}{ll}\vdash & * \quad \dashv,\\ & <\cdot \ \cdot>;\end{array}$$

е) произведем умножение двух верхних элементов стека, оставляя результат в стеке; удалим символ *:

$$\vdash\ \dashv;$$

ж) останов из-за отсутствия отношений предшествования; ответ находится в стеке.

Конечно, вместо выполнения арифметических операций мы могли бы породить код и после этого вычислить выражение — это как раз то, что сделал бы компилятор.

Упражнение 8.5.

1. В проведенном ранее обсуждении привлекаемая семантика принималась само собой разумеющейся, однако она была тесно переплетена с грамматической структурой. Показать, что каждая из следующих грамматик является грамматикой операторного предшествования, и исследовать, как ее внутренняя семантика отличается от обычных соглашений:

$$P_1 = \{E \to E * T \mid T,\ T \to T + P \mid P,\ P \to (E) \mid x\},$$

$$P_2 = \{E \to T + E \mid T - E \mid T,\ T \to T * P \mid P,\ P \to (E) \mid x\}.$$

ГЛАВА 9

# КОНЕЧНЫЕ АВТОМАТЫ

Автоматом является устройство, управляющее и контролирующее само себя. Обычный компьютер с программой способен при достаточном запасе энергии контролировать сам себя и, следовательно, является автоматом. Как таковые, компьютеры изучались много лет, однако более естественно рассматривать программу и машину, в которой она содержится, как отдельные компоненты.

Конечно, для выполнения вычислений нам нужна не только программа, но и машина, на которой эти вычисления могут выполняться. Однако здесь мы не стремимся приступать к детальному изучению теории вычислений, поэтому, за исключением тех случаев, где это необходимо для полноты, мы ограничим наше внимание математическим описанием некоторых конечных машин.

Несмотря на эти замечания, мы начнем (в § 1) с общего введения, которое показывает границы того, что машины могут выполнять. Далее (§ 2) мы изучим математические модели устройств (обычно их малые фрагменты), а затем (в § 3) связанную с этим алгебру.

## § 1. Общие понятия

Все используемые на практике компьютерные устройства ограничены (некоторым образом) количеством информации, которую они могут хранить,— они конечны. Цель данного параграфа — показать, как можно делать утверждения о программах без утомительного рассмотрения синтаксических деталей; затем мы продемонстрируем, что даже при отсутствии ограничения на размеры памяти существуют задачи, которые нельзя решить.

**1.1. Универсальная машина.** Несмотря на использование многочисленных различных типов данных и множеств символов внутри реальных программ, для теоретического изучения достаточно ограничиться рассмотрением программ, которые действуют на множестве $V = \mathbf{N} \cup \cup \{0\}$. Это равносильно изучению программ, вычисляю-

щих теоретико-числовые функции, которые будут введены в § 2; наша непосредственная задача — описать идеализированный компьютер, позволяющий запоминать элементы $V$, с которыми можно осуществлять преобразования, и дать детальное описание того, как могут быть представлены программы для машины.

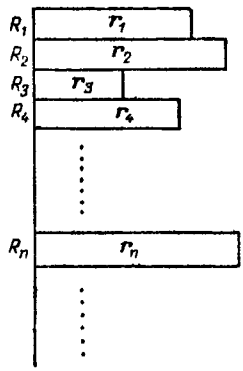


Рис. 9.1

Предположим, что машина имеет память, состоящую из неограниченного числа регистров: $R_1$, $R_2$, $R_3$ и т. д., и что содержание каждого регистра $R_i$ есть $r_i \in V$. Это можно изобразить так, как это сделано на рис. 9.1.

Для более ясного изложения удобно разрешить использование многих операций, действующих над регистрами, однако на самом деле необходимы лишь две операции:

$$R_n \leftarrow R_n + 1, \quad R_n \leftarrow R_n - 1.$$

Обозначая содержимое регистра $n$ до операции и после через $r_n$ и $r_n'$ соответственно, можно описать результаты выполнения этих операций следующим образом:

$$R_n \leftarrow R_n + 1 \equiv r_n' = r_n + 1,$$

$$R_n \leftarrow R_n - 1 \equiv r_n' = \begin{cases} r_n - 1, & \text{если } r_n > 0, \\ 0, & \text{если } r_n = 0. \end{cases}$$

Кроме операций, изменяющих значения регистров памяти, необходимо, чтобы машина имела связь с программой для того, чтобы влиять на ее работу. Необходимой в этом случае является только одна операция, а именно та, которая осуществляет сравнение $R_n = 0$. Однако, как и прежде, мы будем разрешать более общие формы операций такого рода. Результат $R_n = 0$ справедлив тогда и только тогда, когда $r_n = 0$, и это условие используется для управления программами. Множество регистров вместе с описанными выше определениями называется *машиной с неограниченной памятью* (МНП).

Программы состоят из конечной совокупности операций, занумерованных от 1 до некоторого $n \in \mathbf{N}$. Мы не будем вдаваться в детали, а выведем более общие заключения об этих программах. Поэтому не будем давать формального определения такой структуры. Рисунков 9.2 и

9.3 достаточно, чтобы понять, какого рода конструкции допустимы в ней.

Программа на рис. 9.2 не предназначена для выполнения особо разумных вычислений, однако она указывает

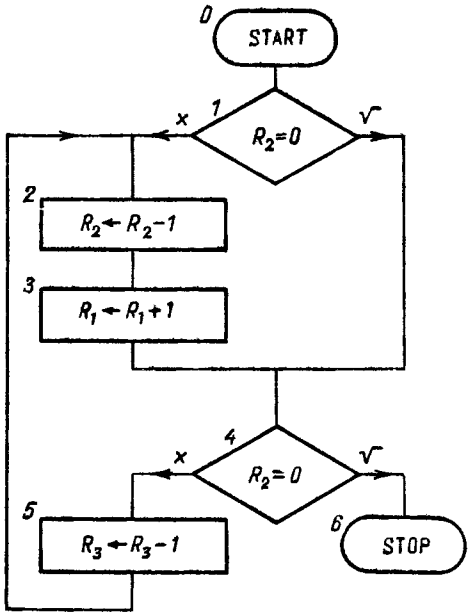



Рис. 9.2

вид блок-схемы программы для нашей машины. Заметим, что все это может быть записано не обязательно в виде рисунка. Например, можно написать:

**1: если $R_2 = 0$ то перейти к 4 иначе перейти к 2**

**2: $R_2 \leftarrow R_2 - 1$ (затем перейти к 3)**

**3: $R_1 \leftarrow R_1 + 1$ (затем перейти к 4)**

**4: если $R_2 = 0$ то перейти к 6 иначе перейти к 5**

**5: $R_3 \leftarrow R_3 - 1$ (затем перейти к 2)**

**6: STOP**

На рис. 9.3 представлена более общая форма программы, в которую включены *макрокоманды* $F_i$ и проверки $T_i$. Это могут быть стандартные команды того типа, который уже определен, или же они могут быть представлены последовательностью основных команд, кото-

рым для удобства чтения даны имена (подобно подпрограмме), детали которых уже где-то определены. Такие последовательности будем называть макропоследовательностями.

Сейчас мы можем описать некоторые из этих макропоследовательностей, которые обеспечат связь с последующими темами, а также помогут убедить читателя,

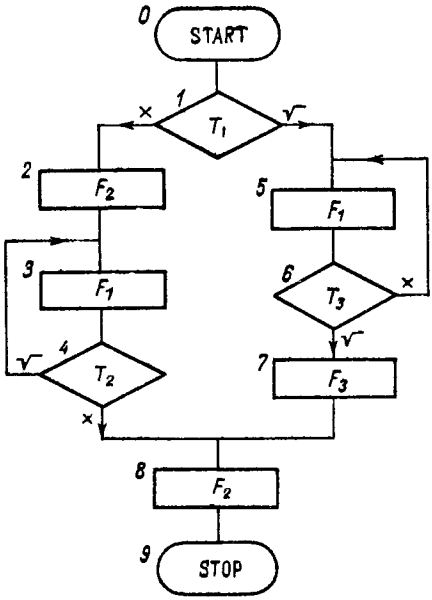



Рис. 9.3

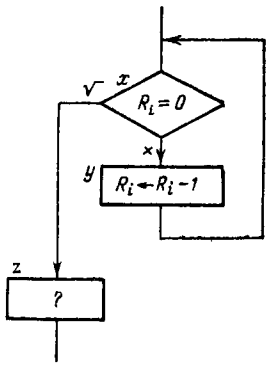



Рис. 9.4

что наше небольшое число простых команд на самом деле является достаточно мощным средством.

Пример 1.1. $R_i \leftarrow 0$ может быть реализовано следующим фрагментом программы, где метки $x$, $y$ и $z$ выбраны так, чтобы не пересекаться с другими командами в программе:

$x$: **если** $R_i = 0$ **то перейти к** $z$ **иначе перейти к** $y$

$y$: $R_i \leftarrow R_i - 1$ (**затем перейти к** $x$)

$z$: ?

В терминах блок-схем этот фрагмент программы изображен на рис. 9.4.

Пример 1.2. Сейчас, определяя макрокоманду, удовлетворяющую предыдущему примеру, и включая в явном виде выражения «go to» («перейти к») только тог-

да, когда мы уклоняемся от выполнения команды «go to next instruction» («перейти к следующей команде»), мы можем дать раскрытие формулы $R_i \leftarrow m$ (для некоторого $m \in \mathbf{N}$):

$$\left.\begin{array}{l} R_i \leftarrow 0 \\ R_i \leftarrow R_i + 1 \\ R_i \leftarrow R_i + 1 \\ \ldots\ldots \\ R_i \leftarrow R_i + 1 \end{array}\right\} m \text{ раз}$$

Очередное расширение множества команд требует использования «рабочей памяти». Мы будем предполагать, что по крайней мере в простейших случаях читатель в состоянии придумать подходящую стратегию работы с памятью, и, следовательно, не будем специально упоминать о том, как выбираются эти дополнительные регистры. Случаи, когда количество требуемой памяти неизвестно (такие, как стеки и т. п.), будут рассмотрены ниже*).

Пример 1.3. Используя $R_k$ в качестве рабочего регистра, мы можем скопировать содержимое $R_j$ в $R_i$ ($R_i \leftarrow R_j$). Делая это, мы уничтожаем содержимое $R_i$, которое в дальнейшем должно быть восстановлено. Следующая программа осуществляет требуемые вычисления:

$R_k \leftarrow 0$
$x$: **if** $R_j = 0$ **then go to** $y$
$R_k \leftarrow R_k + 1$
$R_j \leftarrow R_j - 1$ **then go to** $x$
$y$: $R_i \leftarrow 0$
$w$: **if** $R_k = 0$ **then go to** $z$
$R_i \leftarrow R_i + 1$
$R_j \leftarrow R_j + 1$
$R_k \leftarrow R_k - 1$ **then go to** $w$
$z$: //

Сейчас мы вернемся к «собственно» арифметическим вычислениям. Сложение и вычитание строятся непосредственно, однако, поскольку в регистрах существуют огра-

*) В дальнейшем будем использовать символы: «go to $z$»—«перейти к $z$»; «if ⟨условие⟩ then ⟨оператор⟩»—«если ⟨условие⟩, то ⟨оператор⟩»; «else»—«иначе»; «then go to $z$»—«затем перейти к $z$».— *Примеч. пер.*

ничения на значения величин, операция вычитания должна быть несколько модифицирована. Подобным же образом можно выполнить умножение и усеченное деление.

Пример 1.4. Сложение $R_i \leftarrow R_i + R_k$, имеющее результатом $r_i = r_i + r_k$, может быть выполнено следующим образом:

$R_j \leftarrow R_k$

$x$: if $R_j = 0$ then go to $y$

$R_i \leftarrow R_i + 1$

$R_j \leftarrow R_j - 1$ then go to $x$

$y$:

Чтобы получить полную программу в терминах основных команд, мы должны расшифровать макрокоманду «$R_j \leftarrow R_k$», как это было сделано в примере 1.3. С этого времени мы не будем требовать доказательства таких расшифровок.

Подобным образом «ограниченное вычитание» $R_i \leftarrow R_i - R_k$ может быть выполнено как

$R_j \leftarrow R_k$

$x$: if $R_j = 0$ then go to $y$

$R_j \leftarrow R_j - 1$

$R_i \leftarrow R_i - 1$ then go to $x$

$y$:

Заметим, что если первоначальные значения $R_i$ и $R_k$ были такие, что $r_i < r_k$, то после $r_i$ итераций операция $R_i \leftarrow R_i - 1$ не будет иметь эффекта.

Аналогично $R_i \leftarrow R_i * R_k$ может быть представлено как

$R_j \leftarrow 0$

$R_l \leftarrow R_k$

$x$: if $R_l = 0$ then go to $y$

$R_l \leftarrow R_l - 1$

$R_j \leftarrow R_j + R_i$ then go to $x$

$y$: $R_i \leftarrow R_j$ //

В большинстве случаев таким же способом, каким может быть расширено множество операций над регистрами путем определения макрокоманд, можно также ввести ко-

манды сравнения, которые на первый взгляд оказываются более сложными, однако в действительности строятся из последовательностей стандартных операций и операций сравнения.

Пример 1.5. Мы можем записать операцию «if $R_i > R_k$ then go to $x$ else go to $y$» («если $R_i > R_k$, то перейти к $x$; иначе перейти к $y$») следующим образом:

$R_j \leftarrow R_i$

$R_j \leftarrow R_j - R_k$

**if** $R_j = 0$ **then go to** $y$ **else go to** $x$

Этими операциями условного перехода мы можем дополнить наше множество первоначальных арифметических операций вместе с операцией деления «$R_i \leftarrow R_i \div R_k$, где $r_i = 0$, если $r_k = 0$». В нашем случае этой операции соответствует следующий алгоритм:

$R_l \leftarrow 0$

**if** $R_k = 0$ **then go to** $x$

$y$: **if** $R_i < R_k$ **then go to** $x$

$R_l \leftarrow R_l + 1$

**if** $R_i = R_k$ **then go to** $x$

$R_i \leftarrow R_i - R_k$ **then go to** $y$

$x$: $R_l \leftarrow R_l$

Из менее общего, но необходимого приложения, которое кратко приведено ниже, следуют две операции, связанные с точностью усечения при делении целых чисел.

Пример 1.6. Если $r_i \neq 0$, операция «если $R_k$ — делитель $R_i$, то перейти к $x$; иначе перейти к $y$» действует следующим образом:

**if** $R_k = 0$ **then go to** $y$

$R_j \leftarrow R_i$

$R_j \leftarrow R_j \div R_k$

$R_j \leftarrow R_j * R_k$

**if** $R_j = R_i$ **then go to** $x$ **else go to** $y$

При помощи этой (возможно, несколько странной) операции и операции сравнения вида $R_i = m$ (оставленной в качестве упражнения) мы можем проверить, является ли $r_i$ простым числом.

Ясно, что можно операцию «если $R_i$ простое, то перейти к $x$; иначе перейти к $y$» смоделировать следующим образом:

**if** $R_i = 0$ **then go to** $y$

**if** $R_i = 1$ **then go to** $y$

$R_j \leftarrow R_i - 1$

$z$: **if** $R_j = 1$ **then go to** $x$

**if** $R_i$ делитель $R_j$ **then go to** $y$

$R_j \leftarrow R_j - 1$ **then go to** $z$ //

Перед последним примером этого параграфа мы должны упомянуть, как в машине вводятся данные и выводится результат. Предположим, что мы хотим вычислить значение теоретико-числовой функции $f$: $V^n \to V^m$. Значения $n$ и $m$ известны перед началом выполнения соответствующей программы; поэтому мы можем вначале выделить $n$ регистров (в которые должны быть загружены начальные значения перед началом выполнения программы) для входных данных и $m$ регистров (которые не должны быть обязательно отличными от выбранных вначале) для выходных данных. Когда программа останавливается, мы предполагаем, что некоторый «внешний представитель» может выдать «ответы» из соответствующих мест. Это, конечно, разумный путь моделирования потоков данных в программе, поскольку точно показывает взаимодействие операций ввода-вывода. С этими соглашениями пример 1.7 может рассматриваться или как полная программа (в которой $R_i$ и $R_j$ выбраны как входные и выходные регистры), или как схема для подпрограммы.

Пример 1.7. Последовательность команд приведенных ниже, помещает $n$-е простое число в $R_j$, где $n$ является содержимым $R_i$; предполагается, что $n$ не равно нулю:

(Start)

$R_k \leftarrow R_i - 1$

$R_j \leftarrow 2$

$x$: **if** $R_k = 0$ **then go to** $y$

$R_j \leftarrow R_j + 1$

$z$: **if** $R_j$ простое **then go to** $w$

$R_j \leftarrow R_j + 1$ **then go to** $z$

$w$: $R_k \leftarrow R_k - 1$ then go to $x$

$y$: (stop) //

Сейчас мы объясним наш очевидный интерес к простым числам.

**1.2. Кодирование программ.** Программы п. 1.1 могли работать с элементами из $V = \mathbf{N} \cup \{0\}$. Мы должны объяснить, как в принципе можно любую программу рассматривать в качестве программы такого типа. Это можно сделать, описав способы, при помощи которых различные типы данных могут быть закодированы в элементы из $V$. Для этого рассматриваем данные предложения над подходящими алфавитами и, следовательно, почти как постороннее следствие этого процесса получаем также метод для кодирования предложений на языках программирования, а именно сами программы.

Основное математическое средство, используемое для этой цели, это *теорема о единственности разложения*, известная также как основная теорема арифметики. Эта теорема устанавливает, что произвольный элемент из $V$ является или 0, или 1 либо может быть выражен единственным образом как произведение упорядоченных простых чисел. Ясно, что если $n \in V \setminus \{0, 1\}$ и

$$n = q_1 * q_2 * \ldots * q_i = s_1 * s_2 * \ldots * s_j,$$

где $q_1, \ldots, q_i, s_1, \ldots, s_j$ — все простые числа такие, что

$$q_1 \leqslant q_2 \leqslant \ldots \leqslant q_i,\ s_1 \leqslant s_2 \leqslant \ldots \leqslant s_j,$$

то $i = j$ и $s_k = q_k$ для всех $k$, $1 \leqslant k \leqslant i$. Напомним, что простые числа — это элементы из $\mathbf{N}$, которые делятся только на 1 и на самих себя.

Полное доказательство этой теоремы несложно, но его запись существенно отвлекла бы нас от основной задачи. Поэтому вместо доказательства мы предлагаем рассмотреть конструкцию алгоритма (процедуры/программы) для выделения упорядоченных простых множителей $q_1, \ldots, q_i$ для любого данного $n$. Таким образом, избегая деталей ввода и вывода, мы можем использовать схему, представленную на рис. 9.5.

Предположим теперь, что некоторая программа читает данные, которые состоят из последовательности символов алфавита $A = \{x, y, z\}$. Произвольно выбирая порядок $\varphi$ для элементов из $A$, можно получить $\varphi(1) = x$, $\varphi(2) = y$, $\varphi(3) = z$. Если вводимая последовательность $\alpha$ имеет длину $n$ и выражается как $\alpha_1\alpha_2 \ldots \alpha_n$, где $\alpha_i \in A$,

тогда существует последовательность

$$\varphi^{-1}(\alpha_1)\varphi^{-1}(\alpha_2)\ldots\varphi^{-1}(\alpha_n)$$

над {1, 2, 3}. Взяв первые $n$ простых чисел $p_1, \ldots, p_n$, мы можем составить число

$$\prod_{i=1}^{n} p_i^{\varphi^{-1}(\alpha_i)} = p_1^{\varphi^{-1}(\alpha_1)} * \ldots * p_n^{\varphi^{-1}(\alpha_n)};$$

назовем его $\Phi(\alpha)$. Пусть по соглашению $\Phi(\Lambda) = 1$.

Чтобы проиллюстрировать использование этой общей формулы, закодируем строку «*xyxz*» посредством 1, 2, 3

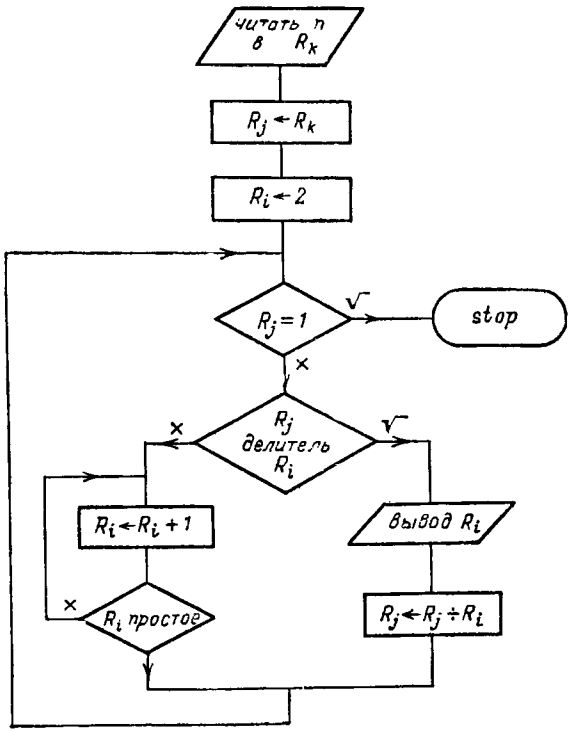



Рис. 9.5

следующим образом: $2^1 3^2 5^1 7^3 = 30\,870$. Используя теорему о единственности разложения на простые множители и тот факт, что $\Phi^{-1}$ является биекцией, имеем, что «*xyxz*» является единственной строкой над $A$, которая дает это значение. Следовательно, применяя $\Phi$, мы можем

«обратить» вычисления, чтобы восстановить строку «$xyxz$».

При помощи этой процедуры все строки над $A$ можно закодировать таким образом, чтобы они имели разные значения в множестве $V$. Это в силу упорядоченности, индуцируемой $\mathbf{Z}$ (поскольку $V \subseteq \mathbf{Z}$), влечет упорядоченность строк из $A^*$. Например, «$xy$»>«$yx$».

Необходимо отметить два фактора, связанных с этим методом. Во-первых, поскольку $A$ конечно, в $\mathbf{N}$ существуют значения, которые не являются кодами какой-либо строки в $A^*$ (например, не существует $\alpha \in A^*$ такого, что $\Phi(\alpha) = 2^5$, так как иначе $a \in A$: $\varphi^{-1}(5) = a$, а такого $a$ не существует). Во-вторых, поскольку элементы $A^*$ являются неограниченными, то таковыми являются и коды. (Для любого $n \in \mathbf{N}$ возьмем простое $p_m > n$ и рассмотрим строку

$$\alpha = \alpha_1 \ldots \alpha_m \ldots \alpha_q$$

такую, что $|\alpha| \geqslant m$; тогда отсюда следует, что

$$\Phi(\alpha) = \prod_{i=1}^{q} p_i^{\varphi^{-1}(\alpha_i)} \geqslant p_m^{\varphi^{-1}(\alpha_m)} \geqslant p_m^1 = p_m > n.)$$

Используя методику, описанную в § 3 гл. 3, преобразуем кодирование $\Phi$ так, чтобы получить новое кодирование, которое сохраняет порядок, полученный $\Phi$, но использует все $V$. Это один из тех случаев, когда трудно дать *арифметическую* формулу для вычисления измененных кодов, однако все еще достаточно легко описать способ их получения.

Очевидно, что новое кодирование $\Psi$ с областью значений $V$ задается формулой $\alpha \mapsto |\{\beta\colon \Phi(\beta) < \Phi(\alpha)\}|$. В частном случае для рассмотренного в примере множества $A$ и его упорядочения $\varphi$ первые десять значений $\Phi$ и $\Psi$ приведены в табл. 9.1.

Таблица 9.1

| Строка $\alpha$ | $\Lambda$ | $x$ | $y$ | $xx$ | $z$ | $yx$ | $xy$ | $zx$ | $xxx$ | $yy$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $\Phi(\alpha)$ | 1 | 2 | 4 | 6 | 8 | 12 | 18 | 24 | 30 | 36 |
| $\Psi(\alpha)$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Чтобы получить эти коды и выполнить процедуры кодирования, требуется машина, способная понимать и создавать символы вне алфавита $D = \{0, 1, 2, \ldots, 8, 9\}$, на котором определены элементы множества $V$. Для этого

необходимо только устройство, позволяющее имитировать действие $\varphi$ (и $\varphi^{-1}$), некоторым простым способом воздействуя на таблицу ввода-вывода; остаток процедуры кодирования, включающий $\Phi$ и $\Psi$, можно потом учесть при помощи машины с неограниченной памятью.

Следовательно, любой ввод в данную компьютерную систему может рассматриваться как конечная строка (взятая из потенциально бесконечного множества строк над некоторым конечным алфавитом), и если применить описанный выше процесс к соответствующему алфавиту $A$, то эту строку можно преобразовать в единственный элемент $V$. Обратная процедура, примененная к полученному значению из $V$, дает значение над другим алфавитом $B$.

Следовательно, программа $P\colon A^* \to B^*$ похожа на соответствующую программу $P'\colon V \to V$, где

$$P\colon \alpha \mapsto \Phi^{-1}(P'(\Phi(\alpha))), \quad P'\colon n \mapsto \Phi\left(P\left(\Phi^{-1}(n)\right)\right).$$

Величина используемых значений даже в простых ситуациях делает примеры непригодными, однако связанные

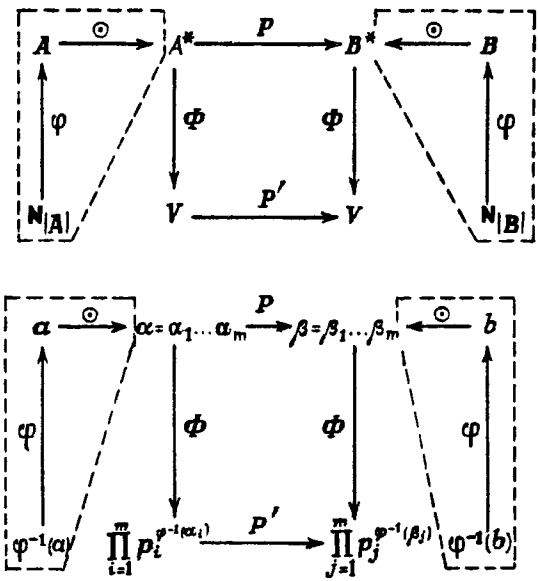



Рис. 9.6

диаграммы на рис. 9.6 восстанавливают сущность пропущенных этапов.

Переходы, включающие детальную спецификацию $P'$, являются сложными и не обязательно соответствуют $P$ «очевидным» образом. Однако если $P$ вычислимо, то и $P'$ вычислимо.

Как отмечалось выше, мы можем также применять такие же кодирующие процедуры к программам, и, следовательно, расширив $\Psi$ таким образом, чтобы можно было игнорировать семантически неверные программы так же, как и программы, включающие синтаксические ограничения, можно перечислить все программы для данной системы, использующей числа из $V$. Мы предполагаем, что для того, чтобы программа превосходила любое заданное в $V$ число, к ней можно добавить неограниченное множество кодов, которые содержат произвольно много нулевых утверждений, таких, как $R_i \leftarrow R_i$. Однако в большинстве языков программирования существует много избыточности в синтаксисе; поэтому в лучшем случае мы будем обращать внимание только на лексические знаки. Прежде чем пытаться сделать замечания общего характера, рассмотрим, что можно сделать с простым языком, который использовался ранее в машине с неограниченной памятью. Существует четыре типа команд:

$x$: $R_i \leftarrow R_i + 1$ **then go to** $y$

$x$: $R_i \leftarrow R_i - 1$ **then go to** $y$

$x$: **if** $R_i = 0$ **then go to** $y$ **else go to** $z$

$x$: stop

(Без потери общности мы можем предполагать, что все программы начинаются с команды, помеченной индексом 1.) Все, что мы должны знать из команды $x$,— это, какой тип операции должен выполняться, на каком регистре и какая команда выполняется следующей. Для того чтобы можно было использовать ту же самую схему для всех команд, введем формат

(тип, регистр, правильный выход, неправильный выход).

Обозначая команды «останов» («stop»), «больше», «меньше» и «равно нулю» через 0, 1, 2 и 3 соответственно, получаем Ф-уровневое кодирование типичных утверждений, как показано ниже:

$\Phi(R_i \leftarrow R_i + 1$ **then go to** $y) = 2^1 3^i 5^y 7^y$

$\Phi(R_i \leftarrow R_i - 1$ **then go to** $y) = 2^2 3^i 5^y 7^y$

$\Phi($**if** $R_i = 0$ **then go to** $y$ **else go to** $z) = 2^3 3^i 5^y 7^z$

$\Phi(\text{stop}) = 2^0 3^0 5^0 7^0 = 1.$

Поэтому команда Φ содержит всю информацию, необходимую для ее выполнения или, если необходимо, для перехода к следующему шагу выполнения программы.

Поскольку все строки в $A^*$ конечны, а блок-схема программы для машины с бесконечной памятью сверху не ограничена, то каждая отдельная программа имеет $n$ утверждений, $n \in \mathbf{N}$. Тогда можно распространить Φ на программы следующим образом:

$$\Phi(\text{prog}) = \prod_{i=1}^{n} p_i^{\Phi(s_i)},$$

где «prog» состоит из $n$ помеченных утверждений, из которых $i$-м является $s_i$.

Из Φ(prog) можно получить $\Phi(s_i)$ выделением компоненты $p_i$, и, следовательно, разлагая это значение на простые сомножители, мы можем узнать детали утверждения, помеченного номером $i$.

Используя подходящую «урезающую» функцию Ψ, мы получаем кодирование, которое является биекцией между $\mathscr{P}$, множеством всех программ для машины с бесконечной памятью и $V$.

**1.3. Проблема останова.** Теперь мы можем доказать, что конструкции некоторых общих программ невозможны: не только потому, что мы не находим решения, но и потому, что решение может не существовать. Формальные доказательства ограничим рассмотрением двух основных случаев. Первое из доказательств получим, исходя из начальных принципов, а затем покажем, как второе выводится из первого, иллюстрируя, таким образом, общее использование техники сведения одних задач к другим. После этого сформулируем перечень проблем, относительно которых будет показано, что они неразрешимы подобным образом.

Строго говоря, все утверждения, приведенные ниже, относятся к программам на машине с бесконечной памятью со специальным кодированием функций $\Psi_1$ и $\Psi_2$. Однако, задав произвольную «универсальную» машину и подходящий язык программирования, мы в состоянии построить адекватные кодирующие функции, и, следовательно, полученные результаты применимы и в общем случае.

Первой задачей, относительно которой мы докажем ее неразрешимость, является задача *самоприменимости*.

Теорема. *Для машины с бесконечной памятью не существует программы, которая для любой заданной про-*

*граммы $A$ при ее кодировании $a = \Psi_1(A)$ будет останавливаться с выходным значением* 0, *если $A(a)$ останавливается (т. е. программа $A$ останавливается, если входное значение равно $a$), и с выходным значением* 1, *если $A(a)$ не останавливается.*

Доказательство. Будем строить доказательство от противного. Предположим, что такая программа существует; назовем ее $B$. Итак, $B(a)$ останавливается с результатом, равным 0, если $a$ приводит программу $A$ к останову, и с результатом, равным 1, если $a$ не приводит к останову $A$.

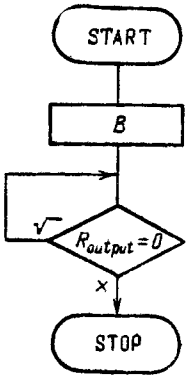


Рис. 9.7

Изменим программу $B$ следующим образом. Заменим команду останова условной петлей, так что, если выходной регистр имеет значение 0, мы обходим эту команду; в противном случае — останов. Назовем эту программу $C$ (рис. 9.7). Имеем следующее: $C(x)$ останавливается (и имеет то же самое выходное значение, что и $B(x)$) тогда и только тогда, когда выходное значение равно 1.

Рассматривая $C$ при входном значении $c = \Psi(C)$, получаем противоречие: так как $C(c)$ останавливается тогда и только тогда, когда $C(c) = B(c) = 1$, то $B(c) = 1$ тогда и только тогда, когда $C(c)$ не останавливается. Отсюда следует, что $C$ не может существовать, а поэтому и $B$ не существует. //

Предыдущее доказательство по стилю подобно тому, которое использовалось, чтобы опровергнуть существование множества Рассела в примере 1.1 гл. 1. Кажется, что оно носит тривиальный характер, однако, подобно многим другим кратким математическим рассуждениям, является достаточно тонким. Следует на это обратить внимание перед тем, как использовать этот факт при доказательстве более общего результата — неразрешимости проблемы останова.

Теорема. *Для машины с бесконечной памятью не существует программы, которая по произвольным входным данным, представляющим программу и ее данные, будет определять, останавливается программа на этих данных или нет.*

Доказательство. Покажем, что если бы такая программа существовала, то мы могли бы путем выбора правильных входных данных решить проблему самопри-

менимости. Поскольку последнее невозможно, то отсюда будет следовать, что решение проблемы останова также невозможно.

Очевидно, что программа требует ввода двух сортов данных: $a(=\Psi_1(A))$ — кода программы $A$ и $x(=\Psi_2(X))$, где $X$ — входные данные для $A$. Это может быть закодировано обычным путем — с использованием двух различных простых чисел $p$ и $q$. Пусть входные данные будут $p^a q^x$, и предположим, что существует программа для решения нашей задачи; назовем ее $B$. Тогда $B(p^a q^x)$ останавливается с результатом 1, если $A(X)$ останавливается, и останавливается с результатом 0, если $A(X)$ не останавливается. В этом случае из $B$ можно создать новую программу $C$, добавляя к началу $B$ последовательность команд, которые превращают содержимое $i$-го входного элемента в $(pq)^i$. Тогда

$$C(a) = B(p^a q^a) = \begin{cases} 1, & \text{если } A(a) \text{ останавливается,} \\ 0, & \text{если } A(a) \text{ не останавливается.} \end{cases}$$

Следовательно, $C$ решает проблему самоприменимости, о которой мы знаем, что она неразрешима. Следовательно, $C$ не существует, а поэтому не существует и $B$. //

Проблема останова является классическим результатом о неразрешимости в теории компьютеров. Приведенное здесь доказательство использует общую технику сведения одной задачи к другой. Далее показывается, что первая задача неразрешима, поскольку если бы это было не так, то была бы разрешима и другая задача, о которой известно, что она неразрешима.

Используя этот принцип (и соответствующие конструкции, детали которых здесь опускаются), можно показать, что многие другие проблемы также неразрешимы. В частности, неразрешимы следующие проблемы:

а) останавливается или нет произвольная программа, если входное значение равно 0;

б) останавливается или нет произвольная программа при любых входных данных;

в) выполнено ли равенство $L(G) = \varnothing$ для произвольной контекстно-зависимой грамматики $G$;

г) выполнено ли равенство $L(G_1) \cap L(G_2) = \varnothing$, где $G_1$ и $G_2$ — произвольные контекстно-свободные грамматики;

д) выполнено ли равенство $L(G) = T^*$, где $G = (N, T, P, S)$ — произвольная контекстно-свободная грамматика;

е) выполнено ли равенство $L(G_1)=L(G_2)$, где $G_1$ и $G_2$ — произвольные контекстно-свободные грамматики;

ж) является ли произвольная контекстно-свободная грамматика неоднозначной.

**1.4. «Расширенная» машина.** Насколько реалистичной является машина с бесконечной памятью в качестве модели компьютера? Память такой машины, очевидно, превосходит память любой существующей машины, которая подвергается ограничениям как на число регистров, так и на значения, которые могут содержаться в каждом регистре. Однако эта идеализация в пределах машины с неограниченной памятью расширяет, а не ограничивает возможности машины. Существуют ли аспекты в компьютерных системах, которые не могут быть смоделированы на машине с неограниченной памятью?

Мы утверждаем, что нет. Хотя формально мы не можем доказать это высказывание, остановимся на основных моментах такой «расширенной» машины, которая может моделировать все нужные свойства, хотя в качестве абстрактной модели она применяется редко.

Нас будут интересовать следующие свойства:

а) большой набор операций;

б) более широкий набор внешних типов данных;

в) массивы;

г) стеки;

д) более сильные команды управления;

е) общий механизм управления;

ж) рекурсия.

В пп. 1.1 и 1.2 мы показали, как в случае а) можно добавить арифметические операции к множеству команд, а в случае б), добавляя простые периферийные устройства (для того чтобы иметь возможность работать с символами, которых нет в машине с неограниченной памятью), можно расширить диапазон представлений входных и выходных данных над произвольным алфавитом.

Остановимся на работе с массивами. Предположим, что нам необходим массив с десятью компонентами $A[1], \ldots, A[10]$ и что содержимое этих регистров суть $a_1, \ldots, a_{10}$. С помощью методики, которая уже использовалась, мы можем сохранить эти значения, вводя величину

$$a = \prod_{i=1}^{10} p_i^{a_i},$$

которую затем можно запомнить в одном регистре. Мы

уже описывали процедуру выделения каждого $a_i$ из $a$. Этот метод не зависит от границы массива и поэтому переносится на случай работы со стеками. В стеках можно хранить результаты промежуточных вычислений, не требуя существенно больше регистров. С помощью подходящего механизма управления (см. ниже) мы можем также поместить в стек адреса возврата из подпрограмм.

Любая программа на машине с неограниченной памятью конечна, и строки программы могут быть пронумерованы (помечены) числами от 1 до $n$, $n \in \mathbf{N}$. В контексте блок-схемы программы мы можем сконструировать в случае е) «вычисление **go to**», используя табличную технику, данную выше, для того, чтобы получить «**go to** $(R_k)$», как указано на рис. 9.8.

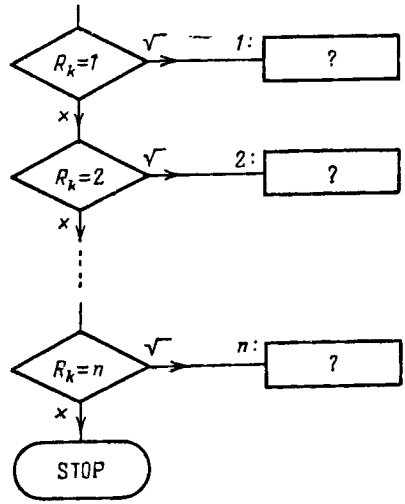


Рис. 9.8

Используя схему прямого кодирования для программ Φ (а не Ψ), мы можем закодировать всю программу одним значением, которое запоминается в регистре. Разложение на простые множители затем может использоваться для выделения кода «следующего утверждения». Это моделирует свойство ж), а вместе с использованием стеков дает возможность рекурсии з).

Следовательно, используя машину с неограниченной памятью, можно произвести «расширение» путем добавления этих свойств. Однако в действительности такие расширения носят лишь внешний характер, а результирующая система может быть смоделирована с использованием другой машины с неограниченной памятью, у которой блок-схема программы состоит из простых команд увеличения, уменьшения и «равенства нулю».

Упражнение 9.1.

1. Построить на машине с неограниченной памятью блок-схему программы, осуществляющей проверку «$R_i = m$».

2. Спроектировать машину с неограниченной памятью, которая, если дана программа, закодированная в $R_0$, будет выполнять эту программу. (За основу взять кодирующую схему из п. 1.2.)

## § 2. Конечные автоматы

Уже достаточно много сказано об универсальных машинах (их свойствах и ограничениях на эти свойства), которые могут вычислять *все*, что вычисляется, и, следовательно, проводя рассуждения в обратном порядке, получим, что любая вычислительная проблема, которая неразрешима в такой общей системе, будет неразрешима и в *любой* другой системе. Однако возникает вопрос: как это связано с реальными компьютерами? Основная проблема — это конечность памяти реальной машины (правда, и ее можно несколько сгладить, используя машину с неограниченной памятью, описанную в п. 1.1). Машина с неограниченной памятью состояла из неограниченного числа регистров, каждый из которых имел возможность содержать любое число из множества $V$. Хотя, как это следует из построений п. 1.4, мы можем обменять «короткую и широкую» память (имеющую несколько регистров, каждый с большой емкостью) на «длинную и тонкую» память (в которой емкость каждого регистра уменьшена, зато число регистров увеличено), существенное ограничение состоит в конечности числа различных конфигураций или состояний, которые может принимать память. Такое ограничение вызвано физическими ограничениями. Это понятие формирует основу нашей математической модели конечных машин.

**2.1. Детерминированные машины.** В соответствии с опытом электронного машиностроения (низким уровнем аппаратуры при построении элементов типа «не и» и т. п., микропроцессорных систем или универсальных цифровых машин) наша первая модель воплощает в себе понятие детерминизма, т. е. если некоторая ситуация достигается более одного раза, то в каждом случае устройство ведет себя одинаковым образом. Начнем с формального описания.

Определение. *Конечным (детерминированным) автоматом* $M$ называется алгебраическая структура

$$M = (Q, \Sigma, t, q_0, F, p),$$

где $Q$ — непустое *множество состояний*; $\Sigma$ — конечный *входной алфавит*; $t$ — отображение $Q \times \Sigma \to Q$, называемое *переходом* (или *функцией переходов*); $q_0 \in Q$ — *начальное состояние*; $F \subseteq Q$, $F$ — множество *заключительных состояний* (или *принимающих состояний*); $p$ — функция $Q \times \Sigma \to \Sigma$, называемая *функцией печати* (или *функцией выходов*).

(В некоторых случаях полезно модифицировать функцию выходов таким образом, чтобы она имела вид $p$: $Q \times \times \Sigma \to \Sigma'$, где $\Sigma'$ — некоторый другой алфавит.)

Идея состоит в том, что мы начинаем из состояния $q_0$, и если $q_i \in t(q_0, s)$, то под действием входного символа $s$ автомат переходит в состояние $q_i$. Аналогично, если $(q_0, s) \in D_p$, то, когда автомат перейдет в состояние $(q_0, s)$, на выходе появится $p(q_0, s)$. Продолжая таким образом и читая каждый раз очередной входной символ, будем переходить от одного состояния к следующему, пока или не прочитаем символ, которого нет в $\Sigma$, или входные данные будут исчерпаны — в этих случаях обработка прекращается.

Входная последовательность называется представимой (автоматом $M$), если состояние, в которое перешел автомат $M$, принадлежит $F$. Для наглядности рассмотрим пример. Однако прежде опишем представление $M$ в виде диаграммы.

Во-первых, мы представим элементы $Q$ вершинами ориентированного графа, которые изображаются маленькими кругами; имя состояния указывается внутри круга. Элементы из $F$ имеют дополнительные круги, начерченные вокруг маленьких кругов. Если $((q_i, s_j), q_k) \in t$, то проведем ориентированное ребро от $q_i$ к $q_k$ и пометим его символом $s_j$. Далее, если $((q_i, s_j), s_l) \in p$, то пометим ребро через $s_j$: $s_l$. К одному и тому же ребру может быть добавлено несколько меток. Наконец, определим $q_0$ стрелкой, входящей в $q_0$.

Пример 2.1. Рассмотрим машину, изображенную на рис. 9.9. Предположим, что мы читаем строку «*abbaa*». Ребра, обозначающие функции перехода, заставляют машину проходить через состояния $q_0$, $q_1$, $q_2$, $q_1$, $q_0$ и $q_1$ в указанном порядке. Легко видеть, что, начиная с $q_0$, под действием $t$ получаем

$$(q_0, a) \mapsto q_1, \quad (q_1, b) \mapsto q_2, \quad (q_2, b) \mapsto q_1,$$

$$(q_1, a) \mapsto q_0, \quad (q_0, a) \mapsto q_1.$$

Однако $q_1$ не является заключительным состоянием, и поэтому этот вход не подходит. Проверка диаграммы показывает, что строками, представимыми этой машиной, являются только те строки над алфавитом $\{a, b\}$, в которых

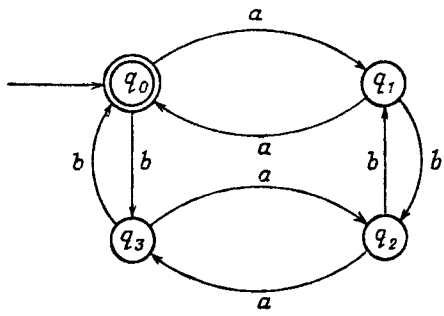



Рис. 9.9

имеется четное число символов $a$ и четное число символов $b$. У этой машины нет явного выхода; требуемая информация может быть извлечена из результирующего состояния только после останова машины. //

Следующие два примера относятся к арифметическому «оборудованию» и фактически порождают непосредственный входной результат.

Пример 2.2. Машина, изображенная на рис. 9.10 для пары строк над $\{0, 1\}$, вычисляет и выводит их сумму. Входные данные начинаются с битов с наименьшими

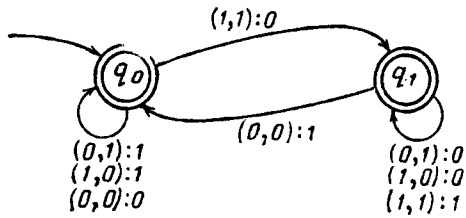



Рис. 9.10

значениями, и пары читаются справа налево. По данным двум $n$-разрядным числам эта машина вычисляет их $n$-разрядную сумму. Однако она не распознает условий переполнения и не работает с отрицательными числами. //

Проверяя свойства функции выхода (функция выхода имеет вид $Q \times \{0, 1\}^2 \to \{0, 1\}$), полезно вспомнить факты,

касающиеся двоичной арифметики и описанные в § 3 гл. 4. В более сложной модели можно учитывать условия, при которых возможны ошибки при двоичном сложении.

Пример 2.3. Машина, изображенная на рис. 9.11, также осуществляет двоичное сложение тем же самым

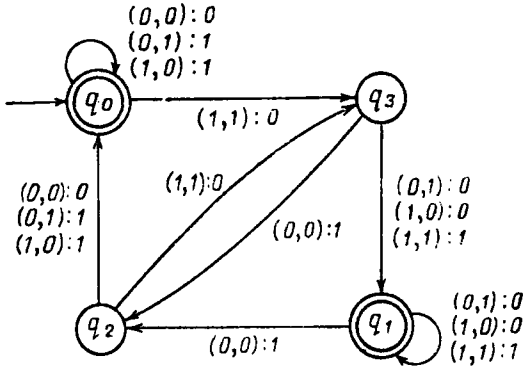



Рис. 9.11

способом, что и в предыдущем случае, за исключением того, что она содержит два дополнительных состояния $q_2$ и $q_3$, которые проверяют наличие ошибок, связанных с внесением в знаковый бит и вынесением из знакового бита соответственно. Заключительные состояния $q_0$ и $q_1$ обозначают выходы, в которых либо произошли оба эти переноса одновременно, либо не произошел ни один из них. Следовательно, эти выходы являются арифметически правильными. //

Эти примеры показывают, что можно выполнить некоторые полезные преобразования и вычисления, однако остается открытым вопрос: насколько «сильными» являются конечные автоматы? В ответе на этот вопрос встречаются две точки зрения. Первая, хотя и нетипичная, заключается в следующем: можно заметить, что использованные в примерах алфавиты основывались лишь на двух символах. Мы можем использовать любой конечный алфавит, однако, как отмечалось в § 1 гл. 8, алфавита размерности 2 всегда достаточно.

Вторая точка зрения заключается в том, что общая доступная память в данное время в определенном компьютере является конечной. Если она содержит $n$ битов, то существует ровно $2^n$ конечных состояний, в которых

память может находиться. Это число может быть большим, но оно всегда конечно (см. упражнение 9.2), и, следовательно, мы имеем конечную машину.

**2.2. Недетерминированные машины.** Рассмотрим с математической точки зрения, что́ могут делать эти абстрактные машины. Чтобы упростить дело, в определении, данном выше, будем игнорировать функцию $p$. Следовательно, мы будем иметь дело только с устройствами, имеющими конечное число состояний; будем их для краткости называть КР (конечными распознавателями или акцепторами). В дальнейшем будет удобно расширить область значений функции переходов $t$ (см. определение) так, чтобы выполнялось условие

$$t\colon\ Q\times\Sigma\to\mathscr{P}(Q).$$

Следовательно, из данного состояния под действием заданных входных значений машина может иметь выбор, куда идти дальше. Это ничего не добавляет к определению, однако обеспечивает простой способ конструирования КР, который будет активно использоваться в этом параграфе. Машины такого типа называют *недетерминированными* КР.

Чтобы еще более прояснить, дело для заданного КР $M$ определим множество строк, представимых машиной $M$ (обозначим это множество через $A(M)$), как указано ниже.

Пусть $M=(Q,\ \Sigma,\ t,\ q_0,\ F)$ и $s\in\Sigma^*$. Запишем $s$ в виде $s=s_1s_2\ldots s_n$ и определим множество $T(s)$ индукцией по длине слова $s$ следующим образом:

$$T(\Lambda)=\{q_0\}\ \text{ и }\ T(\sigma s_k)=\bigcup_{q\in T(\sigma)}t(q,s_k),$$

где $\sigma=s_1\ldots s_{k-1}$ для всех $k$, $1\leqslant k\leqslant n$. Таким образом, $T(s)$ — это множество всех заключительных состояний $M$, которые могут быть достигнуты под действием входной последовательности $s$. Если $T(s)\cap F=\varnothing$, то слово $s$ представимо; поэтому множество строк, представимых машиной $M$, имеет вид

$$A(M)=\{s\colon\ T(s)\cap F\neq\varnothing\}.$$

Недетерминированные КР полезны тем, что они упрощают задачу построения сложных машин. Существенным является то, что нам хотелось бы иметь возможность собрать машины вместе таким образом, чтобы не было нужды, по крайней мере на первом этапе, касаться вопроса,

является или нет результат перехода функцией или отношением над $(Q\times\Sigma)\times Q$. (Напомним, что функция $Q\times\Sigma\to\mathcal{P}(Q)$ соответствует отношению $Q\times\Sigma\to Q$.)

Введение недетерминированности не увеличивает потенциала КР. От недетерминированной машины всегда можно перейти к детерминированной, как будет показано в следующей теореме.

Теорема. *Для любого недетерминированного КР $M_1$ существует детерминированный КР $M_2$ такой, что $A(M_1)=A(M_2)$.*

Доказательство. Дадим вначале общую схему доказательства. Пусть

$$M_1=(Q_1,\ \Sigma_1,\ t_1,\ q_0,\ F_1),\ \text{а}\ M_2=(Q_2,\ \Sigma_2,\ t_2,\ q_0',\ F_2).$$

Установим вначале, что $Q_2=\mathcal{P}(Q_1)$ (в общем случае все эти состояния нам будут не нужны) и $\Sigma_2=\Sigma_1$. Для этого, начиная с $q_0\in Q_1$, рассмотрим множество всех состояний в $t(q_0,\ s_1)$ (для $s_1\in\Sigma_1$) и назовем это множество образом $(q_0',\ s_1)$ функции $t_2$; т. е. мы переводим различные возможности машины $M_1$ в единственное состояние машины $M_2$. Предположим, что далее мы читаем символ $s_2$. В $M_1$ это могло бы вызвать перевод состояния $t_1(q_0,\ s_1)$ в состояние из множества $t_1(t_1(q_0,\ s_1),\ s_2)$. Множество всех состояний, полученных таким образом, сейчас дает единственное состояние в $M_2$, получающееся в результате применения $t_2$ к образу из $(q_0',\ s_1)$ и $s_2$. Таким образом, мы приходим к конструкции переходов между состояниями в $Q_2$.

Так как $Q_1$ конечно, то конечно и $Q_2$; следовательно, необходимо вернуться к ранее построенным состояниям, и, таким образом, процесс в конце концов оборвется. Если одно из «выбранных» состояний в $Q_1$ было в $F_1$, то состояние в $Q_2$ является заключительным и, следовательно лежит в $F_2$.

Перейдем к подробному доказательству. В соответствии с описанной выше конструкцией $q_0'=\{q_0\}_n$

$$t_2\colon\ (\{q_{i_1},\ \ldots,\ q_{ij}\},\ s_k)\mapsto\left\{\bigcup_{l=1}^{j}t_1(q_{il},\ s_k)\right\}$$

и

$\{q_{i1},\ \ldots,\ q_{ij}\}\in F_2$ тогда и только тогда, когда $q_{ik}\in F_1$ при некотором $k$, $1\leqslant k\leqslant j$.

Возвращаясь назад к определению множества $A(M)$ для данной машины $M_1$, мы можем расширить $t_1$ и $t_2$,

чтобы получить $T_1$ и $T_2$ так, как это показано ниже:

$$T_1(\Lambda) = \{q_0\}, \quad T_1(\sigma s_k) = \bigcup_{\in T_1(\sigma)} t_1(q, s_k),$$

$$T_2(\Lambda) = \{q_0'\}, \quad T_2(\sigma s_k) = \bigcup_{q \in T_2(\sigma)} t_2(q, s_k) = t_2(q, s_k),$$

где $T_2(\sigma) = \{q\}$ и $\sigma = s_1 \ldots s_{k-1}$.

Поскольку $A(M) = \{s\colon\ T(s) \cap F \neq \varnothing\}$ и существует естественное соответствие между $F_1$ и $F_2$, то мы должны лишь продемонстрировать такое же соответствие между $T_1(s)$ и $T_2(s)$ для любого $s$; т. е. $\{\ldots, q_i, \ldots\} = T_1(s)$ тогда и только тогда, когда $\{\{\ldots, q_i, \ldots\}\} = T_2(s)$. Сделаем это индукцией по длине $s$.

Если $|s| = 0$, то $s = \Lambda$, откуда

$$T_1(\Lambda) = \{q_0\}, \quad T_2(\Lambda) = \{q_0'\} = \{\{q_0\}\}.$$

Предположим теперь, что соответствие имеет место для всех строк $\sigma$: $|\sigma| \leqslant k - 1$, и рассмотрим строку $\sigma s_k$. По предположению индукции
$\{\ldots, q_i, \ldots\} = T_1(\sigma)$ тогда и только тогда, когда
$\{\{\ldots, q_i, \ldots\}\} = T_2(\sigma)$.
Однако тогда, если $q_j \in t_1(q_i, s_k)$, то отсюда следует, что $q_j \in T_1(\sigma s_k)$ и (по определению $t_2$)

$$\{\ldots, q_j, \ldots\} \in t_2(\{\ldots, q_i, \ldots\}, s_k) = T_2(\sigma s_k).$$

Из определения $T_2$ и $t_2$ следует, что все элементы из $T_2(\sigma s_k)$ должны выводиться таким же образом; поэтому $\{\ldots, q_i, \ldots\} = T_1(\sigma s_k)$ тогда и только тогда, когда
$\{\{\ldots, q_i, \ldots\}\} = T_2(\sigma s_k)$.
Следовательно, $T_1(s)$ для любого $s \in \Sigma^*$ «соответствует» $T_2(s)$, и поэтому

$$A(M_1) = \{s\colon\ T_1(s) \cap F_1 \neq \varnothing\} =$$
$$= \{s\colon\ T_2(s) \cap F_2 \neq \varnothing\} = A(M_2). \quad //$$

Приведенная выше аргументация была достаточно сложной и включала некоторое сомнительное (однако строго определенное) понятие «соответствие», которое подразумевало уровень вложения включаемых множеств. В частных примерах мы можем обойти это путем введения подходящих имен для результирующих состояний в детерминированной машине $M_2$.

Как будет показано в последующем примере, построение $M_2$ из $M_1$ является непосредственным, однако, чтобы сохранить математическую строгость, напомним вначале

общепринятое соглашение, связанное с функциями. В силу используемых построений такое соглашение было бы здесь неуместным, однако краткое напоминание об этой погрешности послужит объяснением, «откуда берутся некоторые множества».

Пусть задана функция $f\colon A \to B$ такая, что $f\colon x \mapsto y$. Мы должны были бы писать $f(x) = \{y\}$, однако часто запись сводится к $f(x) = y$.

Аналогично обычно детерминированный перенос обозначаем как

$$t\colon Q \times \Sigma \to Q,$$

где

$$t\colon (q_i, s) \mapsto q_j, \quad t(q_i, s) = q_j.$$

Однако, когда мы имеем $t\colon Q \times \Sigma \to \mathscr{P}(Q)$ (как в $M_1$ и $M_2$), то мы должны заключать множества в скобки даже тогда, когда $|t(q_i, s)| = 1$, и писать $t(q_i, s) = \{q_j\}$. Перейдем к примеру.

Пример 2.4. Возьмем $M_1$ таким, как указано на рис. 9.12, *a*. Тогда

$$t_1(q_0, x) = \{q_1, q_2\}$$

и $q_1 \in F_1$; следовательно, получаем $\{q_1, q_2\}$ в $F_2$. Аналогично

$$t_1(q_1, x) = \{q_0\}, \quad t_1(q_2, x) = \{q_2, q_3\};$$

следовательно,

$$t_2(\{q_1, q_2\}, x) = \{q_0, q_2, q_3\} \text{ в } M_2 \text{ и т. д.}$$

Окончательно получаем ситуацию, изображенную на рис. 9.12, *b*. Тогда перенумерация дает нам рис. 9.12, *c*. //

С этого момента мы не будем выяснять, какие машины рассматриваются — детерминированные или нет; они всегда могут рассматриваться как детерминированные.

**2.3. Составные машины.** Сейчас мы в состоянии описать, как заданную совокупность КР можно «собрать вместе» для того, чтобы получить некоторые корректно определенные множества строк. Основные свойства даны в виде предложения, однако вместо формальных доказательств мы дадим лишь описание включаемых в доказательство построений.

Предложение. *По заданным машинам*

$$M_1 = (Q_1, \Sigma, t_1, q, F_1), \quad M_2 = (Q_2, \Sigma, t_2, p, F_2)$$

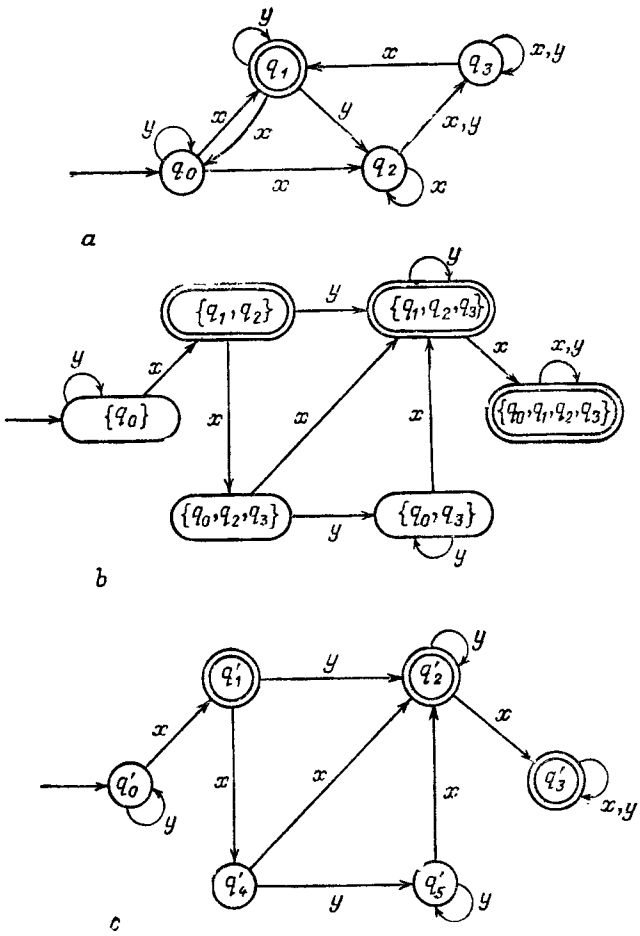



Рис. 9.12

*мы можем построить машины* $M_3, \ldots, M_8$ *такие, что*

$$A(M_3) = \Sigma^* \setminus A(M_1),$$
$$A(M_4) = A(M_1) \cap A(M_2),$$
$$A(M_5) = A(M_2) \setminus A(M_1),$$
$$A(M_6) = A(M_1) \cup A(M_2),$$
$$A(M_7) = A(M_1) A(M_2),$$
$$A(M_8) = A^*(M_1).$$

Построение $M_3$. Машина $M_3$ получается из $M_1$ заменой множества $F_1$ заключительных состояний на множество $Q_1 \backslash F_1$. Следовательно,

$$M_3 = (Q_1,\ \Sigma,\ t_1,\ q,\ Q_1 \backslash F_1).$$

Чтобы получить $M_4$, возьмем $Q_4 = Q_1 \times Q_2$, возможно, с некоторыми подходящими переобозначениями. Пусть $r = (q,\ p)$. Определим $t_4$ как

$$t_4\colon\ ((q_i,\ p_j),\ s) \mapsto (t_1(q_i,\ s),\ t_2(p_j,\ s)),$$

и множество

$$F_4 = \{(q_i,\ p_j)\colon\ q_i \in F_1,\ p_j \in F_2\}.$$

Тогда

$$M_4 = (Q_4,\ \Sigma,\ t_4,\ r,\ F_4) \quad \text{(см. пример 2.5)}.$$

Построение $M_5$ следует из $M_2$ и $M_3$, так как

$$A(M_2) \backslash A(M_1) = A(M_2) \cap (\Sigma^* \backslash A(M_1)) = A(M_2) \cap A(M_3).$$

Построение $M_6$ проводится подобно $M_4$. Имеем $Q_6 = Q_4$, $t_6 = t_4$, однако

$$F_6 = \{(q_i,\ p_j)\colon\ q_i \in F_1 \text{ или } p_j \in F_2\}.$$

Тогда

$$M_6 = (Q_6,\ \Sigma,\ t_6,\ r,\ F_6).$$

При построении $M_7$ используется идея присоединения выхода из $M_1$ к входу $M_2$. Однако при этом следует быть внимательными, чтобы не проделать лишний шаг при движении от $M_1$ к $M_2$. Итак, возьмем $Q_7 = Q_1 \sqcup Q_2$ (разъединяющее объединение $\sqcup$ изменяет названия так, что $Q_1$ и $Q_2$ не имеют общих состояний, и использует $Q_1 \cup Q_2$), множество

$$F_7 = \begin{cases} F_2, & p \notin F_2, \\ F_1 \cup F_2, & p \in F_2, \end{cases}$$

и добавим переходы от элементов $F_1$ к другим состояниям $M_2$, а именно

$$t_7 = t_1 \cup t_2 \cup \{((q_i,\ s),\ p_j),\ \text{где } q_i \in F_1,\ p_j \in t_2(p,\ s)\}.$$

Тогда

$$M_7 = (Q_7,\ \Sigma,\ t_7,\ q,\ F_7).$$

Построение $M_8$ проводится аналогично. Сначала необходимо позаботиться о том, чтобы $A^0(M_1) = \varnothing$ было

представимо. Для этого добавим новое начальное состояние $u$ ($u \notin Q_1$) таким образом, чтобы выполнялись условия $Q_8 = Q_1 \cup \{u\}$, $F_8 = F_1 \cup \{u\}$, а $t_8$ расширим (так же, как и $t_7$) следующим образом:

$$t_8 = \begin{cases} t_1 \cup \{((u, s), q_j), & q_j \in t_1(q, s), s \in \Sigma\}, \\ t_1 \cup \{((q_i, s), q), & t_8(q_i, s) \in F_1, s \in \Sigma\}. \end{cases}$$

Тогда

$$M_8 = \{Q_8, \Sigma, t_8, u, F_8\}. \ //$$

Пример 2.5. На рис. 9.13—9.15 даются последовательность диаграмм состояний для машин $M_1$ и $M_2$ и результирующие составные машины $M_3$, ..., $M_8$, которые были построены выше. Заметим, что некоторые из этих машин являются недетерминированными, однако от недетерминированности можно избавиться при помощи предыдущих результатов. //

Построения, включающие объединение, конкатенацию и замыкание (операция «звездочка»), обеспечивают основу алгебраических систем для описания множеств, представимых конечными распознавателями. Эта алгебра будет обсуждаться в § 3, а в оставшейся части этого параграфа обратим внимание на моделирование *реальных* компьютеров.

**2.4. Моделирование «реальных» компьютеров.** Как уже отмечалось, если данный компьютер имеет память, состоящую из $n$ битов, то для того, чтобы смоделировать его поведение, вообще говоря, необходим конечный автомат с $2^n$ состояниями. Пусть начальное состояние имеет такую конфигурацию, что программа и значения данных находятся в «ядре». Тогда не существует внешних стимулов, которые могли бы дать начало последовательности переходов. Однако рассмотрение машины в несколько другом виде может привести к выполнению команд, использующих конечное число (уже определенных) состояний машины.

Предположим, что память состоит из $m$ $l$-битных слов; тогда $n = m * l$. Теперь отделим программный указатель (называемый также программным счетчиком или указателем следующей команды) от оставшейся части машины; получим машину с $n' = l * (m - 1)$ битами памяти. (Для простоты предполагается, что указатель имеет ту же длину, что и другие компоненты памяти.) Теперь

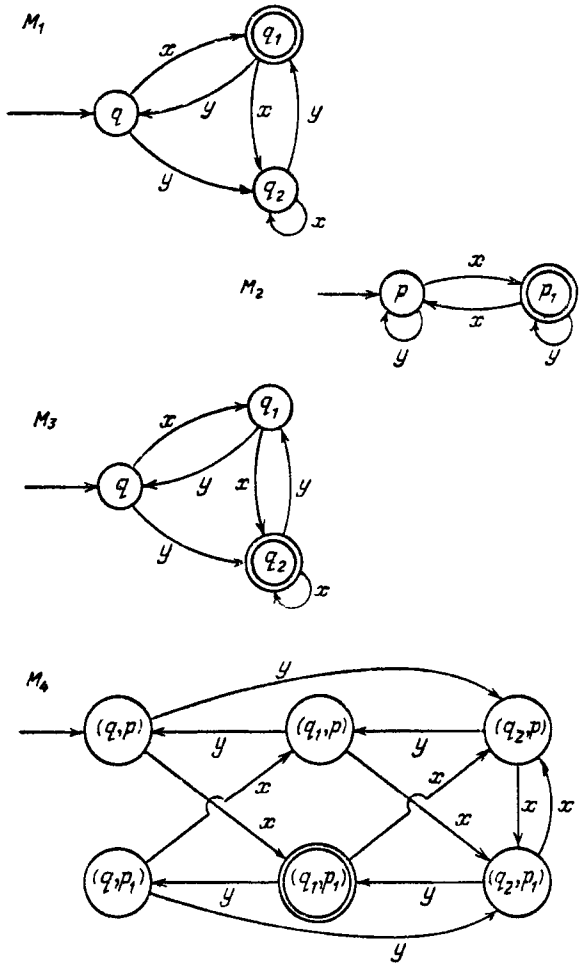



Рис. 9.13

можно программный указатель рассматривать после каждого изменения состояния и его значение использовать

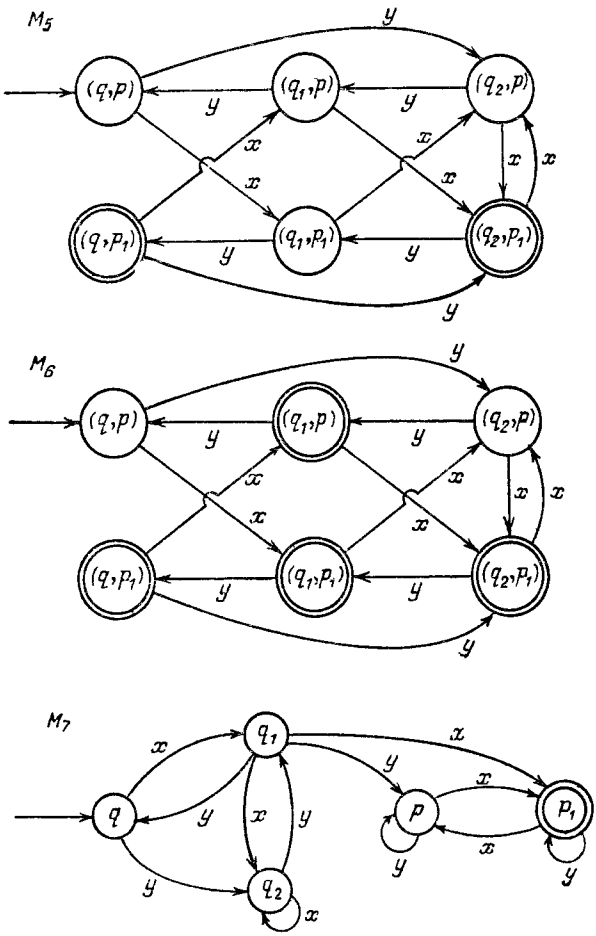



Рис. 9.14

для того, чтобы осуществлять очередной переход. Таким образом, функция перехода имеет вид

$$t\colon Q \times P \to Q \times P,$$

где $t\colon (q_a, p_b) \mapsto (q_c, p_d)$ соответствует значению указателя $p_b$, что означает: выделить команду из состояния $q_a$ так, чтобы перевести машину в состояние $q_c$, а значение

$p_b$ в программном указателе заменить на $p_a$. (Здесь удобно рассматривать команду как действие, частично осуществляемое программным обеспечением, а частично — аппаратным методом. В этой модели мы не делаем различия между ними; на практике бывает, что программное обеспечение — это просто начальные данные, которые управляют аппаратными процессами.)

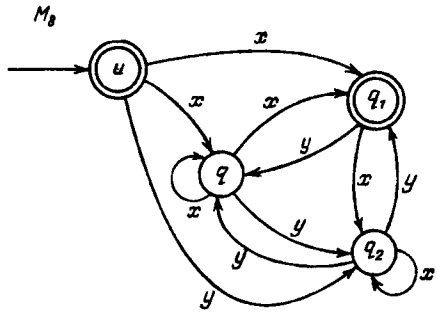



Рис. 9.15

Обычно программы начинают работу с программного указателя, установленного на определенное значение. Начиная с этого значения, все последующие значения порождаются автоматически при помощи описанного выше процесса до тех пор, пока не выполнится команда останова.

Конечно, не все $2^l$ значений, которые могут содержаться в программном указателе, должны быть верными. Часто $2^l$ может быть больше $m$, и, следовательно, $2^l - m$ значений будут неправильными адресами и будут вызывать ошибку, приводящую к останову.

Число правильных значений, которые может принимать программный указатель, определяет число переходов, которые могут быть нарисованы в виде стрелок из каждого состояния. Однако это не ограничивает число состояний, которое равно $2^{n'}$.

При отсутствии дополнительной информации о моделируемой системе мы можем сказать о ней довольно мало, однако на один момент следует обратить внимание. Если построенный нами конечный автомат останавливается за «короткое» время или под воздействием команды останова, или из-за полученного неправильного значения указателя, то результат известен. Однако если машина

продолжает считать в течение длительного промежутка времени, то встает вопрос: остановится ли она когда-нибудь и получим ли мы результат? Поскольку машина конечна (и, следовательно, имеет ограниченную память), то эта задача разрешима, однако неизвестно, как долго ждать результат. В силу способа построения машины пара $(q, p)$, являющаяся указателем состояния, единственным образом определяет последовательность состояний, которые должны последовать за данным состоянием, и, следовательно, повторение любой данной пары подразумевает бесконечный цикл. Поэтому, чтобы гарантировать, что последовательность вычислений никогда не остановится, мы должны обеспечить, чтобы она пересекала заданную дугу перехода *дважды*. Все это требует достаточно много времени, так как необходимо выполнить $2^{n'} * m = 2^{l*(m-1)} * m$ команд. Даже при достаточно малых значениях $m$ и $l$ число операций очень велико, и поэтому реальное время их выполнения требует десятков, сотен и даже тысяч лет.

Урок, который мы должны извлечь из рассмотрения этого упражнения, заключается в том, что практически невозможно проверить *корректность* программы разумных размеров, используя тестовые данные. Требуется слишком много времени для выполнения всех прогонов программы. Такие тестовые прогоны могут только находить ошибки.

Упражнение 9.2.

1. Построить конечный автомат, который будет распознавать четные числа, записанные в двоичной форме и читаемые слева направо.

2. Построить конечный автомат с входным алфавитом $\{a, b\}$, который останавливается в заключительном состоянии тогда и только тогда, когда входные данные не содержат рядом двух соседних элементов $a$ и двух соседних элементов $b$.

3. Показать невозможность построения конечного автомата, на вход которого поступают только строки над алфавитом $\{a, b\}$, и только те из них являются представимыми, которые имеют равное количество символов $a$ и $b$.

4. Построить (детерминированный) конечный автомат, для которого представимыми являются только строки над $\{0, 1\}$, состоящие из чередующихся единиц и нулей, следующих за чередующимися парами из единиц и нулей.

5. Придумать конечный автомат, способный распознавать десятичные числа, записанные в виде

$$\pm dd^*.\ d^*E \pm dd,$$

где $d \in \{0, 1, 2, \ldots, 9\}$.

## § 3. Регулярная алгебра

Способы комбинирования конечных распознавателей могут использоваться для проверки аксиом множеств, из которых могут развиваться алгебраические системы. Классическая система, известная как регулярная алгебра, основана на трех операциях — объединении, пересечении и конкатенации, определенных на множестве строк. Уравнения в алгебре могут точно определять некоторые множества строк; эти множества являются решением уравнений (п. 3.1). Результаты, вытекающие из решений, имеют непосредственное применение к аспектам теории языков (п. 3.2).

**3.1. Выражения и уравнения.** Перед тем как дать формальное определение, мы введем две константы внутри системы. Существует множество строк $\varnothing$ и $\{\Lambda\}$, которые обозначим символами 0 и 1 соответственно. Мы обосновываем их использование изображением конечных распознавателей соответственно на рис. 9.16, *a* и *b*.

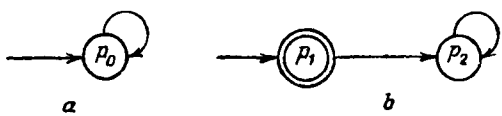



Рис. 9.16

Определение. Пусть $X$ — множество строк, представимое конечным распознавателем. Тогда $X$ называется *регулярным выражением*. Если $X$ и $Y$ обозначают множества строк, представимых двумя определенными машинами, то (в силу определения построения конечных распознавателей) $X \cup Y$ (далее будем писать $X + Y$), $X^*$ и $XY$ также являются регулярными выражениями.

Разрешенные конструкции составных конечных автоматов предполагают следующие аксиомы для работы с произвольными регулярными выражениями $A$, $B$ и $C$:

1) $A + B = B + A$; 2) $A + (B + C) = (A + B) + C$;

3) $A + A = A$; 4) $A + 0 = A$;

5) $A(BC)=(AB)C$; 6) $A1=A=1A$;

7) $A0=0=0A$; 8) $A(B+C)=AB+AC$;

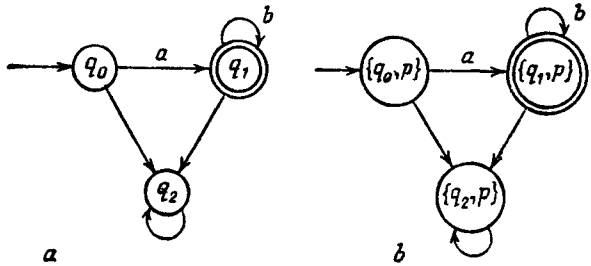
9) $(A+B)C=AB+BC$; 10) $0^*=1$;

11) $A^*=A+A^*$; 12) $(A^*)^*=A^*$.

Такую алгебраическую систему называют *регулярной алгеброй.*

Проверка достоверности этих аксиом требует рассмотрения совершенно *общих* случаев. Поэтому их трудно должным образом описать. Тем не менее, чтобы обозначить используемые ниже идеи, рассмотрим два примера, исходя из двух аксиом, примененных к *конкретному* регулярному выражению.

Пример 3.1. Пусть $A=\{ab^n\colon 0\leqslant n\}$. Это выражение регулярно, так как является представимым для машины $M$, изображенной на рис. 9.17, *а*.




*а* *б*

Рис. 9.17

Используя машину, изображенную на рис. 9.16, *а*, которая распознает 0, и применяя подходящее построение, получим машину, изображенную на рис. 9.17, *б*, которая, очевидно, эквивалентна $M$. Следовательно, $A=A+0$. //

Пример 3.2. Используя $A$ и 1, определяемые машинами, данными выше, и обычной конструкцией конкатенации, получаем недетерминированный конечный распознаватель, изображенный на рис. 9.18, *а*, который будет представлять строки в регулярном выражении $1A$. Эту машину делаем детерминированной (рис. 9.18, *б*), затем получаем машину (рис. 9.18, *в*), эквивалентную $M$, которая удовлетворяет (в данном случае) части аксиомы 6), а именно $1A=A$. //

Определив регулярную алгебру, посмотрим, как ее использовать. Предположим, что $A$, $B$ и $C$ являются заданными регулярными выражениями (множествами строк)

и что $X$ и $Y$ — неизвестные регулярные выражения такие, что выполнены следующие соотношения:

$$X = AX + BY, \quad Y = XC + B.$$

Существуют ли решения этих «уравнений», и если да, то как их найти? Как мы вскоре увидим, регулярные

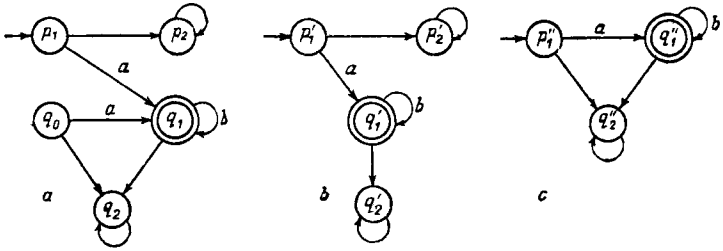



Рис. 9.18

уравнения не имеют единственного решения; поэтому мы вначале введем понятие аппроксимации для регулярных выражений.

Определение. Для регулярных выражений $X$ и $Y$ *отношение* $\leqslant$ определяем следующим образом:

$X \leqslant Y$ ($X$ аппроксимирует $Y$), если $X + Y = Y$. //

Теорема. *Отношение* $\leqslant$, *определенное над регулярными выражениями, является отношением порядка.*

Доказательство.

Транзитивность имеет место, так как

$$X \leqslant Y,\ Y \leqslant Z \Rightarrow X + Y = Y,\ Y + Z = Z. \qquad (*)$$

Поэтому

$$X + Z = X + (Y + Z) = (X + Y) + Z = Y + Z = Z$$

и, следовательно, $X \leqslant Z$.

Антисимметричность имеет место, поскольку

$$X \leqslant Y,\ Y \leqslant X \Rightarrow Y = X + Y = Y + X = X.$$

Рефлексивность следует из $X + X = X$. //

Отсюда также следует, что $X \leqslant Y \Rightarrow ZX \leqslant ZY$ для регулярных выражений $X$, $Y$ и $Z$, однако детали доказательства оставим в качестве упражнения.

Перед тем как рассматривать уравнения, напомним, что из определения операции замыкания следует

$$A^* = A^0 + A^1 + \ldots + A^n + \ldots = \sum_{n=0}^{\infty} A^n,$$

где $A^0 = 1$. Это соотношение оказывается полезным при доказательстве следующих утверждений.

Лемма. *Пусть $Y$ и $Z$ — решения регулярного уравнения $X = AX + B$. Тогда*

а) $B \leqslant Y$,

б) $AY \leqslant Y$,

в) $Y + Z$ — *решение.*

Доказательство.

а) Так как $Y$ — решение уравнения $X = AX + B$, то $Y = AY + B$, и поэтому

$$Y + B = (AY + B) + B = AY + (B + B) = AY + B = Y.$$

Следовательно, $B \leqslant Y$.

б) Аналогично $Y + AY = (B + AY) + AY = B + AY = Y$. Следовательно, $AY \leqslant Y$.

в) Имеем

$$Y + Z = AY + B + AZ + B = A(Y + Z) + B. \ //$$

Уже достаточно много сказано о свойствах решений уравнения, однако существует ли хотя бы одно решение?

Лемма. *$A^*B$ является решением уравнения $X = AX + B$.*

Доказательство. Подставляя $A^*B$ в уравнение, имеем

$$A(A^*B) + B = (AA^* + A^0)B = \left(A \sum_{n=0}^{\infty} A^n + A^0\right) B =$$

$$= \left(\sum_{n=1}^{\infty} A^n + A^0\right) B = \left(\sum_{n=0}^{\infty} A^n\right) B = A^*B. \ //$$

Теорема. *$A^*B$ является наименьшим (по отношению к $\leqslant$) решением регулярного уравнения $X = AX + B$.*

Доказательство. Из лемм следует, что $A^*B$ — решение, и если $Y$ — какое-либо другое решение, то $B \leqslant Y$, $AB \leqslant AY \leqslant Y$. Поэтому $AB \leqslant Y$, $A^nB \leqslant Y$, и, добавляя неравенства для всех $n \in \mathbb{N} \cup \{0\}$, получаем $A^*B \leqslant Y$. Следовательно, утверждение теоремы верно. //

Заметим, что $A^*B$ — это множество строк, каждая из которых является решением уравнения; если $A = \{a\}$ и $B = \{b\}$, то все строки вида $a^nb$ также являются решениями.

Заметим также, что так как $A^*B$ является наименьшим решением уравнения $X = AX + B$, то оно является наименьшим выражением таким, что $X \mapsto AX + B$ совпадает с тождественным отображением. Поэтому его также

называют *наименьшей стационарной точкой* уравнения (или *минимальной стационарной точкой*).

Рассмотрев одно частное уравнение, мы сейчас можем непосредственно получить аналогичные результаты для других похожих уравнений; доказательства в этих случаях будем опускать.

Теорема. *Для заданных регулярных выражений $A$ и $B$ уравнение $X = AX + B$ эквивалентно паре уравнений*

$$X = YB,\ Y = YA + 1$$

*в том смысле, что они имеют одно и то же наименьшее решение, и это решение есть $X = A^*B$.* //

Теорема. *Для заданных регулярных выражений $A$ и $B$ уравнение $X = AX + B$ эквивалентно паре уравнений*

$$X = BY,\ Y = AY + 1.$$

(*В этом случае обе системы наименьшее решение $X = BA^*$.*) //

Эти две теоремы позволяют нам преобразовывать правые линейные формы ($X = AX + B$) в левые линейные формы ($X = XA + B$), которые соответствуют правым и левым рекурсиям внутри регулярных грамматик. Детали этого соответствия даны в п. 3.2, однако сначала мы обобщим результаты на системы алгебраических уравнений.

Предположим, что $X_1, \ldots, X_n$ — неизвестные регулярные выражения и что $A_{ij}$ $(1 \leqslant i, j \leqslant n)$ и $B_1, \ldots, B_n$ — известные регулярные выражения такие, что

$$\begin{aligned}
X_1 &= X_1A_{11} + X_2A_{21} + \ldots + X_nA_{n1} + B_1,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
X_i &= X_1A_{1i} + X_2A_{2i} + \ldots + X_nA_{ni} + B_i,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
X_n &= X_1A_{1n} + X_2A_{2n} + \ldots + X_nA_{nn} + B_n.
\end{aligned}$$

Теперь можно использовать операции, определенные на регулярных выражениях, чтобы ввести операции на матрицах из регулярных выражений:

$$(\mathbf{C} + \mathbf{D})_{ij} = \mathbf{C}_{ij} + \mathbf{D}_{ij},$$

$$(\mathbf{CD})_{ij} = \sum_k \mathbf{C}_{ik}\mathbf{D}_{kj},$$

$$\mathbf{C}^* = \sum_{n=0}^{\infty} \mathbf{C}^n;$$

$\mathbf{C} \leqslant \mathbf{D}$ тогда и только тогда, когда $\mathbf{C}_{ij} \leqslant \mathbf{D}_{ij}$ для всех $i$, $j$,

где **C** и **D** — согласованные матрицы, элементы которых являются регулярными выражениями. Обозначив через **X** «регулярную матрицу», мы можем представить приведенные выше уравнения в виде

$$\mathbf{X} = \mathbf{XA} + \mathbf{B}.$$

Применяя рассуждения, аналогичные тем, которые использовались в случае одного уравнения, можно показать, что эта система имеет минимальную стационарную точку $\mathbf{BA}^*$, которая достигается на решениях $\mathbf{X} = \mathbf{BY}$, где $\mathbf{Y} = \mathbf{AY} + \mathbf{I}$; **I** — единичная матрица. Соответствующие результаты справедливы для правых линейных систем. Из замечаний следует, что эти системы эквивалентны. Это помогает удалению левых рекурсий в контекстно-свободных грамматиках.

Ясно, что для уравнения $\mathbf{X} = \mathbf{XA} + \mathbf{B}$ решение может быть определено как

$$\mathbf{X}_i = (\mathbf{BY})_i = \sum_k \mathbf{B}_k \mathbf{Y}_{ki},$$

где

$$\mathbf{Y}_{ij} = (\mathbf{AY} + \mathbf{I})_{ij} = \sum_k \mathbf{A}_{ik} \mathbf{Y}_{kj} + \mathbf{I}_{ij}.$$

Пример 4.4 гл. 8 показывает, как этот результат переносится на контекстно-свободные грамматики. Преобразования, используемые в этом примере (хотя они и не строго обоснованы), появились при помощи рассуждений по аналогии. Мы завершаем этот раздел, указывая на формальную связь между регулярной алгеброй конечных распознавателей и регулярными грамматиками.

**3.2. Представления регулярных грамматик.** Напомним, в гл. 8 структурная грамматика $G = (N, T, P, S)$ называлась грамматикой Хомского типа 3 или ($\Lambda$-свободной) *регулярной грамматикой,* если все элементы $P$ имели вид $A \to x$ или $A \to xB$, где $x \in T$ и $A, B \in N$, или же (если $\Lambda \in L(G)$) $S \to \Lambda$; в этом случае $S$ не встречается ни в какой правой части.

Множество предложений, порожденных регулярной грамматикой, называют *регулярным множеством* или *регулярным языком.* В настоящий момент мы в состоянии обосновать использование дважды одной и той же терминологии и, следовательно, использовать регулярную алгебру в грамматиках.

Основной результат состоит из двух частей и дан в виде теоремы с конструктивным доказательством. Часть I является непосредственной, так как каждый пе-

реход определяют на всей его области определения; в части II не обязательно, чтобы грамматика $(N, T, P, S)$ для данных $X \in N$ и $a \in T$ имела продукцию вида $X \to a$ или $X \to aY$ при некотором $Y \in N$.

Теорема. *Множество, представимое конечным автоматом, является в точности таким же, как если бы выводилось из регулярных грамматик.*

Доказательство. Опишем конструкции доказательства.

Вначале рассмотрим конечный автомат $M = (Q, \Sigma, t, q, F)$. Теперь построим регулярную грамматику $G = (N, T, P, S)$. Если $\Lambda \notin A(M)$, то можно сделать следующее. Пусть $N = Q$, $T = \Sigma$ и $S = q$; построим $P$ такое, что

$$P = \{X \to aY,\ Y \in t(X, a)\} \cup \{X \to a,\ t(X, a) \cap F \neq \varnothing\}.$$

Если $\Lambda \in A(M)$, то расширяем конструкцию, создавая новый нетерминал $\bar{q}$, полагая $S = \bar{q}$ и добавляя к $P$ продукции

$$S \to \Lambda,$$
$$S \to aY \qquad (\text{если } Y \in t(q, a)),$$
$$S \to a \qquad (\text{если } t(q, a) \cap F = \varnothing).$$

Сейчас легко показать, что для каждого $x \in \Sigma^*$ условие $x \in A(M)$ выполнено тогда и только тогда, когда $x \in L(G)$. Очевидно, что если $q \in F$, то $\Lambda$ представимо автоматом $M$ и $S \to \Lambda$ — продукция $G$ и наоборот. Если также $a \in A(M)$ и $|a| = n$, то

$$a = a_1 a_2 \ldots a_n,\ a_i \in \Sigma,$$

и поэтому существует путь в графе $M$, как показано на

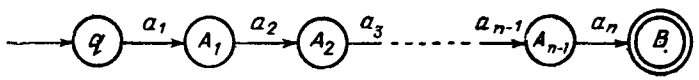



Рис. 9.19

рис. 9.19. Здесь $A_i \in N$ (не обязательно все различны) и $B \in F$, и по построению

$$\{S \to a_1 A_1,\ A_1 \to a_2 A_2,\ \ldots,\ A_{n-2} \to a_{n-1} A_{n-1},\ A_{n-1} \to a_n\} \subseteq P$$

(рис. 9.20). Таким образом, $a \in L(G)$. Проводя рассуждения в обратном порядке, аналогично получаем $L(G) \subseteq A(M)$, и, следовательно, равенство доказано.

Сейчас надо показать, что для данной ($\Lambda$-свободной) регулярной грамматики предложения этой грамматики могут быть представимы некоторым конечным автоматом, а все другие строки не представимы этим автоматом.

Возьмем $G_1 = (N_1, T_1, P_1, S_1)$ и построим $M_1 = (Q_1, \Sigma_1, t_1, q_1, F_1)$, как показано далее. Пусть $\Sigma_1 = T_1$ и $Q_1 =$

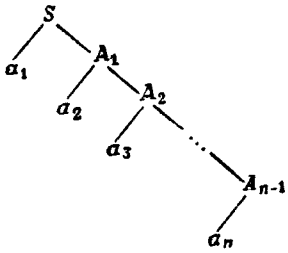



Рис. 9.20

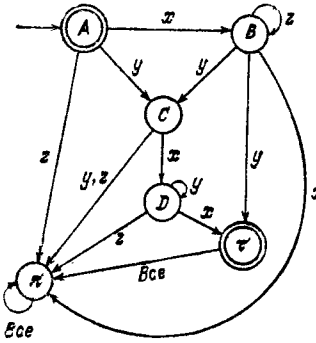



Рис. 9.21

$= N_1 \cup \{\tau\} \cup \{\pi\}$ (где $\tau$ и $\pi$ — специальные символы, которых нет в $N_1$; $\tau$ представляет правильную терминальную строку, а $\pi$ представляет ошибку), $q_1 = S_1$ и $F_1 = \{\tau\}$. Если также $\Lambda \in L(G_1)$, т. е $S_1 \to \Lambda$ принадлежит $P_1$, то $F_1 = \{\tau, q_1\}$. Окончательно имеем

$$t_1 = \{((X, a), Y)\colon X \to aY \text{ есть в } P_1\} \cup$$

$$\cup \{((X, a), \tau)\colon X \to a \text{ есть в } P_1\} \cup$$

$$\cup \{((\tau, a), \pi) \text{ для всех } a \in \Sigma_1\} \cup$$

$\cup\{((X, b), \pi)$: если $X \equiv N_1$ и не $X \to bY$ для любого $Y \in N_1$ и не выполняется условие $X \to b$ есть в $P_1\} \cup$

$$\cup \{((\pi, a), \pi) \text{ для всех } a \in \Sigma_1\}.$$

Обоснование того факта, что эта машина представляет $L(G_1)$, оставляем в качестве упражнения. //

Пример 3.3. Пусть задана регулярная грамматика $G = (\{A, B, C, D\}, \{x, y, z\}, P, A)$, где

$$P = \{A \to \Lambda \mid xB \mid yC,\ B \to zB \mid y \mid yC,\ C \to xD,\ D \to yD \mid x\}.$$

Используя описанную выше конструкцию, получим машину, изображенную на рис. 9.21. //

Упражнение 9.3.

1. Пусть заданы регулярные выражения $A$, $B$, $C$ и $D$ такие, что $A \leqslant C$ и $B \leqslant C$. Доказать, что

а) $A + D \leqslant C + D$; б) $AD \leqslant CD$;

в) $DA \leqslant DC$; г) $A^* \leqslant C^*$; д) $A + B \leqslant C$.

2. Исследовать, как преобразуются результаты п. 1 данного упражнения, если **A**, **B**, **C** и **D** — регулярные матрицы размера $n \times n$.

3. Определить конечный автомат для языка $L$, определенного как 11*01* (т. е. $L = \{w \in \{0, 1\}^*: w\}$ начинается с 1 и имеет только один нуль. Выписать правую линейную грамматику, определяемую этим автоматом, и последовательность перемещений, делающих представимой строку 111011.

ГЛАВА 10

# КОМПЬЮТЕРНАЯ ГЕОМЕТРИЯ

Эта глава дает некоторые сведения из геометрии, которые широко используются в компьютерной графике и компьютерном вспомогательном математическом обеспечении; при этом делается попытка изложить все более строго и унифицированно, чем это делается во многих книгах по компьютерам. Мы стремимся представить серию «приемов» в некотором, очевидно, случайном порядке, а также использовать концепции, развитые в предыдущих главах (особенно в гл. 1—3, 5, 6), чтобы обеспечить основу для обсуждения выбранных тем, включающих однородные координаты, кривые и поверхности. Дано также четырехмерное представление данных трехмерной геометрии, широко используемое на практике.

Перед началом обсуждения будет полезно, чтобы читатель имел некоторое представление о разнице между терминами «топология» и «геометрия»: геометрия изучает расстояния и углы, тогда как топология занимается более общими свойствами. Например, две сферы с различными радиусами являются топологически эквивалентными, а геометрически — нет. Два объекта топологически эквивалентны, если один из них может быть получен из другого искривлением и растяжением последнего без разрыва (чтобы быть более точными, путем использования непрерывного отображения, обратное к которому также непрерывно), тогда как в геометрии эквивалентные объекты должны быть идентичны во всех отношениях, за исключением их положения и ориентации в пространстве. Теория графов является частью топологии, поскольку вершины не обладают свойством положения в пространстве и топология графа есть отношение ребер.

Обычно удобно запоминать в памяти компьютера два различных множества данных, относящихся к рассматриваемому объекту,— топологические данные и геометрические данные. Например, можно представить широкий

класс объектов при помощи каркасных моделей. На рис. 10.1 дано такое представление конечного цилиндра. Топология модели такого типа может рассматриваться как граф и может быть представлена в виде связной списочной структуры. Геометрические данные могут быть просто списком векторов, определяющих положения вершин в $R^3$. Целью разделения содержания геометрии и топологии является то, что мы хотим преобразовать нашу модель некоторым образом, например чтобы она была физически меньше, или передвинуть ее в некоторое новое положение, изменив ориентацию в пространстве. Такие преобразования не изменяют топологии модели, и нам просто изменить соответствующим образом геометрическое множество данных. В § 1, 2 мы будем заниматься системами координат, которые дают возможность представить геометрические данные, и некоторыми полезными в дальнейшем множествами преобразований, которые могут применяться к геометрическим множествам данных.

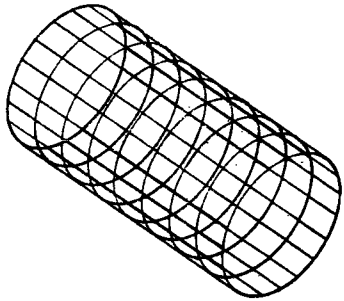


Рис. 10.1

## § 1. Системы координат для подмножеств $R^3$

В общем случае система координат (координатная система) или параметризация множества $S \subseteq R^n$ является идентификацией каждой точки в $S$ при помощи единственного упорядоченного набора чисел $(\xi_1, \ldots, \xi_q) \in R^q$ ($q \in N$). В терминологии отображений система координат для $S$ может быть определена как непрерывная биекция $\Phi$: $S \to \mathscr{S}$, где $\mathscr{S} \subseteq R^q$. Любое преобразование или другое вычисление, осуществляемое на $S$, тогда выполняется в терминах координат $(\xi_1, \ldots, \xi_q)$. Конкретное отображение, которое выбирается для данной задачи, часто будет определяться геометрией пространства $S$. Можно показать, что $q$ для фиксированного $S$ является постоянным для всех систем координат и называется *размерностью* $S$. К сожалению, в одном из приведенных ниже примеров (однородные системы координат) требуется, чтобы $\mathscr{S}$ было всем пространством; кроме того, существует много по-

лезных подмножеств, для которых такие отображения не существуют (например, круг или сфера). Чтобы разобраться с этим, следовало бы совершить небольшой экскурс в топологию; однако вместо этого будем надеяться, что проведенное ниже рассуждение, несмотря на недостатки определения, поможет хорошо разобраться в данном вопросе.

В основном нас будут интересовать пространства размерности 3 и меньше; например, кривые являются одномерными объектами, а поверхности имеют размерность 2. В следующих примерах представлены некоторые наиболее общие координатные системы в $\mathbf{R}^2$ и $\mathbf{R}^3$, однако там, где легко обсудить более общий случай, мы это будем делать.

Пример 1.1. Прямоугольная система координат в $\mathbf{R}^n$ ($n \in \mathbf{N}$). Если $B = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ — базис в $\mathbf{R}^n$, то каждый элемент $\mathbf{x} \in \mathbf{R}^n$ может быть записан единственным образом в виде

$$\mathbf{x} = \sum_{i=1}^{n} a_i \mathbf{e}_i, \text{ где } a_i \in \mathbf{R}, \quad 1 \leqslant i \leqslant n.$$

Отображение $\Phi$: $\mathbf{R}^n \to \mathbf{R}^n$, задаваемое как $\Phi(\mathbf{x}) = (a_1, \ldots, a_n)$, определяет координатную систему в $\mathbf{R}^n$. Если $B$ ортонормирован, то соответствующие координаты называются *декартовыми координатами* в $\mathbf{R}^n$. В $\mathbf{R}^2$ и $\mathbf{R}^3$ обычно ортонормированный базис интерпретируют геометрически как правостороннюю систему координат в смысле § 4 гл. 5. //

Следующий пример в $\mathbf{R}^n$ является более сложным с математической точки зрения. Необходимо некоторое предварительное обсуждение перед тем, как мы опишем эту систему.

Начнем с определения отношения $\sim$ на $\mathbf{R}^{n+1}$. Определим его следующим образом: $\mathbf{x} \sim \mathbf{y}$, если $\mathbf{x} = \alpha \mathbf{y}$ для некоторого $\alpha \in \mathbf{R} \setminus \{0\}$. Важные свойства этого отношения сформулированы в следующем предложении, доказательство которого оставляется в качестве упражнения.

Предложение. *Отношение $\sim$ является отношением эквивалентности на $\mathbf{R}^{n+1}$.*

Отношение $\sim$ определяет следующие классы эквивалентности:

$\mathbf{P}^n = \{L \setminus \{0\}$: $L$ — одномерное векторное подпространство $\mathbf{R}^{n+1}$, $0$ — нулевой вектор в $\mathbf{R}^{n+1}\}$ и $\{0\}$.

Очевидно, что $\mathbf{R}^{n+1} = \mathbf{P}^n \cup \{0\}$. Определим подмножество $\mathbf{P}^n$.

Определение. Пусть $\mathbf{x} \in \mathbf{R}^{n+1}$ имеет вид $(x_1, \ldots, x_n, 0)$, где не все $x_i$ равны нулю. Тогда $[\mathbf{x}] \in \mathbf{P}^n$ называют *бесконечно удаленной точкой*. Обозначим множество всех бесконечно удаленных точек $\mathbf{P}^n$ через $\mathbf{L}^n_\infty$ и определим $H^n$ как $\mathbf{P}^n \setminus \mathbf{L}^n_\infty$. //

Следующий результат играет важную роль: он устанавливает координатное отображение $\mathbf{R}^n$.

Предложение. *Существует биекция* $Q^n$: $H^n \to \mathbf{R}^n$.

Доказательство. Определим отношение $Q^n$: $H^n \to \mathbf{R}^n$ следующим образом: $Q^n([(x_1, \ldots, x_{n+1})]) = \frac{1}{x_{n+1}}(x_1, \ldots, x_n)$. Вначале заметим, что правая часть всегда определена, так как $L^n_\infty$ исключена из области определения $Q^n$; следовательно, $\mathscr{D}(Q^n) = H^n$. Необходимо установить три факта:

$Q^n$ — отображение на $H^n$;

$Q^n$ инъективно;

$Q^n$ *сюръективно.*

$Q^n$ является отображением, если

$$(x_1, \ldots, x_{n+1}) \sim (y_1, \ldots, y_{n+1}) \Rightarrow$$
$$\Rightarrow Q^n([(x_1, \ldots, x_{n+1})]) = Q^n([(y_1, \ldots, y_{n+1})]).$$

Пусть $(x_1, \ldots, x_{n+1}) \sim (y_1, \ldots, y_{n+1})$; тогда существует $\alpha \in \mathbf{R}\setminus\{0\}$ такое, что $x_i = \alpha y_i$ для всех $i$, $1 \leqslant i \leqslant n+1$. Тогда по определению $Q^n([(x_1, \ldots, x_{n+1})]) = Q^n([\alpha y_1, \ldots, \alpha y_{n+1})]) = \frac{1}{\alpha y_{n+1}}(\alpha y_1, \ldots, \alpha y_n) = \frac{1}{y_{n+1}}(y_1, \ldots, y_n) = Q^n([(y_1, \ldots, y_{n+1})])$. Следовательно, $Q^n$ — отображение на $H^n$.

$Q^n$ инъективно, если $Q^n([\mathbf{x}]) = Q^n([\mathbf{y}]) \Rightarrow \mathbf{x} \sim \mathbf{y}$. Записывая выражение для $\mathbf{x}$ и $\mathbf{y}$ в виде $\mathbf{x} = (x_1, \ldots, x_n, x_{n+1})$ и $\mathbf{y} = (y_1, \ldots, y_n, y_{n+1})$, получаем

$$Q^n([\mathbf{x}]) = Q^n([\mathbf{y}]) \Rightarrow \frac{1}{x_{n+1}}(x_1, \ldots, x_n) = \frac{1}{y_{n+1}}(y_1, \ldots, y_n),$$

откуда $x_i = (x_{n+1}/y_{n+1})y_i \Rightarrow x \sim y$, $\alpha = x_{n+1}/y_{n+1}$. Следовательно, $Q^n$ инъективно.

$Q^n$ сюръективно, если для всех $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{R}^n$ существует $\mathbf{x}^* \in \mathbf{R}^{n+1}$ такое, что $Q^n([\mathbf{x}^*]) = \mathbf{x}$. Если определим $\mathbf{x}^* = (x_1, \ldots, x_n, 1) \in \mathbf{R}^{n+1}$, тогда очевидно, что $Q^n([\mathbf{x}^*]) = \mathbf{x}$, т. е. $Q^n$ сюръективно. //

Пример 1.2. Однородные координаты в $\mathbf{R}^n (n \in \mathbf{N})$. *Однородные координаты в* $\mathbf{R}^n$ определяют как отображение $\mathbf{R}^n$ на $H^n$, имеющее обратное отображение $Q^n$, так что $\Phi$: $\mathbf{R}^n \to H^n$, где $\Phi = (Q^n)^{-1}$. Другими словами, однородными координатами точки $(x_1, \ldots, x_n) \in \mathbf{R}^n$ являются $(n+1)$-мерные наборы классов эквивалентности

$$[(x_1, \ldots, x_n, 1)] = \{(px_1, \ldots, px_n, p): p \neq 0\}.$$

Элемент из $[(x_1, \ldots, x_n, 1)]$ называют *однородным представлением* для $(x_1, \ldots, x_n)$.

Часто бывает удобно представить в компьютере геометрические данные в однородной форме (см. § 3). Чтобы из однородных координат получить физические, осуществим отображение $(x_1, \ldots, x_{n+1}) \mapsto \frac{1}{x_{n+1}}(x_1, \ldots, x_n)$. Сформулированное выше предложение показывает, что все однородные представления данной физической точки имеют *те же самые* физические координаты.

Чтобы прояснить ситуацию, рассмотрим геометрическую интерпретацию однородных координат в R. Случай более высоких размерностей имеет подобную геометрическую интерпретацию, однако ее нелегко изобразить на рисунке. Элементы $\mathbf{P}^1$ являются бесконечными линиями в $\mathbf{R}^2$, проходящими через начало координат, однако начало координат выброшено (рис. 10.2). Ясно, что единственной, бесконечно удаленной точкой будет ось $Ox_1$ с выброшенным началом координат. Рис. 10.3 проясняет понятие «бесконечно удаленная точка в $\mathbf{P}^1$». Точки на линии $L_n = [(n, 1)]$ являются однородным представлением $n \equiv \mathbf{R}$. Из рис. 10.3 видно, что при $n \to \infty$ прямая $L_n$ стремится к бесконечно удаленной точке $L_\infty$. //

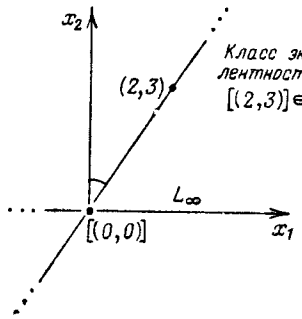


Рис. 10.2

Пример 1.3. Полярные координаты в $\mathbf{R}^2$. Пусть $L \subseteq \mathbf{R}^2$ — полубесконечная линия $L = \{(x, 0): x \geqslant 0\}$ и $(x, y) \equiv \mathbf{R}^2 \backslash L$. Тогда, если $\Phi$: $\mathbf{R}^2 \backslash L \to ]0, \infty[ \times ]0, 2\pi[$ определено как $\Phi(x, y) = (r, \theta)$, где $r = (x^2 + y^2)^{1/2}$, а $\theta = \operatorname{arctg}(y/x)$, то $\Phi$ определяет множество полярных ко-

ординат в $\mathbf{R}^2$. Обратное к $\Phi$ отображение задается соотношением $\Phi^{-1}(r, \theta) = (r\cos\theta, r\sin\theta)$. Геометрическая интерпретация $r$ и $\theta$ дана на рис. 10.4. Исключение $L$ из области определения $\Phi$ часто неправильно комментируется в элементарных учебниках. Линию $L$ удаляют, чтобы получить непрерывность $\Phi$.

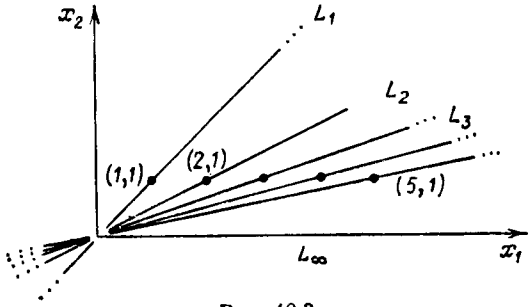



Рис. 10.3

Пример 1.4. Цилиндрические координаты в $\mathbf{R}^3$. Пусть $P \subset \mathbf{R}^3$ — полуплоскость $P = \{(x, 0, z): x \geqslant 0\}$ и

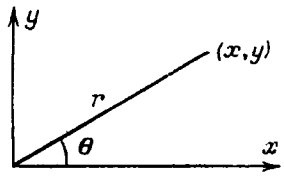



Рис. 10.4

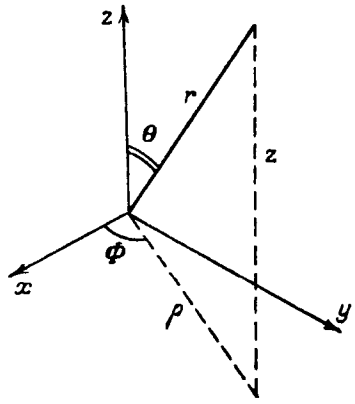



Рис. 10.5

$(x, y, z) \in \mathbf{R}^3 \backslash P$. Тогда отображение $\Phi$: $\mathbf{R}^3 \backslash P \to$ $\to ]0, \infty[ \times ]0, 2\pi[ \times \mathbf{R}$, определяемое как $\Phi(x, y, z) =$ $= (\rho, \varphi, z)$, где $\rho = (x^2 + y^2)^{1/2}$, $\varphi = \operatorname{arctg}(y/x)$, определяет множество *цилиндрических координат* в $\mathbf{R}^3 \backslash P$. Обратное к $\Phi$ отображение задается формулой $\Phi^{-1}(\rho, \varphi, z) = (\rho\cos\varphi, \rho\sin\varphi, z)$. На рис. 10.5 изображены величины $\rho$, $\varphi$, $z$, относящиеся к осям декартовой системы ко-

ординат. Полуплоскость $P$ исключена из области определения Ф, чтобы получить непрерывность. //

Пример 1.5. Сферическая система координат в $\mathbf{R}^3 \backslash P$; $P$ определено, как в примере 1.4. *Сферическую систему координат* определяют как отображение

$$\Phi\colon \mathbf{R}^3/P \to ]0, \infty[ \times [0, \pi] \times ]0,2\pi[,$$

где $\Phi(x, y, z) = (r, \theta, \varphi)$ а $r = (x^2 + y^2 + z^2)^{1/2}$, $\theta = \arccos(z/r)$, $\varphi = \operatorname{arctg}(y/\ )$.

## § 2. Преобразования

Одной из наиболее часто используемых операций над геометрическими данными является преобразование. Например, вращение трехмерного пространства $\mathbf{R}^3$ дает возможность рассматривать трехмерные геометрические объекты с любой удобной точки. Это может оказаться полезным при проверке геометрической целостности объекта, или при подробном изучении какого-либо свойства, или по какой-либо другой причине. Некоторые подходы основаны на библиотеках осповных геометрических «строительных блоков», из которых путем преобразования могут быть построены более сложные геометрии. Они следуют из некоторых теоретико-множественных операций, которые будут построены далее. Алгоритмы для такого типа геометрического синтеза являются весьма специальными и не будут здесь обсуждаться, однако мы упомянем, что обобщенная форма теоремы Эйлера для графов может быть использована для исследования топологии конструируемого объекта. Преобразования используются более широко. Некоторые из наиболее употребляемых преобразований и их линейные представления будут сейчас описаны.

**2.1. Преобразования в $\mathbf{R}^2$.** *Переносом* в $\mathbf{R}^2$ называется отображение вида $T\mathbf{r} = \mathbf{r} + \mathbf{r}_0$ для всех $\mathbf{r} \in \mathbf{R}^2$, где $\mathbf{r}_0 \in \mathbf{R}^2$ — фиксированный вектор. Перенос перемещает всю плоскость сдвигом на фиксированный вектор и может быть записан в координатной форме:

$$T(x, y) = (x, y) + (x_0, y_0) = (x + x_0, y + y_0).$$

Обозначим через $T(2)$ множество всех переносов па $\mathbf{R}^2$ и определим произведение в $T(2)$, используя композицию отображений. Пусть $T_1$ — перенос на вектор $\mathbf{r}_1$, а $T_2$ — перенос на вектор $\mathbf{r}_2$. Тогда произведение $T_2 \circ T_1$ опреде-

лим как

$$T_2 \circ T_1\mathbf{r} = T_2(T_1\mathbf{r}) = \mathbf{r} + \mathbf{r}_1 + \mathbf{r}_2.$$

Очевидно, что $T_2 \circ T_1$ есть перенос на $\mathbf{r}_1 + \mathbf{r}_2$, и, следовательно, операция $\circ$ является бинарной на $T(2)$. В действительности справедливо более сильное утверждение.

Предложение. $(T(2), \circ)$ — *коммутативная группа нелинейных преобразований* $\mathbf{R}$.

Доказательство. Замкнутость относительно операции $\circ$ уже доказана. Аксиомы группы следуют из групповой структуры $(\mathbf{R}^2, +)$; например, если $T_i (1 \leqslant i \leqslant 3)$ — переносы на $\mathbf{r}_n$, то

$$\begin{aligned}(T_3 \circ (T_2 \circ T_1))r &= T_3((T_2 \circ T_1)r) = T_3(r + r_1 + r_2) = \\ &= \mathbf{r} + \mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3 = (\mathbf{r} + \mathbf{r}_1) + (\mathbf{r}_2 + \mathbf{r}_3) = \\ &= (T_3 \circ T_2)(T_1\mathbf{r}) = ((T_3 \circ T_2) \circ T_1)\mathbf{r};\end{aligned}$$

поэтому $(T(2), \circ)$ ассоциативна. Единицей является перенос на вектор $(0, 0)$, а обратным переносом к $T\mathbf{r} = \mathbf{r} + \mathbf{r}_0$ будет $T'\mathbf{r} = \mathbf{r} - \mathbf{r}_0$; поэтому

$$T \circ T'\mathbf{r} = T'T\mathbf{r} = \mathbf{r} \text{ для всех } \mathbf{r} \in \mathbf{R}^2.$$

Коммутативность также выполняется, поскольку

$$(T_2 \circ T_1)\mathbf{r} = \mathbf{r} + \mathbf{r}_1 + \mathbf{r}_2 = \mathbf{r} + \mathbf{r}_2 + \mathbf{r}_1 = (T_1 \circ T_2)\mathbf{r}.$$

В общем случае, если $T \in T(2)$, то $T(0, 0) \neq (0, 0)$; поэтому начало координат смещается и, следовательно, перенос является нелинейным отображением. //

На практике это означает, что можно применить любое число переносов к геометрическому объекту и получить один и тот же результат. Кроме того, нелинейный характер элементов $T(2)$ означает, что мы не можем выполнить $T(2)$ при помощи элементов из $\mathcal{M}(2, \mathbf{R})$; другими словами, уравнение

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + x_0 \\ y + y_0 \end{bmatrix} \text{ для всех } (x, y) \in \mathbf{R}^2$$

не имеет решения. Следствия из этого факта будет рассмотрены ниже.

Опишем поворот в $\mathbf{R}^2$, используя рис. 10.6. Интуитивно ясно, что поворот на угол $\theta$, обозначаемый $W(\theta)$, отображает точку $(x, y)$ в точку $(x', y')$, где $\|\mathbf{r}\| = \|\mathbf{r}'\| = r$. Из рисунка видно, что

$$x = r\cos\theta,\ y = r\sin\theta,$$

$$x' = r\cos(\theta + \theta_1),\ y' = r\sin(\theta + \theta_1).$$

Применяя тригонометрические формулы, получаем

$$x' = r(\cos\theta\cos\theta_1 - \sin\theta\sin\theta_1) = x\cos\theta - y\sin\theta,$$
$$y' = x\sin\theta + y\cos\theta,$$

так что

$$W(\theta)(x, y) = (x\cos\theta - y\sin\theta,\ x\sin\theta + y\cos\theta).$$

В матричной форме это можно записать следующим образом:

$$W(\theta)(x, y) = \begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x \\ y\end{bmatrix};$$

$W(\theta)$ геометрически соответствует повороту вокруг начала координат $O$.

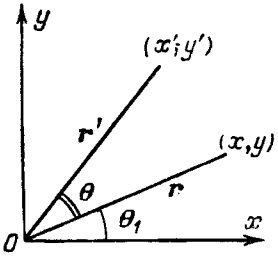


Рис. 10.6

Следующее предложение непосредственно вытекает из представления $\mathscr{M}(2, \mathbf{R})$ и, с нашей точки зрения, суммирует основные свойства поворотов.

Предложение.

а) *Преобразование* $W(\theta)$:

— *линейно*,

— *ортогонально*, $\det W(\theta) = 1$.

б) *Множество* $\{W(\theta): 0 \leqslant \theta < 2\pi\}$ *является коммутативной группой по отношению к композициям отображений* (*умножением матриц*). //

С практической точки зрения из а) следует, что для вычисления преобразования, обратного к повороту, достаточно лишь транспонировать матрицу. Часть б) означает, что любое число поворотов может применяться к множеству геометрических данных в любом порядке и будет давать одинаковые результаты. На самом деле верно обратное к а) утверждение. В частности, если $W \in \mathscr{M}(2, \mathbf{R})$ и выполнены условия, а) $W$ является поворотом. Другими словами, мы можем определить поворот в $\mathbf{R}^2$ как элемент группы $SO(2)$.

Хотя $T(2)$ и $SO(2)$ являются коммутативными группами, преобразования из $T(2)$ не коммутируют с преобразованиями из $SO(2)$, так как если $W \in SO(2)$ и $T \in T(2)$ — сдвиг на $\mathbf{r}_0$, то мы имеем $TW\mathbf{r} = W\mathbf{r} + \mathbf{r}_0$ и $WT\mathbf{r} = W(\mathbf{r} + \mathbf{r}_0) = W\mathbf{r} + W\mathbf{r}_0 \neq TW\mathbf{r}$ при $\mathbf{r}_0 \neq (0, 0)$. На практике это означает, что преобразования переноса и поворота надо применять строго в том порядке, в каком они записаны. Некоммутативность проиллюстрирована на

рис. 10.7, где показаны результаты применения преобразований $T \cdot W(\pi/2)$ и $W(\pi/2) \cdot T$ к полусфере. Из рис. 10.7 видно, что если площадь $\{(x, y): 0 \leqslant x \leqslant a, 0 \leqslant y \leqslant b\}$ соответствует экрану графического терминала, то результат применения $W(\pi/2) \cdot T$ к полукругу даст в итоге пустой экран.

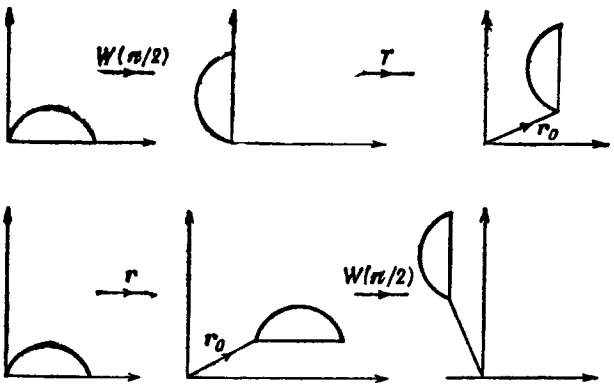



Рис. 10.7

Группы $T(2)$ и $SO(2)$ могут быть объединены в виде третьего множества $E(2)$ преобразований плоскости, называемых *евклидовыми*. Элементы $U \in E(2)$ имеют вид $U\mathbf{r} = W\mathbf{r} + \mathbf{r}_0$ для всех $\mathbf{r} \in \mathbf{R}^2$, где $W \in SO(2)$, а $\mathbf{r}_0 \in \mathbf{R}^2$ — фиксированный вектор. Эти преобразования часто записывают в виде пар вида $(W, \mathbf{r}_0)$. Если $(W_1, \mathbf{r}_1) \in E(2)$ и $(W_2, \mathbf{r}_2) \in E(2)$, то $(W_2, \mathbf{r}_2) \cdot (W_1, \mathbf{r}_1)\mathbf{r} = (W_2, \mathbf{r}_2)(W_1\mathbf{r} + \mathbf{r}_1) = W_2(W_1\mathbf{r} + \mathbf{r}_1) + \mathbf{r}_2 = (W_2W_1, W_2\mathbf{r}_1 + \mathbf{r}_2)\mathbf{r}$, так что

$$(W_2, \mathbf{r}_2) \cdot (W_1, \mathbf{r}_1) = (W_2W_1, W_2\mathbf{r}_1 + \mathbf{r}_2) \in E(2)$$

и $E(2)$ замкнуто. В действительности имеет место следующее предложение.

Предложение. *$E(2)$ является группой по отношению к операции композиции.*

Доказательство. Ограничимся указанием основных моментов доказательства. Замыкание уже доказано. Ассоциативность следует из применения преобразований несколько раз. Если $I \in GL(2, \mathbf{R})$ — единица, то справедливы соотношения

$$(I, 0) \cdot (W, \mathbf{r}_0) = (W, \mathbf{r}_0) \cdot (I, 0) = (W, \mathbf{r}_0)$$

для всех $(W, \mathbf{r}_0) \in E(2)$;

следовательно, $(I, 0)$ является единицей в $E(2)$. Обратный

элемент $(W, \mathbf{r}_0)^{-1}$ к $(W, \mathbf{r}_0)$ будет равен $(W^T, -W^T\mathbf{r}_0)$, при $(W^T, -W^T\mathbf{r}_0) \in E(2)$, и

$$(W, \mathbf{r}_0) \circ (W^T, -W^T\mathbf{r}_0) = (I, 0) = (W^T, -W^T\mathbf{r}_0) \circ (W, \mathbf{r}_0). \text{ //}$$

$SO(2)$ и $T(2)$ являются подгруппами $\{(W, 0)\colon W \in SO(2)\}$ и $\{(I, \mathbf{r}_0)\colon \mathbf{r}_0 \in \mathbf{R}^2\}$ из $E(2)$ соответственно, и каждый элемент $(W, \mathbf{r}_0) \in E(2)$ может быть представлен следующим образом:

$$(W, \mathbf{r}_0) = (I, \mathbf{r}_0) \circ (W, 0).$$

$E(2)$ — некоммутативная группа, так как

$$(W_2, \mathbf{r}_2) \circ (W_1, \mathbf{r}_1) = (W_2W_1, W_2\mathbf{r}_1 + \mathbf{r}_2),$$

$$(W_1, \mathbf{r}_1) \circ (W_2, \mathbf{r}_2) = (W_1W_2, W_1\mathbf{r}_2 + \mathbf{r}_1) \neq (W_2, \mathbf{r}_2) \circ (W_1, \mathbf{r}_1).$$

В приложениях компьютерной графики часто хотят повернуть объект около некоторой фиксированной точки, отличной от начала координат. Например, на рис. 10.8

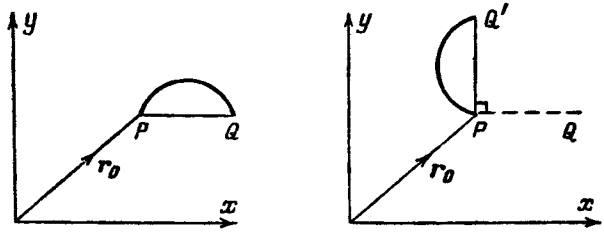



Рис. 10.8

полукруг повернут на $\pi/2$ около точки $P$. Общая ошибка при этом заключается в попытке применить преобразование $W(\pi/2)$; однако это приводит к результату, изображенному на рис. 10.9. Проиллюстрируем последовательность преобразований поворота вокруг фиксированной точки $\mathbf{r}_0 \in \mathbf{R}^2$ на рис. 10.10, используя понятия $E(2)$. Требуемое преобразование, таким образом, будет равно

$$(I, \mathbf{r}_0) \circ (W(\pi/2), 0) \circ (I, -\mathbf{r}_0).$$

Эту запись можно упростить, используя правило произведения в $E(2)$; получим

$$(W(\pi/2), -W(\pi/2)\mathbf{r}_0 + \mathbf{r}_0).$$

Ясно, что применять составное преобразование к множеству геометрических данных более эффективно, чем применять в отдельности каждое преобразование. $E(2)$ дает

возможность расположить объект в любой точке плоскости и в требуемой ориентации.

Множество полезных операций на плоскости включает в себя преобразование масштаба. Преобразованием масштаба на $\mathbf{R}^2$ называют отображение вида

$$S(x, y)=(\lambda x, \mu y), \ \lambda, \mu>0,$$

или же (в матричной форме)

$$S(x, y)=\begin{bmatrix}\lambda & 0\\ 0 & \mu\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}.$$

Множество всех преобразований такого типа обозначается

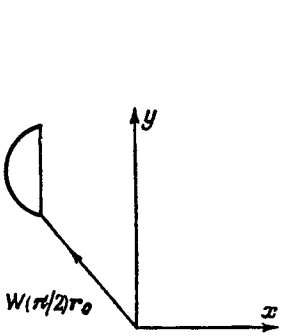



Рис. 10.9

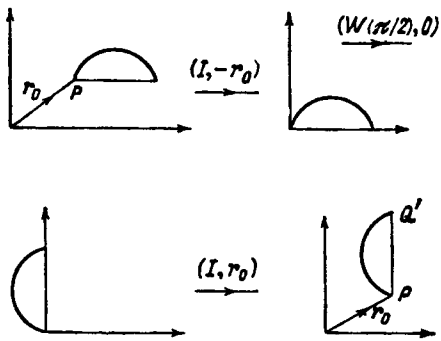



Рис. 10.10

$S(2)$. Важные свойства $S(2)$ сформулированы в следующем предложении, доказательство которого оставляем в качестве упражнения.

Предложение. *$S(2)$ является коммутативной группой линейных преобразований по отношению к операции композиции. (Другими словами, $S(2)$ является коммутативной подгруппой $GL(2, \mathrm{R})$.)* //

Преобразование масштаба не коммутирует с переносом, так как если $T$ — перенос на вектор $r_0$ и

$$S(x, y)=(\lambda x, \mu y),$$

то

$$S \circ T(x, y)=S(x+x_0, y+y_0)=(\lambda(x+x_0), \mu(y+y_0)),$$
$$T \circ S(x, y)=T(\lambda x, \mu y)=(\lambda x+x_0, \mu y+y_0) \neq S \circ T(x, y)$$

в общем случае некоммутативность проиллюстрирована

на рис. 10.11; используются преобразование масштаба

$$S(x, y) = (2x, 2y)$$

и перенос

$$T(x, y) = (x + 1, y + 1)$$

единичного квадрата с вершиной в начале координат. Аналогично можно показать, что преобразования масштаба не коммутируют с поворотами. Доказательство этого

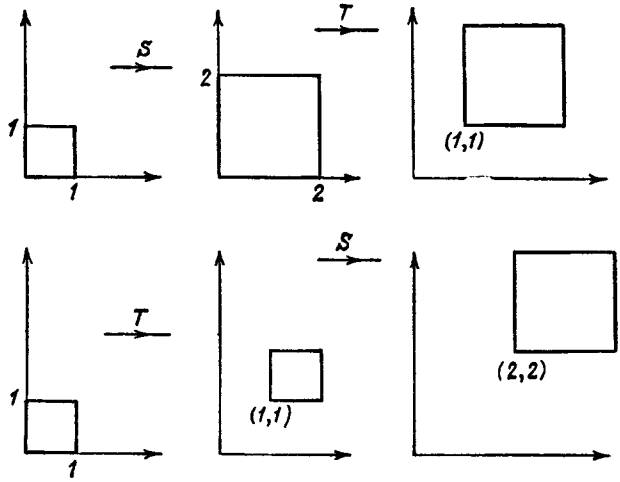



Рис. 10.11

и построение соответствующего рисунка, который это демонстрирует, оставляем в качестве упражнения.

На практике это означает, что если преобразование масштаба комбинируют с переносом или с поворотом, то следует обращать особое внимание на порядок применения преобразований.

Языки высокого уровня предусматривают массивы как структуры данных; следовательно, матричное представление всех рассмотренных выше преобразований должно существенно упростить их применение. Например, все произведения преобразований тогда могли бы вычисляться путем умножения матриц (см. гл. 6) вместо того, чтобы работать с определениями этих преобразований. Нелинейная природа переносов препятствует представлению в $\mathscr{M}(2, \mathbf{R})$; однако возможно получить подходящее представление обсуждаемых выше преобразований в $\mathscr{M}(3, \mathbf{R})$,

где матрицы оперируют в пространстве $H^2$ однородных координат $\mathbf{R}^2$. Поскольку техника получения представления в однородных координатах носит весьма общий характер и, в частности, может быть применена к преобразованиям в $\mathbf{R}^3$ (где получается представление в $\mathscr{M}(4, \mathbf{R})$), мы отложим рассмотрение до описания преобразований в $\mathbf{R}^3$.

**2.2. Преобразования в $\mathbf{R}^3$.** *Перенос в* $\mathbf{R}^3$ есть отображение $T$: $\mathbf{R}^3 \to \mathbf{R}^3$ вида

$$T\mathbf{r} = \mathbf{r} + \mathbf{r}_0 \text{ для всех } \mathbf{r} \in \mathbf{R}^3,$$

где $\mathbf{r}_0 \in \mathbf{R}^3$ — фиксированный вектор. Если $\mathbf{r}_0 = (x_0, y_0, z_0)$, то перенос на $\mathbf{r}_0$ может быть записан в компонентной форме как

$$T(x, y, z) = (x + x_0, y + y_0, z + z_0).$$

Пусть $T(3)$ обозначает множество всех переносов $\mathbf{R}^3$. Тогда $T(3)$ образует коммутативную группу (по отношению к операции композиции) нелинейных преобразований $\mathbf{R}^3$, изоморфную группе $(\mathbf{R}^3, +)$. Следствием этого является тот факт, что последовательность переносов $\mathbf{R}^3$ может применяться в произвольном порядке и будет давать один и тот же результат и что мы не можем выполнить $T(3)$ при помощи элементов из $\mathscr{M}(3, \mathbf{R})$.

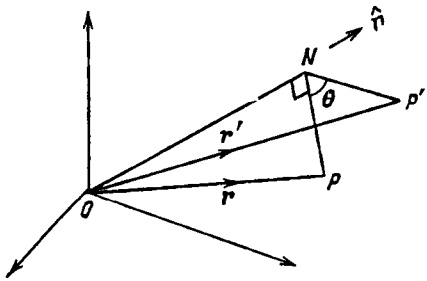



Рис. 10.12

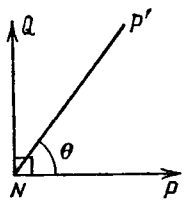



Рис. 10.13

Группа поворотов в $\mathbf{R}^3$ имеет более сложную структуру, чем $SO(2)$. Рассмотрим поворот $W_{\hat{\mathbf{n}}}(\theta)$ на угол $\theta$ вокруг оси, определяемой единичным вектором $\hat{\mathbf{n}} \in \mathbf{R}^3$, как это изображено на рис. 10.12. Тогда

$$W_{\hat{\mathbf{n}}}(\theta)\,\mathbf{r} = \mathbf{r}',$$

где $\mathbf{r}$ — вектор с концом в точке $P$, а $\mathbf{r}'$ — преобразованный вектор с концом в точке $P'$. Чтобы определить явно

$W_{\hat{\mathbf{n}}}(\theta)$, мы должны выразить $\mathbf{r}'$, как функцию $\mathbf{r}$, $\hat{\mathbf{n}}$ и $\theta$. Ясно, что если $\mathbf{r}_N$ — вектор, определяемый точкой $N$, лежащей на оси поворота, то

$$\mathbf{r}_N = (\hat{\mathbf{n}};\ \mathbf{r})\hat{\mathbf{n}},$$

и если $\mathbf{r}_{NP}$ — вектор, соединяющий $N$ и $P$, то

$$\mathbf{r}_{NP} = \mathbf{r} - \mathbf{r}_N = \mathbf{r} - (\hat{\mathbf{n}} \cdot \mathbf{r})\hat{\mathbf{n}} = (\hat{\mathbf{n}} \times \mathbf{r}) \times \hat{\mathbf{n}}.$$

Рисунок 10.13 показывает, как выглядит поворот из точки $N$ в направлении начала координат $O$; $Q$ — точка, полученная поворотом точки $P$ на $\pi/2$ относительно оси $ON$. Таким образом, $W_{\hat{\mathbf{n}}}(\theta)$ — поворот; следовательно,

$$\|\mathbf{r}_{NQ}\| = \|\mathbf{r}_{NP}\| = \|\mathbf{r}_{NP'}\|,$$
$$\mathbf{r}_{NP'} = \mathbf{r}_{NQ} \sin\theta + \mathbf{r}_{NP} \cos\theta,$$

где $\mathbf{r}_{NQ}$ — проекция вектора на ось $NQ$. Отсюда следует, что

$$\mathbf{r}_{NQ} = \hat{\mathbf{n}} \times \mathbf{r}_{NP} = \hat{\mathbf{n}} \times ((\hat{\mathbf{n}} \times \mathbf{r}) \times \hat{\mathbf{n}}) = \hat{\mathbf{n}} \times \mathbf{r}$$

(см. упражнение 10.1); следовательно,

$$\mathbf{r}_{NP'} = (\hat{\mathbf{n}} \times \mathbf{r}) \sin\theta + ((\hat{\mathbf{n}} \times \mathbf{r}) \times \hat{\mathbf{n}}) \cos\theta =$$
$$= (\hat{\mathbf{n}} \times \mathbf{r}) \sin\theta - (\hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{r})) \cos\theta,$$

но

$$W_{\hat{\mathbf{n}}}(\theta)\mathbf{r} = \mathbf{r}' = \mathbf{r}_N + \mathbf{r}_{NP'} =$$
$$= \mathbf{r} + (\hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{r}))(1 - \cos\theta) + (\hat{\mathbf{n}} \times \mathbf{r}) \sin\theta.$$

Предложение. $W_{\hat{\mathbf{n}}}(\theta) \in SO(3)$.

Доказательство. Некоторые детали доказательства будем оставлять в качестве упражнений. Запишем $W_{\hat{\mathbf{n}}}(\theta)$ в несколько иной форме. Если $\hat{\mathbf{n}} \in \mathbf{R}^3$ определяет ось поворота, то определим преобразование $A_{\hat{\mathbf{n}}}$: $\mathbf{R}^3 \to \mathbf{R}^3$ следующим образом:

$$A_{\hat{\mathbf{n}}}\mathbf{r} = \hat{\mathbf{n}} \times \mathbf{r};$$

$A_{\hat{\mathbf{n}}}$ является непрерывным и антисимметричным (см. упражнение 10.1). Тогда

$$A_{\hat{\mathbf{n}}}^2\mathbf{r} = A_{\hat{\mathbf{n}}}(A_{\hat{\mathbf{n}}}\mathbf{r}) = A_{\hat{\mathbf{n}}}(\mathbf{n} \times \mathbf{r}) = \hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{r});$$

$A_{\hat{\mathbf{n}}}^2$ линейно и симметрично. Сейчас мы можем записать

$W_{\hat{\mathbf{n}}}(\theta)$ как сумму

$$W_{\hat{\mathbf{n}}}(\theta) = I + (1 - \cos\theta) A_{\hat{\mathbf{n}}}^2 + \sin\theta A_{\hat{\mathbf{n}}}$$

в $\mathscr{M}(3, \mathbf{R})$. Но $\mathscr{M}(3, \mathbf{R})$ — векторное пространство; следовательно, $W_{\hat{\mathbf{n}}}(\theta)$ линейно. Ортогональность следует из свойств $A_{\hat{\mathbf{n}}}$, так как

$$W_{\hat{\mathbf{n}}}(\theta)^T = I + (1 - \cos\theta)\left(A_{\hat{\mathbf{n}}}^2\right)^T + \sin\theta\left(A_{\hat{\mathbf{n}}}\right)^T =$$

$$= I + (1 - \cos\theta) A_{\hat{\mathbf{n}}}^2 - \sin\theta A_{\hat{\mathbf{n}}},$$

$$W_{\hat{\mathbf{n}}}(\theta) W_{\hat{\mathbf{n}}}(\theta)^T = \left[I + (1 - \cos\theta) A_{\hat{\mathbf{n}}}^2 + \sin\theta A_{\hat{\mathbf{n}}}\right] *$$

$$* \left[I + (1 - \cos\theta) A_{\hat{\mathbf{n}}}^2 - \sin\theta A_{\hat{\mathbf{n}}}\right] =$$

$$= I + [2(1 - \cos\theta) - \sin^2\theta] A_{\hat{\mathbf{n}}}^2 + (1 - \cos\theta)^2 A_{\hat{\mathbf{n}}}^4 =$$

$$= I + (1 - \cos\theta)^2\left(A_{\hat{\mathbf{n}}}^2 + A_{\hat{\mathbf{n}}}^4\right),$$

но (см. упражнение 10.1)

$$A_{\hat{\mathbf{n}}}^4 = -A_{\hat{\mathbf{n}}}^2.$$

Следовательно,

$$W_{\hat{\mathbf{n}}}(\theta) W_{\hat{\mathbf{n}}}(\theta)^T = I$$

и $W_{\hat{\mathbf{n}}}(\theta)$ ортогонально.

Доказательство того, что $\det W_{\hat{\mathbf{n}}}(\theta) = 1$, оставляем в качестве упражнения. //

Обратное утверждение также справедливо (см. задачу 12 упражнения 10.1); в частности, если $W \in SO(3)$, то можно показать, что существуют единичный вектор $\hat{\mathbf{n}} \in \mathbf{R}^3$ и угол $\theta$ в пределах $0 \leqslant \theta < 2\pi$ такой, что

$$W = W_{\hat{\mathbf{n}}}(\theta).$$

Следовательно, разумно определить *повороты* в $\mathbf{R}^3$ как группу $SO(3)$.

Определим некоторые подгруппы $SO(3)$. Часто требуется повернуть объект вокруг одной из декартовых осей координат. Соответствующие матрицы могут быть получены из общего вида $W_{\hat{\mathbf{n}}}(\theta)$. Например, чтобы осуществить поворот на угол $\theta$ вокруг оси $OZ$, выберем $\hat{\mathbf{n}} = \hat{\mathbf{k}} = (0, 0, 1)$, для которого

$$A_{\hat{\mathbf{k}}} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_{\hat{\mathbf{k}}}^2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

так что

$$W_{\hat{\mathbf{k}}}(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}.$$

Аналогично матрицы

$$W_{\hat{\mathbf{i}}}(\varphi)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\varphi & -\sin\varphi\\ 0 & \sin\varphi & \cos\varphi\end{bmatrix},\quad W_{\hat{\mathbf{j}}}(\chi)=\begin{bmatrix}\cos\chi & 0 & \sin\chi\\ 0 & 1 & 0\\ -\sin\chi & 0 & \cos\chi\end{bmatrix}$$

соответствуют поворотам на углы $\varphi$ и $\chi$ вокруг осей $OX$ и $OY$ соответственно. Доказательство этого оставляем в качестве упражнения. Сейчас легко показать, что $SO(3)$ — некоммутагивная группа, так как в общем случае

$$[W_{\hat{\mathbf{i}}}(\varphi), W_{\hat{\mathbf{j}}}(\chi)]\neq 0,\quad [W_{\hat{\mathbf{i}}}(\varphi), W_{\hat{\mathbf{k}}}(\theta)]\neq 0,$$
$$[W_{\hat{\mathbf{j}}}(\chi), W_{\hat{\mathbf{k}}}(\theta)]\neq 0.$$

Однако все подгруппы

$$\{W_{\hat{\mathbf{n}}}(\theta):\ 0\leqslant\theta<2\pi\},\quad \hat{\mathbf{n}}\in\mathbf{R}^3$$

коммутативны (см. упражнение 10.1).

С практической точки зрения из этих результатов следует, что для обращения поворота надо просто транспонировать матрицу, и если к некоторому геометрическому объекту применяют несколько поворотов, то порядок их применения важен.

Следуя двумерному случаю, объединим $T(3)$ и $SO(3)$ в виде множества $E(3)$ *евклидовых* преобразований $\mathbf{R}^3$. Элементы $E(3)$ имеют вид $U\mathbf{r}=W\mathbf{r}+\mathbf{r}_0$ для всех $\mathbf{r}\in\mathbf{R}^3$, где $W\in SO(3)$ и $\mathbf{r}_0\in\mathbf{R}^3$. Правило композиции в $E(3)$ записывается следующим образом:

$$(W_2,\ \mathbf{r}_2)\circ(W_1,\ \mathbf{r}_1)=(W_2W_1,\ W_2\mathbf{r}_1+\mathbf{r}_2);$$

по отношению к нему $E(3)$ становится некоммутативной группой нелинейных преобразований $\mathbf{R}^3$. Отсюда вытекают следствия, аналогичные двумерному случаю. В частности, подчеркнем, что $E(3)$ не может выполняться в $\mathcal{M}(3, \mathbf{R})$.

Преобразование *масштаба* в $\mathbf{R}^3$ является линейным отображением вида

$$S(x, y, z)=(\lambda x,\ \mu y,\ \sigma z),\ (x, y, z)\in\mathbf{R}^3,$$

где $\lambda, \mu, \sigma>0$. Множество $S(3)$ всех преобразований

масштаба определяет коммутативную группу по отношению к операции композиции.

Остальные преобразования, рассматриваемые в этом параграфе, имеют цели и свойства, отличные от рассмотренных выше. Нашей целью является задача представления трехмерного геометрического объекта на двумерном графическом терминале. С математической точки зрения мы ищем преобразования $\mathbf{R}^3$ в двумерные подпространства, в которых получаются необходимые графические представления нашего объекта. Напомним, что преобразование $P$ векторного пространства называют проекцией, если $P^2 = P$.

Пусть объект описывается в декартовой системе координат $(x, y, z)$ и в этой системе экран графического дисплея соответствует прямоугольному подмножеству

$$\{(x, y): 0 \leqslant x \leqslant a, 0 \leqslant y \leqslant b\}$$

на плоскости $xy$. Простейший метод получения образа объекта на экране — это применить преобразование $P_1\mathbf{r} = \mathbf{r}'$, где $\mathbf{r} = (x, y, z)$ и $\mathbf{r}' = (x, y, 0)$, к множеству геометрических данных. При условии что все значения $x$ и $y$ для множества данных находятся внутри экрана, получаем полную картину объекта. Если некоторые значения $x$ и(или) $y$ находятся вне пределов экрана, то надо вначале произвести подходящее преобразование масштаба. $P_1$ — линейная проекция, так как

$$P_1(x, y, z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

и $P_1^2\mathbf{r} = P_1\mathbf{r}$ для всех $\mathbf{r} \in \mathbf{R}^3$. Однако $P_1^{-1}$ не существует, так как

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

Геометрически $P_1$ можно получить следующим образом. Начертим линию сегмента (линию проекции) из точки $Q$, определяющей положение вектора $\mathbf{r}$, в точку $Q'$, определяющую положение $\mathbf{r}'$, в плоскости $xy$ так, чтобы линия отрезка $QQ'$ была ортогональна плоскости $xy$ (рис. 10.14). Тогда ясно, что

$$\mathbf{r}' = \mathbf{r} + (-z\hat{\mathbf{k}}) = (x, y, 0) = P_1\mathbf{r}.$$

$P_1$ называют *параллельной ортогональной проекцией* $\mathbf{R}^3$.

В параллельной проекции не отражается глубина получаемого образа; высота объекта появляется на экране такой же независимо от его расстояния до плоскости $xy$.

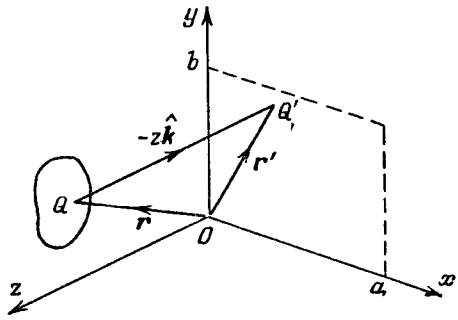



Рис. 10.14

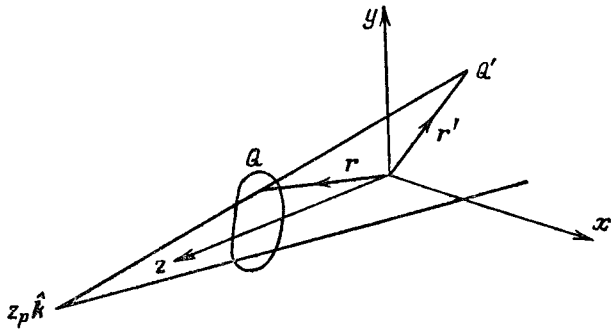



Рис. 10.15

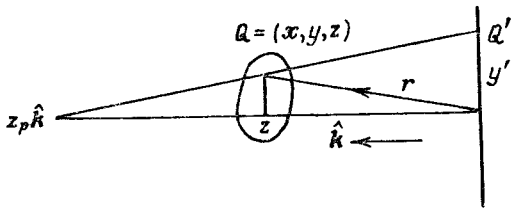



Рис. 10.16

Последнее преобразование, которое мы опишем, предназначено для того, чтобы дать глубину образа или «перспективу». Вместо линий параллельной проекции построим линии проекции, выходящие из некоторой фиксиро-

ванной точки. Определим проекцию $P_2\mathbf{r} = \mathbf{r}'$, где фиксированной точкой является конец вектора $z_p\hat{\mathbf{k}}$ (как показано на рис. 10.15) и, следовательно, точка $Q$ отображается в точку $Q'$. Рисунок 10.16 представляет вид рис. 10.15, если смотреть вдоль оси $OX$. Очевидно, что

$$\frac{y'}{z_p} = \frac{y}{z_p - z},$$

или же

$$y' = \frac{y}{1 - z/z_p}.$$

Аналогично получаем

$$x' = \frac{x}{1 - z/z_p};$$

явная форма для $P_2$ имеет вид

$$P_2(x, y, z) = \frac{1}{1 - z/z_p}(x, y, 0).$$

Предложение. *$P_2$ является нелинейной проекцией* $\mathbf{R}^3$.

Доказательство.

$$P_2\lambda(x, y, z) = P_2(\lambda x, \lambda y, \lambda z) = \frac{1}{1 - \lambda z/z_p}(\lambda x, \lambda y, 0),$$

$$\lambda P_2(x, y, z) = \frac{1}{1 - z/z_p}(\lambda x, \lambda y, 0) \neq P_2\lambda(x, y, z)$$

для произвольного $\lambda$. Следовательно, $P_2$ нелинейно. Имеем

$$P_2(P_2(x, y, z)) = P_2\left(\frac{x}{1 - z/z_p}, \frac{y}{1 - z/z_p}, 0\right) =$$

$$= \frac{1}{1 - 0}\left(\frac{x}{1 - z/z_p}, \frac{y}{1 - z/z_p}, 0\right) = P_2(x, y, z),$$

и, таким образом, $P_2$ — проекция в $\mathbf{R}^3$ по определению. //

Отсюда можно сделать заключение, что $P_2$ нельзя включить в $\mathcal{M}(3, \mathbf{R})$. В образе можно достигнуть хорошей глубины применением $P_2$, при условии что выбрана подходящая точка проекции. На рис. 10.17 показан вид прямоугольного параллелепипеда, полученный проекциями $P_1$ и $P_2$.

**2.3. Однородные координаты и линейное представление.** В этом разделе мы опишем технику представления преобразований $\mathbf{R}^2$ и $\mathbf{R}^3$, рассмотренных выше, при помощи элементов из $\mathcal{M}(3, \mathbf{R})$ и $\mathcal{M}(4, \mathbf{R})$ соответственно.

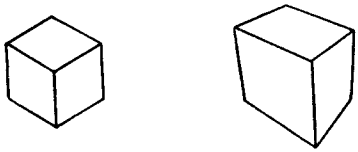


Рис. 10.17

Идея является общей и описывается в $\mathbf{R}^n$; мы ищем вложение некоторых специальных классов преобразований $\mathbf{R}^n$, не все из которых являются линейными, в алгебру матриц $\mathcal{M}(n+1; \mathbf{R})$.

В § 1 было показано, что $Q^n$ является биекцией $\mathbf{H}^n \to \mathbf{R}^n$ и может быть использовано для определения системы координат в $\mathbf{R}^n$. Это означает, что если $T$ — преобразование $\mathbf{R}^n$, то мы можем его выразить как преобразование $\mathbf{H}^n$. Другими словами, существует отображение $\tilde{T}$: $\mathbf{H}^n \to \mathbf{H}^n$ такое, что диаграмма на рис. 10.18 коммутативна. Имеем $Q^n \circ \tilde{T} = T \circ Q^n$, но отображение $Q^n$ имеет обратное. Поэтому мы можем записать $\tilde{T} = (Q^n)^{-1} \circ T \circ Q^n$. Если $S$ — другое преобразование $\mathbf{R}^n$, то

$$\widetilde{S \circ T} = (Q^n)^{-1} \circ S \circ T \circ Q^n =$$
$$= (Q^n)^{-1} \circ S \circ Q^n \circ (Q^n)^{-1} \circ T \circ Q^n = \tilde{S} \circ \tilde{T}.$$

Другими словами, мы можем соединить вместе диаграммы, не нарушая коммутативность, как это показано на рис. 10.19. Для некоторых преобразований $T$ пространства $\mathbf{R}^n$ существует линейное преобразование

$$T_L\colon \mathbf{R}^{n+1} \to \mathbf{R}^{n+1}$$

такое, что $\tilde{T}[\mathbf{x}] = [T_L\mathbf{x}]$ для всех $\mathbf{x} \in \mathbf{R}^{n+1}$.

$T_L$ существует для всех преобразований, рассмотренных в этой главе, и является требуемым представлением $T$ в $\mathbf{R}^{n+1}$, так как если $\mathbf{r} \in \mathbf{R}^n$, то мы имеем

$$T\mathbf{r} = Q^n \circ \tilde{T} \circ (Q^n)^{-1}\mathbf{r} = Q^n \circ \tilde{T}[\mathbf{r}, 1] = Q^n[T_L(\mathbf{r}, 1)].$$

Таким образом, преобразование $T$ получается путем применения $T_L$ к однородному представлению и последующим применением $Q^n$ для возврата к физическим координатам.

Чтобы иметь уверенность в том, что все работает правильно, надо показать, что композиция $S \circ T$ соответствует произведению матриц $S_L T_L$. Это легко сделать, так как

$$S \circ T[\mathbf{x}] = S[T_L \mathbf{x}] = [S_L T_L \mathbf{x}].$$

Графически это означает, что мы можем

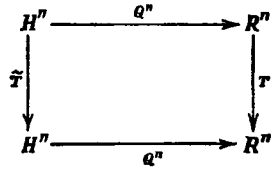



Рис. 10.18

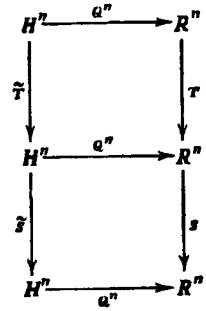



Рис. 10.19

расширить коммутативную диаграмму, изображенную на рис. 10.19, до коммутативной диаграммы, изображенной на рис. 10.20. Отсюда следует, что, один раз определив матрицы $T_L$, $S_L$, ..., соответствующие преобразованиям

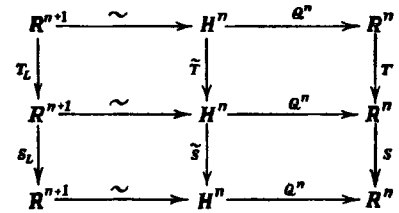



Рис. 10.20

$T$, $S$, ..., которые мы хотим осуществить, мы затем сохраняем их, опуская обозначения классов эквивалентности и работаем с векторами, представляющими эти классы. Произведения преобразований в $\mathbf{R}^n$ получают путем умножения матриц из $\mathcal{M}(n+1, \mathbf{R})$, примененных к векторам, представляющим классы. Все представители данного класса будут давать те же самые физические координаты в соответствии с отображением

$$(x_1, \ldots, x_n, x_{n+1}) \mapsto \frac{1}{x_{n+1}} (x_1, \ldots, x_n).$$

Обычно выбирают вектор $(\mathbf{r}, 1) \in \mathbf{R}^{n+1}$, чтобы представить $\mathbf{r} \in \mathbf{R}^n$. Такой выбор, очевидно, всегда возможен.

В следующих примерах мы получим матрицы $T_L$ для преобразований, описанных в пп. 2.1, 2.2. Все матрицы вычисляются в стандартном базисе.

Пример 2.1. Пусть $T\colon \mathbf{R}^n \to \mathbf{R}^n$ — линейное преобразование с матрицей $A_T$. Тогда матрица $T_L$ будет равна

$$\begin{bmatrix} & & 0 \\ & A_T & \vdots \\ & & 0 \\ 0 & \dots 0 & 1 \end{bmatrix},$$

так как если $\mathbf{x} = (\mathbf{r}, 1) \in \mathbf{R}^{n+1}$, то

$$\tilde{T}[\mathbf{x}] = (Q^n)^{-1} \circ T \circ Q^n[\mathbf{x}] = (Q^n)^{-1} \circ T\mathbf{r} =$$

$$= (Q^n)^{-1} A_T \mathbf{r} = [(A_T\mathbf{r}, 1)] = \left[\begin{bmatrix} & & 0 \\ & A_T & \vdots \\ & & 0 \\ 0 & \dots 0 & 1 \end{bmatrix}\begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix}\right].$$

Для $\begin{bmatrix} & & 0 \\ & A_T & \vdots \\ & & 0 \\ 0 & \dots 0 & 1 \end{bmatrix}$, если это удобно, используют обозначение $\begin{bmatrix} A_T & 0 \\ 0 & 1 \end{bmatrix}$. Раскладывая по последней строке, видим, что $\det \begin{bmatrix} A_T & 0 \\ 0 & 1 \end{bmatrix} = 0$ тогда и только тогда, когда $\det A_T = 0$; очевидно, что $\begin{bmatrix} A_T & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} A_T^{-1} & 0 \\ 0 & 1 \end{bmatrix}$. /

Пример 2.1 показывает, как применять линейные преобразования; например, применяя преобразования к $SO(2)$, получаем представление

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

в $\mathcal{M}(3, \mathbf{R})$ для матрицы

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Аналогично, если $W \in SO(3)$, то $W_L$ имеет вид

$$\begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}$$

в $\mathcal{M}(4, \mathbf{R})$ и

$$\begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} W^T & 0 \\ 0 & 1 \end{bmatrix}.$$

Пример 2.2. Пусть $T$: $\mathbf{R}^n \to \mathbf{R}^n$ — перенос

$$T\mathbf{r} = \mathbf{r} + \mathbf{a} \quad \text{для всех } \mathbf{r} \in \mathbf{R}^n,$$

где $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbf{R}^n$ — фиксированный вектор. Тогда матрица $T_L$ будет равна

$$\begin{bmatrix} & & a_1 \\ & I & \vdots \\ & & a_n \\ 0 & \ldots 0 & 1 \end{bmatrix};$$

она может быть записана в более краткой форме:

$$\begin{bmatrix} I & \mathbf{a} \\ 0 & 1 \end{bmatrix}.$$

Легко показать, что если $\mathbf{x} = (\mathbf{r}, 1)$, то

$$\widetilde{T}[\mathbf{x}] = (Q^n)^{-1} \circ T \circ Q^n[\mathbf{x}] = (Q^n)^{-1} \circ T\mathbf{r} =$$

$$= (Q^n)^{-1}(\mathbf{r} + \mathbf{a}) = [(\mathbf{r} + \mathbf{a}, 1)] = \left[\begin{bmatrix} I & \mathbf{a} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix}\right]. \; //$$

Пример 2.3. Аналогичными вычислениями можно показать, что преобразование $(W, \mathbf{a}) \in E(2)$ можно выполнить, используя

$$\begin{bmatrix} W & \mathbf{a} \\ 0 & 1 \end{bmatrix} \in GL(3, \mathbf{R}), \quad \text{где} \quad W \in SO(2) \text{ и } \mathbf{a} \in \mathbf{R}^2,$$

а преобразование $(W, \mathbf{a}) \in E(3)$ можно выполнить, используя

$$\begin{bmatrix} W & \mathbf{a} \\ 0 & 1 \end{bmatrix} \in GL(4, \mathbf{R}), \quad \text{где} \quad W \in SO(3) \text{ и } \mathbf{a} \in \mathbf{R}^3.$$

Обратные матрицы имеют вид

$$\begin{bmatrix} W & \mathbf{a} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} W^T & -W^T\mathbf{a} \\ 0 & 1 \end{bmatrix}$$

как в $E(2)$, так и в $E(3)$. //

Пример 2.4. Проекция $P_2$: $\mathbf{R}^3 \to \mathbf{R}^3$, определяемая соотношением

$$P_2(x, y, z) = \frac{1}{1 - z/z_p}(x, y, 0),$$

может быть выполнена в $\mathcal{M}(4, \mathbf{R})$ как

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/z_p & 1 \end{bmatrix},$$

так как если $\mathbf{x} = (\mathbf{r}, 1) \in R^4$ и $\mathbf{r} = (x, y, z)$, то

$$\tilde{P}_2[\mathbf{x}] = (Q^n)^{-1} \circ P_2 \circ Q^n[(\mathbf{r}, 1)] = (Q^n)^{-1} \circ P_2 \mathbf{r} =$$

$$= (Q^n)^{-1}\left(\frac{x}{1 - z/z_p}, \frac{y}{1 - z/z_p}, 0\right) = \left[\left(\frac{x}{1 - z/z_p}, \frac{y}{1 - z/z_p}, 0, 1\right)\right] =$$

$$= [(x, y, 0, 1 - z/z_p)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/z_p & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad //$$

Упражнение 10.1.

1. Показать, что группы $(T(2), \circ)$ и $(\mathrm{R}^2, +)$ изоморфны.

2. Доказать, что если

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

то матрица поворота $W(\theta) \in SO(2)$ может быть записана в экспоненциальной форме $W(\theta) = \exp\{\theta J\}$.

3. Определить евклидово преобразование, отображающее треугольник $PQR$, изображенный на рис. 10.21, *a*, в треугольник $P'Q'R'$, изображенный на рис. 10.21, *b*.

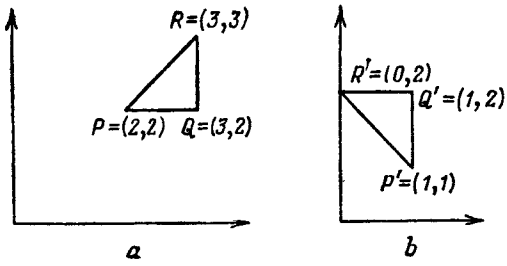



Рис. 10.21

4. Доказать, что $S(2)$ — коммутативная подгруппа линейных преобразований $\mathrm{R}^2$.

5. Показать, что в общем случае элементы $S(2)$ не коммутируют с элементами $SO(2)$, и построить рисунок, демонстрирующий это.

6. Доказать, что единица нетривиальной подгруппы $S(2)$ не коммутирует с $SO(2)$.

7. Используя преобразование $S(x, y) = (3x, 5y)$, изменить масштаб $\mathrm{R}^2$ так, чтобы точка (2, 2) оставалась фиксированной. Определить это преобразование.

8. Показать, что если $\hat{\mathbf{n}} \in \mathbf{R}^2$ — единичный вектор, то $\mathbf{n} \times ((\hat{\mathbf{n}} \times \mathbf{r}) \times \hat{\mathbf{n}}) = \hat{\mathbf{n}} \times \mathbf{r}$ для всех $\mathbf{r} \in \mathbf{R}^3$.

9. Пусть $\mathbf{n} \in \mathbf{R}^3$ и $A_{\mathbf{n}}$: $\mathbf{R}^3 \to \mathbf{R}^3$ определено соотношением $A_{\mathbf{n}}\mathbf{r} = \mathbf{n} \times \mathbf{r}$ для всех $\mathbf{r} \in \mathbf{R}^3$. Показать, что $A_{\mathbf{n}}$ — линейное преобразование $\mathbf{R}^3$, и определить матрицу $A_{\mathbf{n}}$ — в стандартном базисе $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$.

10. Используя результаты п. 9 этого упражнения и выражение $W_{\hat{\mathbf{n}}}(\theta)$ через $A_{\hat{\mathbf{n}}}$, получить в явном виде матричную форму поворота $W_{\hat{\mathbf{n}}}(\theta)$ в $\mathbf{R}^3$. Показать, что $\det W_{\hat{\mathbf{n}}}(\theta) = 1$.

11. В обозначениях п. 9 показать, что для всех $\mathbf{r} \in \mathbf{R}^3$ имеем

$$A_{\hat{\mathbf{n}}}^3\mathbf{r} = -A_{\hat{\mathbf{n}}}\mathbf{r}, \qquad A_{\hat{\mathbf{n}}}^4\mathbf{r} = -A_{\hat{\mathbf{n}}}^2\mathbf{r},$$

$$A_{\hat{\mathbf{n}}}^5\mathbf{r} = A_{\hat{\mathbf{n}}}\mathbf{r}, \qquad A_{\hat{\mathbf{n}}}^6\mathbf{r} = A_{\hat{\mathbf{n}}}^2\mathbf{r},$$

$$A_{\hat{\mathbf{n}}}^{6+k}\mathbf{r} = A_{\hat{\mathbf{n}}}^{2+k}\mathbf{r}$$

для всех $k \in \mathbf{N}$; использовать эти результаты для доказательства равенства

$$W_{\hat{\mathbf{n}}}(\theta) = \exp\left(\theta A_{\hat{\mathbf{n}}}\right).$$

12. а) Используя экспоненциальную форму $W_{\hat{\mathbf{n}}}(\theta)$, получить другое доказательство того, что

$$W_{\hat{\mathbf{n}}}(\theta)\left(W_{\hat{\mathbf{n}}}(\theta)\right)^T = 1.$$

б) Если $A \in \mathcal{M}(n, \mathbf{R})$, то можно показать, что

$$\det(\exp A) = \exp\left\{\sum_{i=0}^{n} A_{ii}\right\};$$

используя этот факт, получить другое доказательство соотношения

$$\det W_{\hat{\mathbf{n}}}(\theta) = 1.$$

13. Доказать, что $\hat{\mathbf{n}}$ является собственным вектором $W_{\hat{\mathbf{n}}}(\theta)$ при любом $\theta$, $0 \leqslant \theta < 2\pi$. Чему равно собственное значение? Дать геометрическую интерпретацию этому результату.

14. Вывести формулу для матриц поворота

$$W_{\hat{\mathbf{i}}}(\varphi), \qquad W_{\hat{\mathbf{j}}}(\chi).$$

15. Показать, что все подгруппы $S_{\hat{\mathbf{n}}} = \{W_{\hat{\mathbf{n}}}(\theta): 0 \leqslant \theta < 2\pi\}$ изоморфны $SO(2)$.

16. Пусть $\mathbf{a} \in \mathbf{R}^3$ — фиксированный вектор и преобразование $T$: $\mathbf{R}^3 \to \mathbf{R}^3$ определено как

$$T\mathbf{r} = \begin{cases} \dfrac{\mathbf{r}}{\mathbf{a}\cdot\mathbf{r}} & \text{при } \mathbf{a}\cdot\mathbf{r} \neq 0, \\ 0 & \text{в противном случае.} \end{cases}$$

а) Доказать, что $T$ — нелинейная проекция в $\mathbf{R}^3$.

б) Доказать, что

$$T[(\mathbf{r}, 1)] = [(\mathbf{r}, \mathbf{a}\cdot\mathbf{r})]$$

для всех $\mathbf{r} \in \mathbf{R}^3$ при $\mathbf{r}\cdot\mathbf{a} \neq 0$.

в) Определить матрицу $A \in \mathcal{M}(4, \mathbf{R})$ такую, что

$$T[\mathbf{x}] = [A\mathbf{x}] \text{ для всех } \mathbf{x} \in \mathbf{R}^4, \text{ где } [\mathbf{x}] \in \mathbf{H}^3.$$

## § 3. Кривые и поверхности

**3.1. Математическое представление.** Кривые и поверхности образуют основу большинства вспомогательных устройств компьютера и графического математического обеспечения. Возможны различные математические описания одной и той же геометрической формы; некоторые из них обсуждались и доступны с точки зрения приложений.

Если $I \subset \mathbf{R}$ — интервал, то через $\varkappa^n$ обозначим множество всех отображений класса $C^1$:

$$c\colon I \to \mathbf{R}^n,$$

причем $c' \neq 0$ на $I$. На $\varkappa^n$ определим отношение $\sim$ следующим образом. Пусть

$$c_1\colon I_1 \to \mathbf{R}^n, \quad c_2\colon I_2 \to \mathbf{R}^4;$$

тогда $c_1 \sim c_2$, если существует отображение $\varphi\colon I_1 \to I_2$ из класса $C^1$ такое, что $c_1 = c_2 \circ \varphi$ и $\varphi' \neq 0$ на $I_1$.

Предложение. *Отношение $\sim$ является отношением эквивалентности на $\varkappa^n$.* //

Доказательство оставляем в качестве упражнения.

Пусть $\mathscr{C}^n$ обозначает классы эквивалентности $\varkappa^n/\sim$.

Определение. *Кривой* в $\mathbf{R}^n$ называется элемент $\mathscr{C}^n$. //

Если $c \in \varkappa^n$ и $c\colon I \to \mathbf{R}^n$, то $I$ называют *параметрическим пространством* $c$, а $t$ — параметром или координатой $c$. График $c$ ($\operatorname{graph} c$) определяют как множество точек

$$\{c(t)\colon t \in I\}.$$

Если $[c]$ обозначает класс эквивалентности $c$ в $\mathscr{C}^n$, то для всех $c_1 \in [c]$ имеем

$$\operatorname{graph} c_1 = \operatorname{graph} c,$$

так что разумнее говорить о графике кривой $[c]$. Обратное неверно; мы можем иметь соотношение $\operatorname{graph}[c_1] = \operatorname{graph}[c_2]$, но при этом $[c_1] \neq [c_2]$. Отношение эквивалентности $\sim$ группирует вместе элементы $\varkappa^n$, которые параметризуются «аналогичным» образом.

Для вычислительных целей выберем элемент $c \in [c]$ и назовем кривую также $c$, хотя с точки зрения терминологии это неправильно. Это возможно при условии, что мы понимаем разницу и проявляем осторожность, когда это необходимо. График кривой в $\mathscr{C}^2$ является множеством точек, которые мы видим отображенными на графическом терминале. Тогда $\mathscr{C}^2$ — множество плоских кривых, а элементы $\mathscr{C}^3$ — множество пространственных кривых. $\mathscr{C}^2$ и $\mathscr{C}^3$ важны для приложений, однако, где легко провести рассуждения в более общем случае, будем это делать.

В литературе по компьютерной графике употребляют термины «параметрический», «явный», «неявный» по отношению к различным методам определения кривых. Если $(\xi_1, \ldots, \xi_n)$ — система координат в $\mathbf{R}^n$ и элемент $c \in \varkappa^n$ определен при помощи $n$ функций $t \mapsto \xi_i(t)$ $(1 \leqslant i \leqslant n)$, то такое задание кривой называют *явным параметрическим описанием*. Иногда выделяют два типа — «симметричный» и «несимметричный». Описание несимметрично, если параметр является одной из координат, т. е. $t = \xi_i$ для некоторого $i$, $1 \leqslant i \leqslant n$. Следовательно, несимметричные описания имеют вид

$$(\xi_i, (\xi_1(\xi_i), \xi_2(\xi_i), \ldots, \xi_i, \ldots, \xi_n(\xi_i))), \quad \xi_i \in I;$$

в случае $\mathscr{C}^2$ и декартовых координат эта запись имеет знакомый вид

$$(x, (x, y(x))), \quad x \in [x_0, x_1],$$

и кривую определяют, задавая явно $y$ как функцию от $x$. С другой стороны, мы можем описать кривую в $\mathbf{R}^n$, определяя подходящую функцию $f$: $\mathbf{R}^n \to \mathbf{R}$. Для фиксированного $a \in \mathbf{R}$ функция $f$ определяет кривую с графиком

$$f^{-1}(a) = \{(\xi_1, \ldots, \xi_n) : f(\xi_1, \ldots, \xi_n) = a\};$$

$f(\xi_1, \ldots, \xi_n)$ называют *уравнением* кривой (строго говоря, следовало бы назвать графиком кривой). В принципе

уравнению такого типа можно придать явную форму, хотя в общем случае это сделать достаточно сложно. Говорят, что кривые, определяемые такими уравнениями, описаны *неявно*.

Заметим, что не все отображения $f\colon \mathbf{R}^n \to \mathbf{R}$ будут задавать кривые описанным выше способом. Например, постоянное отображение задает все $\mathbf{R}^n$. Далее функция $f\colon \mathbf{R}^n \to \mathbf{R}$ может для одних значений постоянной $a$ задавать кривую, а для других — нет, как мы это увидим ниже.

Плоская кривая, заданная неявно уравнением вида

$$ax^2 + by^2 + cxy + dx + ey + f = 0,$$

называется *квадратичной* кривой (*кривой второго порядка*).

Пусть $c_1\colon [a, b] \to \mathbf{R}^n$ и $c_2\colon [c, d] \to \mathbf{R}^n$, где $a < b = c < d$ и $c_1(b) = c_2(b)$. Тогда $c_1 \vee c_2$ определяют следующим образом:

$$c_1 \vee c_2 = \begin{cases} c_1 & \text{на } [a, b], \\ c_2 & \text{на } [c, d] \end{cases}$$

и называют *объединением* функций $c_1$ и $c_2$. Для симметричных описаний, если $c \neq b$, $c_2$ может быть параметризовано таким образом, чтобы это условие выполнялось. Это означает, что мы можем построить $c_2^* \in [c_2]$ такое, что интервал для $c_2^*$ начинается с $b$. Следовательно, при условии $c_1(b) = c_2(b)$ объединение $c_1 \vee c_2$ имеет смысл, но может не быть, строго говоря, кривой. Проблема состоит в том, что $c_1 \vee c_2$ может не принадлежать $C^1$.

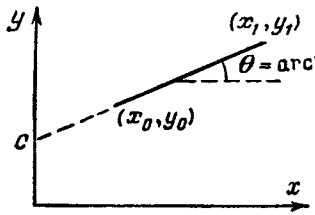


Рис. 10.22

Пример 3.1. Прямолинейная кривая (прямая) может быть определена в $\mathbf{R}^2$ обычным образом как

$$y(x) = mx + c,$$
$$x_0 \leqslant x \leqslant x_1,$$

где $m$ — наклон прямой, а $c$ — точка пересечения с осью $Oy$ (рис. 10.22). В наших обозначениях это будет

$$(x, (x, mx + c)), \quad x_0 \leqslant x \leqslant x_1;$$

$m$ и $c$ могут быть заданы непосредственно, однако более

естественно использовать концы $r_0=(x_0, y_0)$ и $r_1=(x_1, y_1)$, откуда получаем

$$m=(y_1-y_0)/(x_1-x_0), \quad c=y_1-mx_1.$$

Проблема, возникающая в несимметричных описаниях, теперь очевидна. Эта форма не описывает вертикальную линию (когда $x_0=x_1$). Если вместо этого воспользоваться неявной формой

$$(y-y_0)(x_1-x_0)-(y_1-y_0)(x-x_0)=0, \quad x_0\leqslant x\leqslant x_1,$$

то при $x_1=x_0$ получим уравнение вертикальной линии $x=x_0$. Неявное уравнение прямой линии в общем случае имеет вид

$$ax+by+c=0,$$

и вертикальные линии описываются этим уравнением при $b=0$.

Обычное симметричное описание может быть записано в векторной форме:

$$(t, \mathbf{r}_0+t\widehat{\mathbf{u}}), \quad t\in[0, \|\mathbf{r}_1-\mathbf{r}_0\|],$$

где $\widehat{\mathbf{u}}$ — единичный вектор, $\widehat{\mathbf{u}}=(\mathbf{r}_1-\mathbf{r}_0)/\|\mathbf{r}_1-\mathbf{r}_0\|$. //

Рассмотренные выше примеры приводят к общим наблюдениям о природе кривых, определенных несимметричным способом. Перед тем как обсуждать это, дадим несколько необходимых определений.

Определение. Говорят, что плоская кривая $(u, (x(u), y(u)))$, $u\in I=[u_A, u_B]$ является:

— *однозначной*, если для всех $u_1, u_2\in I$ имеем

$$x(u_1)=x(u_2)\Rightarrow y(u_1)=y(u_2);$$

— *многозначной*, если предыдущее условие не выполнено;

— *замкнутой*, если $(x(u_A), y(u_A))=(x(u_B), y(u_B))$. //

На рис. 10.23 проиллюстрированы все три случая. Очевидно, что замкнутая кривая является многозначной. Несимметричное описание $(x, (x, y(x)))$, $x_0\leqslant x\leqslant x_1$ может определять только *однозначную* кривую. Это потому, что $y$ является функцией от $x$ и, следовательно,

$$x_1=x_2\Rightarrow y(x_1)=y(x_2).$$

Вертикальная линия является многозначной кривой.

Многие приложения в компьютерной графике требуют, чтобы замкнутые и многозначные кривые были вклю-

чены в описания. Единственный путь достижения этого — объединять кривые, как это описывалось ранее. При использовании несимметричных форм это не очень удобно

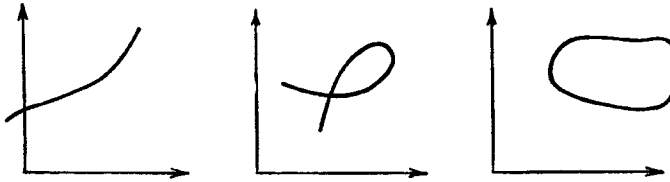



Рис. 10.23

и редко применяется на практике. Симметричные описания не имеют этого ограничения и, следовательно, являются более удобными для таких приложений. Несимметричные формы обычно используют, когда требуется однозначная кривая.

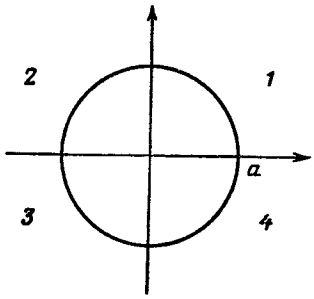



Рис. 10.24

Пример 3.2. Окружность является примером замкнутой кривой. Рассмотрим окружность в $\mathbf{R}^2$ радиуса $a$ с центром в начале координат (рис. 10.24).

Чтобы записать окружность в несимметричной форме, необходимы две кривые: $y_1(x) = (a^2 - x^2)^{1/2}$, $-a \leqslant x \leqslant a$ в квадрантах 1 и 2 и $y_2(x) = -(a^2 - x^2)^{1/2}$, $-a \leqslant x \leqslant a$ в квадрантах 3 и 4. Уравнение окружности в неявной форме имеет вид

$$x^2 + y^2 - a^2 = 0$$

или же в векторных обозначениях,

$$\|\mathbf{r}\| - a = 0.$$

Это уравнение описывает всю окружность; оно может быть записано в симметричной параметрической форме

$$(\theta,\ a(\cos\theta,\ \sin\theta)), \quad \theta \in [0,\ 2\pi[,$$

$$\left(t,\ a\left(\frac{1-t^2}{1+t^2},\ \frac{2t}{1+t^2}\right)\right), \qquad t \in \mathbf{R}.$$

Возможно бесконечное число других симметричных форм,

выбор которых зависит от приложений. Для выполнения рисунков более предпочтительной является параметризация с относительно постоянным изменением $\mathbf{r}(t)$. Математически это означает, что величина $\|\mathbf{r}'(t)\|$ приблизительно постоянна на $I$. Несимметричное представление в большинстве случаев является неудобным, так как требуется проверка, на какой ветви кривой мы сейчас находимся. //

Примерами кривых второго порядка являются эллипс, гипербола и парабола.

Если $G$ — группа преобразований $\mathbf{R}^n$, тогда $G$ преобразует $\mathscr{C}^n$ естественным образом. Если $c \in \mathscr{C}^n$ — кривая

$$c = (t, \mathbf{r}(t)), \quad t \in I,$$

и $g \in G$, то определим кривую $gc$ следующим образом:

$$gc = (t, g\mathbf{r}(t)), \quad t \in I.$$

Например, когда $G = SO(2)$ и $c \in \mathscr{C}^2$, то график $W(\theta)c$ является графиком $c$, повернутым на угол $\theta$ (рис. 10.25).

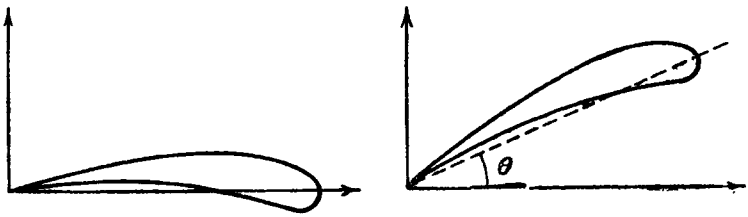



*График c*
*a*

*График W(θ)c*
*b*

Рис. 10.25

Уравнения обычно описывают кривые в «стандартном» положении, когда ось $OX$ является осью симметрии. Чтобы получить уравнение геометрически эквивалентной кривой в некотором другом положении и ориентации в пространстве, надо просто применить подходящий элемент группы $E(2)$ к уравнению. В качестве примера выведем уравнение эллипса с графиком $\mathbf{e}$, изображенным на рис. 10.26. В стандартном положении (изображенном штриховой кривой на рисунке) кривая записывается в виде

$$c = (\varphi, (a\cos\varphi, b\sin\varphi)), \quad \varphi \in [0, 2\pi[;$$

следовательно, кривая с графиком $\varepsilon$ имеет вид

$$(W(\theta), \mathbf{d})c = (\varphi, (W(\theta), \mathbf{d})(a\cos\varphi, b\sin\varphi)) =$$
$$= (\varphi, (a\cos\varphi\cos\theta - b\sin\varphi\sin\theta + d_1, a\cos\varphi\sin\theta + b\sin\varphi\cos\theta + d_2)), \varphi \in [0, 2\pi[.$$

Неявные уравнения для кривых в нестандартном положении могут быть получены аналогично.

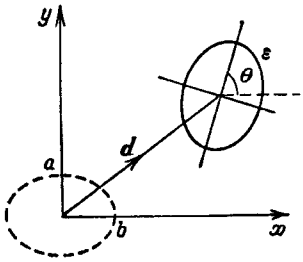


Рис. 10.26

Перед тем как перейти к другой теме, упомянем один интересный факт, проверка которого осуществляется довольно легко, когда кривые заданы неявными уравнениями. Пусть $f(x, y) = 0$ — уравнение плоской кривой, которая делит плоскость $\mathbf{R}^2$ на три части, тогда, если $(x', y')$ — произвольная точка пространства, то знак $f(x', y')$ определяет область, в которой лежит $(x', y')$. Например, возьмем уравнение окружности

$$f(x, y) = x^2 + y^2 - a^2 = 0;$$

тогда

$f(x', y') > 0 \Rightarrow (x', y')$ лежит за пределами круга, $f(x', y') = 0 \Rightarrow (x', y')$ лежит на окружности, $f(x', y') < 0 \Rightarrow (x', y')$ лежит внутри круга.

Проверки такого типа используют в трехмерных случаях в алгоритмах удаления невидимых линий с изображений. Исследование поверхностей может быть проведено по аналогии с исследованием кривых. Методы представления имеют те же самые преимущества и недостатки.

Поверхности двумерны и имеют пространство параметров вида $I_1 \times I_2$, где $I_1, I_2 \subseteq \mathbf{R}$ — интервалы. Параметрическое представление в общем случае имеет вид

$$((u, v), r(u, v)), \quad (u, v) \in I_1 \times I_2.$$

Неявное уравнение поверхности записывается в виде $f(x, y, z) = 0$, а поверхностью второго порядка является поверхность, определяемая уравнением

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0,$$

где $a, b, c, d, e, f, g, h, i, j \in \mathbf{R}$.

Пример 3.3. Пусть $D = \{\mathbf{r}_i\colon \mathbf{r}_i \in \mathbf{R}^3,\ 0 \leqslant t \leqslant 2\}$ — множество линейно независимых векторов. Тогда

$$\mathbf{r}(u, v) = \mathbf{r}_0 + u(\mathbf{r}_1 - \mathbf{r}_0) + v(\mathbf{r}_2 - \mathbf{r}_0),\ (u, v) \in \mathbf{R}^2,$$

есть симметричное описание плоскости $P$, проходящей через точки $D$ (рис. 10.27). Если $\mathbf{q} = (\mathbf{r}_2 - \mathbf{r}_0) \times (\mathbf{r}_1 - \mathbf{r}_0)$,

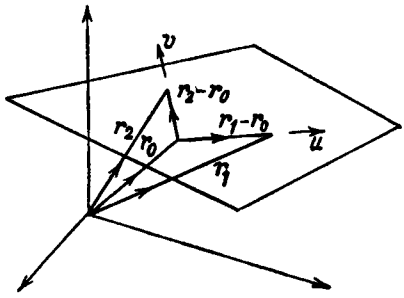



Рис. 10.27

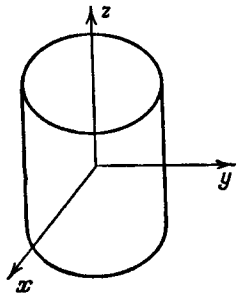



Рис. 10.28

тогда $\hat{\mathbf{n}} = \mathbf{q}/\|\mathbf{q}\|$ — единичный вектор, ортогональный $P$ (называемый *нормалью*). Таким образом, $\mathbf{r} \in P$ тогда и только тогда, когда $(\mathbf{r} - \mathbf{r}_0) \cdot \hat{\mathbf{n}} = 0$, или

$$(x - x_0)n_1 + (y - y_0)n_2 + (z - z_0)n_3 = 0.$$

Это является неявным представлением $P$. //

Пример 3.4. Пусть $S$ — сфера в $\mathbf{R}^3$ радиуса $a$ с центром в начале координат. Очевидно, что $\mathbf{r} \in S$ тогда и только тогда, когда $\|\mathbf{r}\| = a$, или же

$$x^2 + y^2 + z^2 - a^2 = 0.$$

Сферические координаты дают симметричное представление

$$((\theta, \varphi),\ a(\sin\theta\cos\varphi,\ \sin\theta\sin\varphi,\ \cos\theta)),$$
$$(\theta, \varphi) \in [0, \pi] \times ]0, 2\pi[.\ //$$

Пример 3.5. Пусть $C$ — цилиндр радиуса $a$ с осью симметрии $OZ$. В этом случае $(x, y, z) \in C$ тогда и только тогда, когда

$$x^2 + y^2 - a^2 = 0.$$

Используя цилиндрические координаты, получаем симметричную форму

$$((\varphi, z),\ (a\cos\varphi,\ a\sin\varphi,\ z)).$$

Обычно рассматривают «конечный» цилиндр (рис. 10.28). Для него уравнение будет иметь вид

$$x^2 + y^2 - a^2 = 0, \quad z_0 \leqslant z \leqslant z_1. \ //$$

Относительно уравнений поверхностей, находящихся в нестандартном положении, можно сделать те же замечания, что и для случая кривых; надо лишь $E(2)$ заменить на $E(3)$.

**3.2. Геометрия плоских кривых.** Целью данного раздела является определение понятий длины, касательной и кривизны плоских кривых. На рис. 10.29 изображена плоская кривая

$$c = (t, \mathbf{r}(t)), \ t \in [t_A, t_B].$$

$P$ — произвольная точка на $c$, являющаяся концом вектора $\mathbf{r}(t)$, а $P'$ — конец вектора $\mathbf{r}(t + \delta t)$, $\delta t \in \mathrm{R} \setminus \{0\}$.

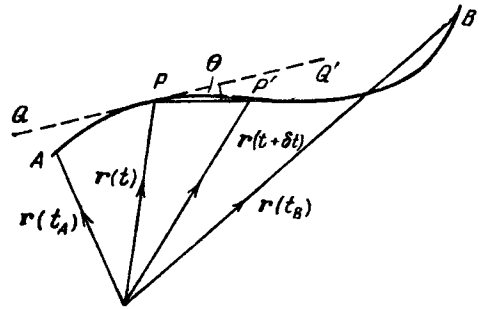



Рис. 10.29

Пусть $\delta\mathbf{r}(t)$ обозначает вектор $\overrightarrow{PP'}$. Тогда, очевидно,

$$\delta\mathbf{r}(t) = \mathbf{r}(t + \delta t) - \mathbf{r}(t),$$

$$\| \delta\mathbf{r}(t) \| = \left\| \frac{\mathbf{r}(t + \delta t) - \mathbf{r}(t)}{\delta t} \right\| \delta t$$

есть длина отрезка $PP'$.

Если разбить интервал $[t_A, t_B]$ на много маленьких интервалов равной длины $[t_{i-1}, t_i]$, $t_A = t_0 < t_1 < \ldots < t_n = t_B$, то можно образовать сумму

$$S_{AB} = \sum_{i=1}^{n} \| \delta\mathbf{r}_i \|,$$

где $\delta\mathbf{r}_i = \mathbf{r}(t_i) - \mathbf{r}(t_{i-1})$, которая равна длине ломаной ли-

нии вдоль $c$. Интуитивно ясно, что мы получаем аппроксимацию кривой $S_{AB}$ между $\mathbf{r}(t_A)$ и $\mathbf{r}(t_B)$. Приближение будет тем лучше, чем больше число разбиений отрезка $[t_A, t_B]$. Эти интуитивные понятия приводят нас к определению *длины кривой* $S_{AB}$:

$$S_{AB} = \lim_{n\to\infty} \sum_{i=1}^{n} \| \delta \mathbf{r}_i \|.$$

С вычислительной точки зрения эта формула трудна для использования. Поэтому мы перепишем ее в более удобном виде:

$$\lim_{n\to\infty} \sum_{i=1}^{n} \| \delta \mathbf{r}_i \| = \lim_{n\to\infty} \sum_{i=1}^{n} \| \mathbf{r}(t_i) - \mathbf{r}(t_{i-1}) \| =$$

$$= \lim_{n\to\infty} \sum_{i=1}^{n} \left\| \frac{\mathbf{r}(t_{i-1} + \delta t) - \mathbf{r}(t_{i-1})}{\delta t} \right\| \delta t,$$

где $\delta t = (t_B - t_A)/n$. Но это по определению интеграл

$$\int_{t_A}^{t_B} \left\| \frac{d\mathbf{r}}{dt} \right\| dt;$$

следовательно,

$$S_{AB} = \int_{t_A}^{t_B} \left\| \frac{d\mathbf{r}}{dt} \right\| dt.$$

Если $\mathbf{r}(t) = (x(t), y(t))$, то формула принимает вид

$$S_{AB} = \int_{t_A}^{t_B} \left[ \left( \frac{dx}{dt} \right)^2 + \left( \frac{dy}{dt} \right)^2 \right]^{1/2} dt.$$

Интуитивно ясно, что линия наклона (на рис. 10.29 это $QQ'$) к кривой $c$ является прямой, касающейся этой кривой и имеющей тот же самый наклон, что и $c$, в точке $P$. Пусть $\theta$ — это угол между отрезком $PP'$ и касательной $QQ'$. Тогда при $\delta t \to 0$ следует ожидать, что $\theta \to 0$. Это означает, что направление $\delta \mathbf{r}(t)$ или же $\delta \mathbf{r}(t)/\delta t$ стремится к направлению $QQ'$ при $\delta t \to 0$. Другими словами, вектор

$$\frac{d\mathbf{r}}{dt} = \lim_{\delta t \to 0} \frac{\delta \mathbf{r}}{\delta t}$$

параллелен $QQ'$. Если

$$s = \int_{t_A}^{t} \left\| \frac{d\mathbf{r}}{dt} \right\| dt,$$

тогда в принципе все кривые могут быть выражены в терминах параметра *длины дуги* $(s, \mathbf{r}(s))$, $s \in [0, S_{AB}]$. Из определения длины дуги получаем

$$s = \int_{0}^{s} \left\| \frac{d\mathbf{r}}{ds'} \right\| ds';$$

дифференцируя обе части по $s$, имеем

$$1 = \left\| \frac{d\mathbf{r}}{ds} \right\|.$$

Другими словами, $d\mathbf{r}/ds$ — единичный вектор, параллельный касательной линии к $\mathbf{r}(s)$. Это вызывает следующее определение.

Определение. Пусть $(s, \mathbf{r}(s))$, $s \in [0, S_{AB}]$,— кривая с длиной дуги в качестве параметра. Определим *единичный вектор касательной* $\widehat{\mathbf{T}}(s)$ к кривой в точке $s$ как

$$\widehat{\mathbf{T}}(s) = \frac{d\mathbf{r}}{ds}(s),$$

а *нормальный вектор* $\widehat{\mathbf{N}}(s)$ к кривой как

$$\widehat{\mathbf{N}}(s) = W(\pi/2)\,\widehat{T}(s). \quad //$$

Базис $(\widehat{T}, \widehat{N})$ образует правостороннюю систему координат в $\mathbf{R}^2$. Если кривая не параметризуется в терминах длины дуги, то выражение для $\widehat{\mathbf{T}}$ (см. упражнение 10.2) будет иметь вид

$$\widehat{\mathbf{T}}(u) = \begin{cases} \dfrac{d\mathbf{r}}{du} \Big/ \left\| \dfrac{d\mathbf{r}}{du} \right\|, & \text{если } \dfrac{ds}{du} > 0, \\ -\dfrac{d\mathbf{r}}{du} \Big/ \left\| \dfrac{d\mathbf{r}}{du} \right\|, & \text{если } \dfrac{ds}{du} < 0. \end{cases}$$

Эти формулы неявно предполагают условия на $s$. В действительности знак $\widehat{\mathbf{T}}$ не образует класса эквивалентности, инвариантного на $\mathscr{C}^2$, а является функцией выбранного параметрического представления. Даже если параметризация осуществлена с помощью параметра — длины дуги, она зависит от того, с какого конца мы на-

чинаем измерять $s$. Касательное пространство к точке кривой является классом, инвариантным на $\mathscr{C}^2$, где касательное пространство к $[c]$ в точке $P$ имеет вид $\{\alpha\widehat{\mathbf{T}}\colon \alpha \in \mathbf{R}$, где $\widehat{\mathbf{T}}$ — касательный вектор в $P$ для некоторого $c \in [c]\}$.

Из упражнений к § 4 гл. 5 следует, что $d\widehat{\mathbf{T}}/ds$ ортогонален к $\widehat{\mathbf{T}}$; следовательно, существует функция $\varkappa$: $[0, S_{AB}] \to \mathbf{R}$ такая, что $\frac{d\widehat{\mathbf{T}}(s)}{ds} = \varkappa(s)\widehat{N}(s)$; $\varkappa(s)$ называют *кривизной* кривой $\mathbf{r}(s)$, а $1/\varkappa(s)$ — *радиусом кривизны* $\mathbf{r}(s)$.

Упражнение 10.2.

1. Получить в $\mathbf{R}^2$ явное уравнение прямой, проходящей через точки $(-1, 3)$ и $(2, -1)$. Определить единичный вектор, параллельный этой прямой, и написать уравнение в параметрической векторной форме.

2. Определить касательный и нормальный векторы $\widehat{\mathbf{T}}$ и $\widehat{\mathbf{N}}$ и кривизну $\varkappa$ следующих плоских кривых:

а) окружности — $(\theta,\ a(\cos\theta\,\hat{i} + \sin\theta\,\hat{j}))$, $0 \leqslant \theta < 2\pi$;

б) эллипса — $(\varphi,\ (a\cos\varphi\,\hat{i} + b\sin\varphi\,\hat{j}))$, $0 \leqslant \varphi \leqslant 2\pi$;

в) параболы — $(t,\ (at^2\hat{i} + 2at\hat{j}))$, $-\infty < t < \infty$, где $a > 0$ и $b > 0$ действительные.

3. Плоская кривая определена формулой

$$(u,\ (a\cos u\hat{i} + b(1 - e^{-u/2})\hat{j})),\quad 0 \leqslant u < \infty,$$

где $a > 0$ и $b > 0$ действительные.

а) Показать, что касательный вектор к кривой задается формулой $\widehat{\mathbf{T}}(u) = \dfrac{-2a\sin u\,\hat{i} + be^{-u/2}\hat{j}}{(4a^2\sin^2 u + b^2e^{-u})^{1/2}}$.

б) Начертить кривую в интервале $0 \leqslant u \leqslant 2\pi$.

в) Найти нормаль к кривой.

4. Начертить кривую $(u,\ (a\cos u,\ b\sin u,\ bu))$, $-\infty < u < \infty$, и найти выражение для касательного вектора $\widehat{\mathbf{T}}(u)$.

5. Пусть $(x,\ (x,\ y(x)))$, $x_0 \leqslant x \leqslant x_1$, описывает плоскую кривую. Показать, что касательный вектор $\widehat{\mathbf{T}}(x)$ может быть записан как

$$\widehat{\mathbf{T}}(x) = \frac{(1,\ dy/dx)}{(1 + (dy/dx)^2)^{1/2}},$$

и, следовательно, кривизна $\varkappa(x)$ в точке $x$ имеет вид

$$\varkappa(x) = \frac{d^2y/dx^2}{(1 + (dy/dx)^2)^{3/2}}.$$

6. Получить формулы, аналогичные полученным в п. 5, для симметричного представления $(u, (x(u), y(u)))$, $u_0 \leqslant u \leqslant u_1$.

7. Найти симметричное параметрическое уравнение плоскости $P$, проходящей через точки (0, 1, 0), (3, —2, 0) и (1, 3, 4) в $\mathbf{R}^3$. Показать, что нормальный вектор будет параллелен (—4, —4, 3), и использовать этот факт для вывода неявного уравнения для $P$.

8. *Поверхностью вращения* называется поверхность, являющаяся результатом вращения плоской кривой вокруг некоторой фиксированной оси в $\mathbf{R}^3$.

а) Показать, что если плоская кривая

$$(u, (p(u)\hat{\mathbf{i}} + q(u)\hat{\mathbf{k}})), \quad u_A \leqslant u \leqslant u_B$$

делает поворот на угол $2\pi$ вокруг оси $OZ$, то соответствующая поверхность вращения описывается уравнением

$$((u, \varphi), (p(u)\cos\varphi\hat{\mathbf{i}} + p(u)\sin\varphi\hat{\mathbf{j}} + q(u)\hat{\mathbf{k}})),$$

где $u_A \leqslant u \leqslant u_B$, $0 \leqslant \varphi \leqslant 2\pi$.

б) Использовать результаты задачи 8, а) для получения симметричных параметрических представлений следующих поверхностей:

— цилиндра; — конуса; — тора.

9. Цилиндр $C_1$ длины $2l$ определяется уравнением $x^2 + z^2 = b^2$, $-l \leqslant y \leqslant l$, и пересекается с цилиндром $C_2$ длины $h$, который определяется уравнением $y^2 + (z - \alpha)^2 = a^2$, $0 \leqslant x \leqslant h$, где $0 < \alpha < b$, $h > b$.

Используя параметрические координаты

$$\{(x, \theta): 0 \leqslant x \leqslant h, 0 \leqslant \theta < 2\pi\} \text{ на } C_2,$$

показать, что кривая, являющаяся пересечением $C_1$ и $C_2$, может быть записана как $(\theta, \mathbf{r}(\theta))$, $0 \leqslant \theta < 2\pi$, где

$$\mathbf{r}(\theta) = ((b^2 - (\alpha + a\sin\theta)^2)^{1/2}, a\cos\theta, \alpha + a\sin\theta).$$

Показать также, что единичный касательный вектор к кривой в точке $\theta = 0$ задается формулой

$$\frac{1}{ab}(-a\alpha, 0, a(b^2 - \alpha^2)^{1/2}).$$

# ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Научное издание

*КУК Д., БЕЙЗ Г.*

**КОМПЬЮТЕРНАЯ МАТЕМАТИКА**

Зуведующий редакцией *Е. Ю. Ходан*
Редакторы: *Е. В. Ильченко, А. Б. Угольников*
Художественный редактор *Т. Н. Кольченко*
Технический редактор *С. Я. Шкляр*
Корректоры: *М. А. Смирнов, И. Я. Кришталь*

**ИБ № 32537**

Сдано в набор 02.03.89. Подписано к печати 20.04.90. Формат 84×108/32. Бумага книжно-журнальная. Гарнитура обыкновенная. Печать высокая. Усл. печ. л. 20,16. Усл. кр.-отт. 20,16. Уч.-изд. л. 19,84. Тираж 23 000 экз. Заказ № 604. Цена 1 р. 70 к.

Ордена Трудового Красного Знамени издательство «Наука»
Главная редакция физико-математической литературы
117071 Москва В-71, Ленинский проспект, 15

4-я типография издательства «Наука»
630077 Новосибирск, 77, Станиславского, 25