МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Нижегородский государственный университет им. Н.И. Лобачевского

В.Е. Алексеев

Дискретная математика

Учебное пособие

Рекомендовано методической комиссией ИИТММ
для студентов ННГУ, обучающихся по направлениям подготовки
010302 Прикладная математика и информатика, 010301 Математика,
020301 Математика и компьютерные науки

Нижний Новгород
2017

УДК 519.1
ББК 22.176

Алексеев В.Е. ДИСКРЕТНАЯ МАТЕМАТИКА: Учебное пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017. –139 с.

Рецензенты: д. ф.-м. н., доц. Д.Т. Чекмарев
к. ф.-м. н., доц. В.А. Зорин

В учебном пособии излагаются основные понятия и фундаментальные факты важнейших разделов дискретной математики: комбинаторики, теории графов, теории логических функций, теории кодирования.

Учебное пособие предназначено для студентов ННГУ, обучающихся по направлениям подготовки 010302 Прикладная математика и информатика, 010301 Математика, 020301 Математика и компьютерные науки

Ответственный за выпуск:
зам. председателя методической комиссии ИИТММ ННГУ,
к.т.н, доцент **В.М. Сморкалова**

УДК 519.1
ББК 22.176

Оглавление

Глава 1. Множества

1.1. Понятие множества

Под *множеством* математики понимают соединение каких-либо объектов в одно целое. Создатель теории множеств немецкий математик Георг Кантор (1845-1918) определил множество как «объединение в одно целое объектов, хорошо различаемых нашей интуицией или нашей мыслью». Он же сформулировал это короче: «множество – это многое, мыслимое нами как единое». На самом деле ни одна из этих фраз не является определением в строгом математическом понимании. Понятие множества вообще не определяется, это одно из первичных понятий математики. Его можно пояснить, приводя более или менее близкие по смыслу слова: коллекция, класс, совокупность, ансамбль, собрание, или примеры: экипаж корабля – множество людей, стая – множество птиц, созвездие – множество звезд. Множества, рассматриваемые в математике, состоят из математических объектов (чисел, функций, точек, линий и т.д.). Объекты, из которых состоит множество, называют его *элементами*. Важно отметить, что в множестве все элементы отличаются друг от друга, одинаковых элементов быть не может.

Тот факт, что элемент x принадлежит множеству A, обозначают так: $x \in A$, а если x не принадлежит A, то пишут $x \notin A$.

Множества бывают *конечные* и *бесконечные*. Конечное множество может быть задано перечислением его элементов, при этом список элементов заключается в фигурные скобки, например:

$\{1, 2, 4, 8, 16\}$;

$\{a, b, c, d\}$;

$\{$красный, желтый, зеленый$\}$.

Элементы могут перечисляться в любом порядке: $\{a, b, c, d\}$ и $\{c, b, d, a\}$ – одно и то же множество.

Число элементов в конечном множестве называется его *мощностью***.** Мощность множества X обозначается $|X|$.

Иногда и бесконечные множества задаются в форме перечисления элементов с использованием многоточия, например:

$\{1,2,3, \dots\}$;

$\{1,3,5, \dots\}$;

$\{1,4,9, \dots\}$.

При этом предполагается, что читающий подобную запись знает, как должен быть продолжен написанный ряд (или его следует предупредить об этом).

Примеры бесконечных множеств:

$\mathbb{N} = \{1,2,3, \dots\}$ – множество всех натуральных чисел;

$\mathbb{N}_{\mathbf{0}} = \{0,1,2,\dots\}$ – множество натуральных чисел с добавленным элементом 0;

$\mathbb{Z} = \{0,1,-1,2,-2,\dots\}$ – множество всех целых чисел;

$\mathbb{Q}$ – множество всех рациональных чисел;

$\mathbb{R}$ – множество всех вещественных чисел.

Пустое множество обозначается знаком $\emptyset$, оно не содержит ни одного элемента: $|\emptyset| = 0$. Иногда полезно считать, что существует некое *универсальное множество* (*универс*, *универсум*), содержащие все элементы, представляющие интерес в данных обстоятельствах. Например, изучая свойства целых чисел, мы можем выбрать в качестве универса множество $\mathbb{Z}$, а занимаясь геометрией на плоскости – множество всех точек плоскости. Обычно универс обозначают буквой U .

Часто множество задают указанием свойства P , выделяющего элементы этого множества среди всех элементов универса U. Тот факт, что элемент x имеет свойство P, записывают так: $P(x)$. Множество всех элементов из U, имеющих свойство P, представляется в форме: $\{x \in U: P(x)\}$ или $\{x: x \in U \text{ и } P(x)\}$ или просто $\{x: P(x)\}$, если ясно, о каком универсе идет речь. Примеры:

$\{x \in \mathbb{N}: x \text{ четно}\}$;

$\{x: x \in \mathbb{Z} \text{ и } x > 0\} = \mathbb{N}$;

$\{x: x^2 - 2 = 0\} = \{\sqrt{2}, -\sqrt{2}\}$.

1.2. Подмножества

Множество A называется *подмножеством* множества B, если каждый элемент из A принадлежит B. Символически это записывается так: $A \subseteq B$. Это можно прочитать как "A включено в B". Отметим некоторые свойства отношения включения:

$\emptyset \subseteq A$ для любого множества A.

$A \subseteq A$ для любого множества A.

Если $A \subseteq B$ и $B \subseteq A$, то $A = B$.

Если $A \subseteq B$ и $B \subseteq C$, то $A \subseteq C$.

Элемент множества сам может быть множеством. Например, множество $X = \{\{a,b\},\{a\},\emptyset,\{b,c\},\{a,b,c\}\}$ состоит из 5 элементов.

Если элементами множества X являются подмножества множества A, то говорят, что X есть *семейство подмножеств* множества A. Приведенное выше множество X есть семейство подмножеств множества $A = \{a,b,c\}$.

Семейство всех подмножеств множества A обозначается через 2^A . Если, например, $A = \{a,b\}$, то $2^A = \{\emptyset,\{a\},\{b\},\{a,b\}\}$.

Теорема 1.1 (о числе подмножеств). *Если A – конечное множество, то* $|2^A| = 2^{|A|}$.

Доказательство. Пусть $|A| = n$. Доказательство проводим индукцией по n. При $n = 0$ утверждение верно, так как $2^0 = 1$, а единственным подмножеством пустого множества является оно само. При $n > 0$ возьмем какой-нибудь элемент $x \in A$ и обозначим через B множество всех элементов множества A, отличных от x. Тогда $|B| = n - 1$ и по предположению индукции $|2^B| = 2^{n-1}$. Каждое подмножество множества A либо содержит, либо не содержит элемент x. Подмножества, не содержащие x, являются подмножествами множества B, таких имеется 2^{n-1}. Всякое подмножество, содержащее x, получается из некоторого подмножества множества B добавлением к нему элемента x. Поэтому таких подмножеств тоже 2^{n-1}. Всего, следовательно, $|2^A| = 2^{n-1} + 2^{n-1} = 2^n$. ■

Для представления подмножеств конечного множества часто используют следующий способ. Пусть U – конечное множество, элементы которого пронумерованы числами 1, 2, …, n: $U = \{u_1, u_2, \dots, u_n\}$. Подмножество $X \subseteq U$ можно задать последовательностью нулей и единиц:

$$h(X) = (h_1, h_2, \dots, h_n), \text{ где } h_i = \begin{cases} 1, & \text{если } u_i \in X, \\ 0, & \text{если } u_i \notin X. \end{cases}$$

Последовательность $h(X)$ называют *характеристическим вектором* множества X. Пусть, например, $U = \{a, b, c, d, e\}$ и элементы нумеруются в алфавитном порядке. Тогда

$$h(\{a, c, d\}) = (1,0,1,1,0), \quad h(\{d, e\}) = (0,0,0,1,1), \quad h(\emptyset) = (0,0,0,0,0).$$

1.3. Алгебра множеств

Рассмотрим некоторые операции над множествами.

Объединение множеств A и B определяется как множество

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

Пример: $A = \{0,1,3\}$, $B = \{1,2,3\}$, $A \cup B = \{0,1,2,3\}$.

Пересечение множеств A и B определяется как множество

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

Пример: $A = \{0,1,3\}$, $B = \{1,2,3\}$, $A \cap B = \{1,3\}$.
Если $A \cap B = \emptyset$, то говорят, что множества A и B *не пересекаются.*

Отметим некоторые свойства операций объединения и пересечения.

1. Для любого множества A выполняются равенства

$A \cup A = A,$ $A \cup \emptyset = A,$ $A \cup U = U,$

$A \cap A = A,$ $A \cap \emptyset = \emptyset,$ $A \cap U = A.$

2. *Коммутативность*: для любых множеств A и B выполняются равенства

$A \cup B = B \cup A$,

$A \cap B = B \cap A.$

3. *Ассоциативность*: для любых множеств A , B и C выполняются равенства

$A \cup (B \cup C) = (A \cup B) \cup C,$

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

4. *Дистрибутивность*: для любых множеств A , B и C выполняются равенства

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

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

Благодаря свойству ассоциативности можно записывать объединение и пересечение любого числа множеств, не пользуясь скобками для указания порядка действий. В этих случаях применяется сокращенная запись:

$$A_1 \cup A_2 \cup \ldots \cup A_n = \bigcup_{i=1}^{n} A_i,$$

$$A_1 \cap A_2 \cap \ldots \cap A_n = \bigcap_{i=1}^{n} A_i.$$

Разность множеств A и B определяется как множество

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

Пример: $A = \{0,1,3\}$, $B = \{1,2,3\}$, $A - B = \{0\}$, $B - A = \{2\}$.

Дополнение множества A относительно универса U определяется как множество

$$\bar{A} = U - A.$$

Симметрическая разность множеств A и B определяется как множество

$$A \otimes B = (A - B) \cup (B - A).$$

Отметим еще некоторые тождества, справедливые для введенных операций.

5. $A - B = A \cap \bar{B}$.

6. $A \cap \bar{A} = \emptyset,\ A \cup \bar{A} = U$.

7. $\bar{\bar{A}} = A$.

8. *Законы де Моргана*:

$\overline{A \cup B} = \bar{A} \cap \bar{B}$,

$\overline{A \cap B} = \bar{A} \cup \bar{B}$.

Каждое из приведенных тождеств доказывается непосредственно на основании определений операций: чтобы доказать, что два множества равны, нужно показать, что всякий элемент одного из них принадлежит другому и наоборот. В качестве примера приведем доказательство одного из законов де Моргана:

$$x \in \overline{A \cup B} \Leftrightarrow x \notin A \cup B \Leftrightarrow x \notin A \text{ и } x \notin B \Leftrightarrow x \in \bar{A} \text{ и } x \in \bar{B} \Leftrightarrow$$

$$\Leftrightarrow x \in \bar{A} \cap \bar{B}.$$

С помощью операций можно выразить отношения между множествами:

$A \subseteq B \ \Leftrightarrow\ A \cup B = B$,

$A \subseteq B \ \Leftrightarrow\ A \cap B = A$,

$A \subseteq B \ \Leftrightarrow\ A - B = \emptyset$,

$A = B \Leftrightarrow A \otimes B = \emptyset$.

С целью рационализации записи формул иногда принимают некоторые дополнительные соглашения. Можно, например, договориться, что операция пересечения имеет более высокий приоритет, чем другие. Это означает, что в отсутствие скобок, указывающих порядок действий, первой выполняется операция пересечения. Иногда знак пересечения вовсе опускают: пишут AB вместо $A \cap B$. Если принять эти соглашения, то, например, формула $A \cup (B \cap C)$ запишется как $A \cup BC$.

1.4. Диаграмма Венна

Диаграмма Венна – способ графического представления отношений между множествами и иллюстрации операций над множествами. Множества изображаются в виде кругов или других фигур, а универс, если он есть – в виде прямоугольника, охватывающего другие фигуры. На рисунке 1.1 изображены диаграммы Венна двух множеств с разными типами взаимоотношений между ними. На рисунке 1.2 показаны результаты различных операций над множествами. На рисунке 1.3 представлена диаграмма Венна для трех множеств. На ней представлены всевозможные пересечения этих множеств и их дополнений.

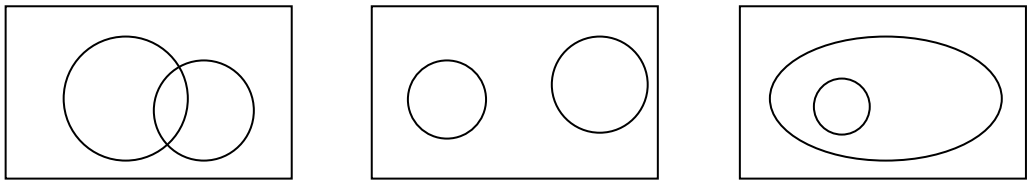


Рис. 1.1. Диаграммы Венна для двух множеств: слева – пересекающихся, в центре – непересекающихся, справа – одно включено в другое.

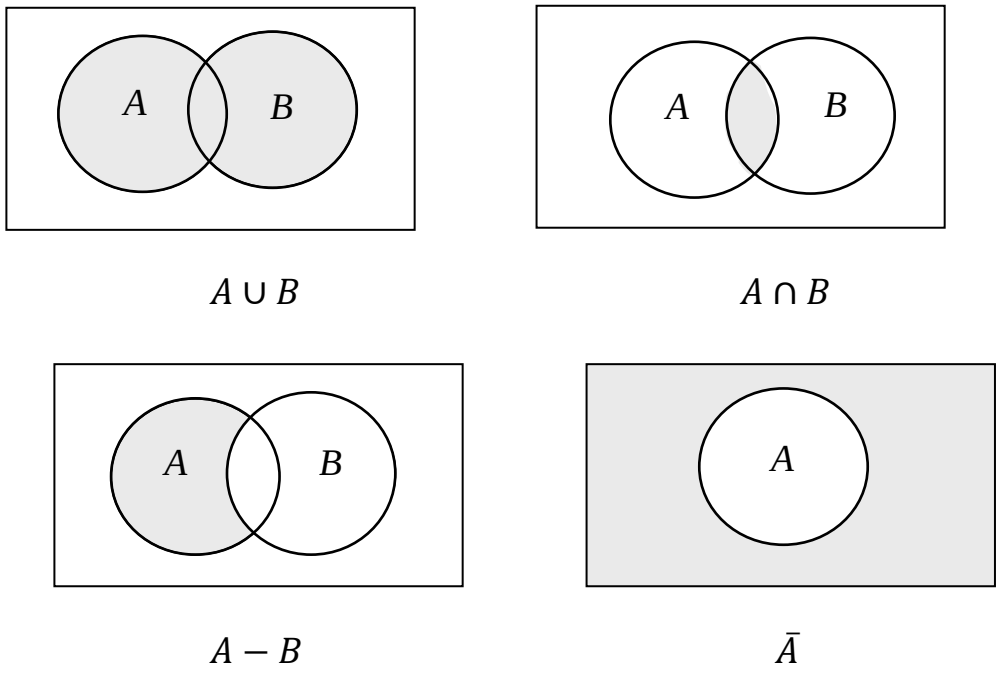


Рис. 1.2. Диаграммы Венна, иллюстрирующие операции над множествами, результат операции выделен цветом

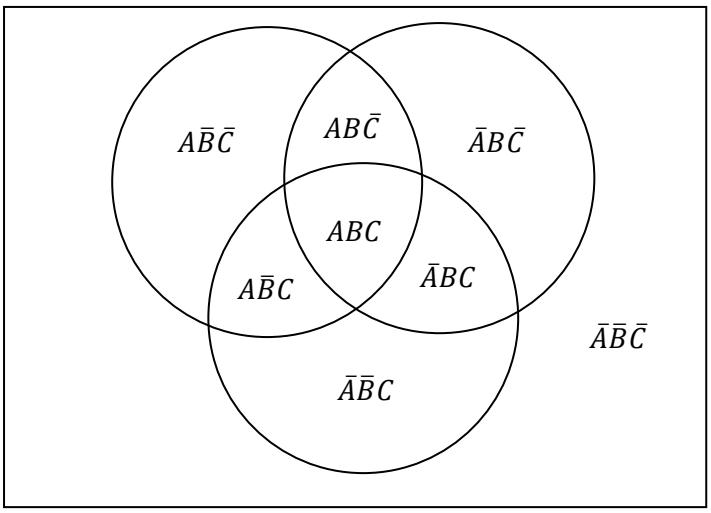


Рис. 1.3. Диаграмма Венна для трех множеств

1.5. Декартово произведение

Пусть A и B – два множества. Их *декартово произведение* (называемое также *прямым произведением*) определяется как множество всех пар (x, y), в которых $x \in A$, $y \in B$:

$$A \times B = \{(x, y): x \in A, y \in B\}.$$

Отметим, что пары здесь имеются в виду упорядоченные: (x, y) и (y, x) – это разные пары (если $x \neq y$).

Множество $A \times A$ называется *декартовым квадратом* множества A и обозначается через A^2.

Примеры:

Если $A = \{a, b, c\}$, а $B = \{1,2\}$, то

$A \times B = \{(a, 1), (a, 2), (b, 1), (b, 2), (c, 1), (c, 2)\}$,

$B^2 = \{(1,1), (1,2), (2,1), (2,2)\}$.

Множество дат типа «3 марта» можно рассматривать как декартово произведение множеств $A = \{1,2, \dots ,31\}$ и $B = \{$январь, … , декабрь$\}$.

Множество всех вещественных чисел, заключенных между 0 и 1, геометрически представляется отрезком $[0,1]$ координатной прямой. Множество $[0,1]^2$ состоит из пар чисел, которым соответствуют точки плоскости, заполняющие квадрат.

В общем случае декартово произведение n множеств $A_1, A_2, \dots, A_n$ определяется как множество, состоящее из последовательностей вида

$(x_1, x_2, \dots, x_n)$, в которых первый элемент принадлежит множеству A_1, второй – множеству A_2, и т.д.:

$$A_1 \times A_2 \times \dots \times A_n = \{(x_1, x_2, \dots, x_n):\ x_1 \in A_1, x_2 \in A_2, \dots, x_n \in A_n)\}.$$

Произведение n одинаковых множеств $A_1 = A_2 = \dots = A_n = A$ называется *n-ой декартовой степенью* множества A и обозначается через A^n. Например, множество $[0,1]^3$ геометрически представляет собой множество точек куба в трехмерном пространстве.

1.6. Мультимножества

Мультимножество – это совокупность элементов, в которую каждый элемент может входить больше одного раза. Как и в множестве, порядок элементов в мультимножестве не важен.

Примеры:

$\{a, a, b, c, c, c\}$ – мультимножество, состоящее из элементов множества $\{a, b, c\}$;

$\{c, a, c, b, c, a\}$ – то же самое мультимножество;

$\{a, b, b, b, c, c\}$ – другое мультимножество.

Глава 2. Отношения

2.1. Понятие отношения, общие свойства отношений

Отношением на множестве A называется любое подмножество множества A^2. Если R – отношение и $(x, y) \in R$, то говорят, что элемент x находится в отношении R с элементом y. Тот факт, что элемент x находится в отношении R с элементом y записывают иногда так: xRy.

Заметим, что элементы множества A^2 – это упорядоченные пары элементов множества A, поэтому из того, что x находится в отношении R с y не следует, что y находится в отношении R с x.

Примеры.

1. Неравенство $<$ является отношением на множестве $\mathbb{N}$ (а также на $\mathbb{Z}$ и на $\mathbb{R}$). Число 2 находится в отношении $<$ с числом 5, но 5 не находится в этом отношении с 2.

2. Равенство $=$ и неравенство $\neq$ являются отношениями на любом множестве A. Каждый элемент находится в отношении $=$ с самим собой и в отношении $\neq$ со всеми остальными.

3. Для любого множества A включение $\subseteq$ является отношением на множестве 2^A.

4. Отношение делимости на множестве $\mathbb{Z}$ определяется следующим образом: x *делит* y (или y *делится* на x), если существует такое целое k, что $xk = y$. Это отношение обозначается так: $x|y$.

5. Пусть A – множество всех прямых на плоскости. Можно определить отношение параллельности $\parallel$ на A : $L_1 \parallel L_2$ означает, что прямая L_1 параллельна прямой L_2.

Если множество A конечно, то и любое отношение на нем конечно и может быть задано перечислением пар элементов, находящихся в этом отношении. В качестве примера рассмотрим отношение делимости на множестве $A = \{1,2,3,4,5,6\}$. Это отношение можно задать как множество пар $\{(1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,2), (2,4), (2,6), (3,3), (3,6), (4,4), (5,5), (6,6)\}$.

Отношение на конечном множестве можно также представить в форме таблицы. Пусть R – отношение на множестве $A = \{a_1, \dots, a_n\}$. Построим таблицу с n строками и n столбцами, в которой на пересечении строки с номером i и столбца с номером j поставим 1, если $(a_i, a_j) \in R$, и 0 в противном случае. Для приведенного выше примера отношения делимости таблица получается такая:

	1	2	3	4	5	6
1	1	1	1	1	1	1
2	0	1	0	1	0	1
3	0	0	1	0	0	1
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1

Наглядное представление отношения дает *граф отношения*. Это диаграмма, которая строится следующим образом. Пусть R – отношение на множестве A. Элементы множества A изобразим кружками (или любыми другими значками), эти кружки называют *вершинами* графа. Если xRy, то рисуем стрелку от x к y. На рисунке 2.1 показан граф рассмотренного выше отношения делимости.

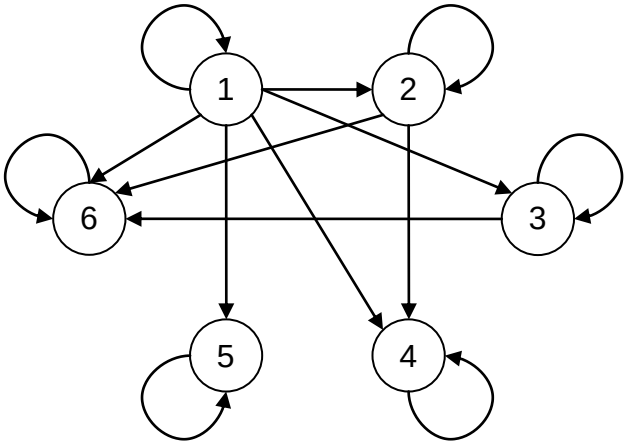


Рис. 2.1. Граф отношения делимости

Так как отношение есть множество (пар), то любые операции над множествами можно применять к отношениям. Если R_1 и R_2 – отношения на множестве A, то $R_1 \cup R_2$, $R_1 \cap R_2$ и т.д. – тоже отношения на A. Если R – отношение на A, то $\bar{R}$ – дополнение R до множества A^2.

Обратное отношение R^{-1} к отношению R определяется следующим образом:

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

На рисунке 2.2 показаны графы двух отношений на множестве $A = \{a, b, c\}$ и результаты различных операций над ними.

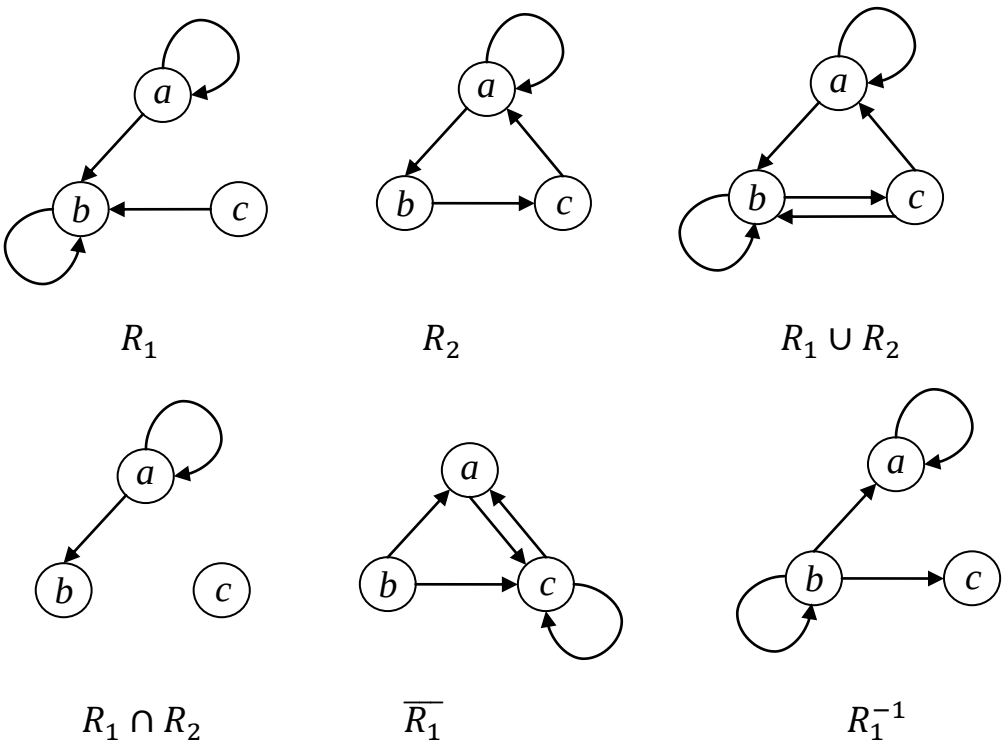


Рис. 2.2. Графическое представление двух отношений и операций над ними

Отношение R на множестве A называется

рефлексивным, если для любого $x \in A$ имеет место xRx;

симметричным, если для любых $x, y \in A$ из xRy следует yRx;

антисимметричным, если для любых $x, y \in A$ из xRy и yRx следует $x = y$;

транзитивным, если для любых $x, y, z \in A$ из xRy и yRz следует xRz.

2.2. Отношения эквивалентности

Всякое рефлексивное, симметричное и транзитивное отношение называется *отношением эквивалентности*.

Примеры.

1. Отношение равенства на любом множестве есть отношение эквивалентности.
2. Отношение параллельности прямых на плоскости – отношение эквивалентности.
3. Пусть $n \in \mathbb{N}$. Рассмотрим следующее отношение R на множестве $\mathbb{Z}$: xRy тогда и только тогда, когда разность $x - y$ делится на n. Очевидно, это

отношение рефлексивно и симметрично. Убедимся, что оно транзитивно. Пусть xRy и yRz. Это означает, что существуют такие p и q, что $x - y = pn$, $y - z = qn$. Складывая эти равенства, получаем $x - z = (p + q)n$, отсюда следует, что xRz. Итак, R – отношение эквивалентности. Если xRy, то говорят, что x *и* y *сравнимы по модулю* n и пишут

$$x \equiv y \pmod{n}.$$

Семейство непустых подмножеств множества A называется *разбиением*, если всякий элемент множества A принадлежит точно одному подмножеству из этого семейства. Эти подмножества называются *частями разбиения*.

Пусть, например, $A = \{0,1,2,3,4,5,6,7,8,9\}$. Тогда

$$F = \{\{0,5\},\{1,8,9],\{2\},\{3,4,6,7\}\}$$

– разбиение множества A на четыре части.

Пусть F – разбиение множества A. Определим отношение R на A следующим образом:

$xRy \Leftrightarrow x$ и y принадлежат одной части разбиения.

Очевидно, R – отношение эквивалентности. Оказывается, все отношения эквивалентности устроены подобным образом.

Теорема 2.1 (о факторизации). *Для любого отношения эквивалентности R на множестве A существует такое разбиение F этого множества, что два элемента множества находятся в отношении R тогда и только тогда, когда они принадлежат одной части разбиения.*

Доказательство. Определим для каждого $x \in A$ множество $R(x)$ всех элементов, находящихся в отношении R с x: $R(x) = \{y: xRy\}$. Рассмотрим семейство F, состоящее из всех таких множеств: $F = \{R(x): x \in A\}$ (отметим, что некоторые из этих множеств могут совпадать, то есть может быть $R(x) = R(y)$ при $x \neq y$, но в F каждое множество входит не более одного раза).

Покажем, что F является разбиением множества A. В самом деле, так как отношение R рефлексивно, то $x \in R(x)$ для любого $x \in A$. Следовательно, каждый элемент принадлежит хотя бы одному множеству из F. Допустим, x принадлежит, кроме $R(x)$, еще какому-нибудь множеству $R(y)$, т.е. имеет место отношение yRx. Покажем, что $R(y) = R(x)$. Пусть $z \in R(x)$. Тогда xRz. Так как отношение R транзитивно то из yRx и xRz следует, что yRz, то есть $z \in R(y)$. Следовательно, $R(x) \subseteq R(y)$. Обратно, если $z \in R(y)$, то yRz, отсюда по транзитивности zRx, значит, $z \in R(x)$. Следовательно, $R(y) \subseteq R(x)$.

Итак, F – разбиение. Покажем, что оно обладает свойством, о котором говорится в утверждении теоремы. Пусть элементы x и y находятся в

отношении R: xRy. Тогда $y \in R(x)$. Но, как мы видели, $x \in R(x)$, значит, оба элемента принадлежат одной части разбиения F. Обратно, пусть элементы x и y принадлежат одной части разбиения. Так как $x \in R(x)$, то эта часть есть $R(x)$. Значит, $y \in R(x)$, следовательно, xRy. ■

Части разбиения, о котором говорится в этой теореме, называются *классами эквивалентности*.

Примеры.

1. Для отношения равенства каждый класс эквивалентности состоит из одного элемента.
2. Для отношения параллельности прямых на плоскости каждый класс эквивалентности состоит из всех прямых одного направления.
3. Для отношения сравнимости по модулю n каждый класс эквивалентности состоит из чисел, имеющих одинаковый остаток при делении на n. Таким образом, всего будет n классов.

Разбиение на классы эквивалентности называют *факторизацией* множества A по отношению R, отсюда и название теоремы.

2.3. Отношения порядка

Всякое рефлексивное, антисимметричное и транзитивное отношение называется *отношением порядка*.

Примеры.

1. Отношение равенства на любом множестве есть отношение порядка.
2. Отношение $\leq$ на множестве $\mathbb{R}$ является отношением порядка.
3. Отношение $<$ на множестве $\mathbb{R}$ не является отношением порядка, так как не рефлексивно.
4. Отношение делимости на множестве $\mathbb{N}$ – отношение порядка.

Если на множестве A задано отношение порядка R, то пару (A, R) называют *упорядоченным множеством*. Примеры упорядоченных множеств:
$(\mathbb{R}, \leq)$,
$(2^U, \subseteq)$,
$(\mathbb{N}, |)$.

Пусть (A, R) – упорядоченное множество и $x, y \in A$. Говорят, что x и y *сравнимы*, если xRy или yRx, в противном случае они *несравнимы*. Если в (A, R) любые два элемента сравнимы, то порядок R называется *линейным*, а (A, R) – *линейно упорядоченным множеством.* Если же имеются несравнимые элементы, то говорят, что R – *частичный порядок*, а (A, R) –

частично упорядоченное множество. Например, $(\mathbb{R}, \leq)$ – линейно упорядоченное, а $(2^U, \subseteq)$ – частично упорядоченное множество.

Пусть (A, R) – упорядоченное множество. Если xRy и $x \neq y$, то говорят, что элемент x *предшествует* элементу y (или, что x *меньше* y). Если при этом не существует такого элемента z, отличного от x и y, что xRz и zRy, то говорят, что x *непосредственно предшествует* y. Таким образом, для всякого отношения порядка R определяется отношение непосредственного предшествования, которое будем обозначать R^*.

Примеры.

1. В упорядоченном множестве $(\mathbb{Z}, \leq)$ каждому элементу x непосредственно предшествует элемент $x - 1$.
2. В упорядоченном множестве $(2^U, \subseteq)$ для множества $A \subseteq U$ непосредственно предшествующими являются все множества, получающиеся из A удалением одного элемента.
3. В упорядоченном множестве $(\mathbb{R}, \leq)$ ни у одного элемента нет непосредственно предшествующих, то есть в этом случае $R^* = \emptyset$.

Пусть R – отношение порядка на множестве A и xRy. Последовательность элементов $z_1, z_2, \dots, z_n$ множества A, в которой $z_1 = x$, $z_n = y$, и $z_k R z_{k+1}$ для $k = 1,2, \dots, n-1$, назовем *цепочкой между x и y*.

Теорема 2.2 (о конечных упорядоченных множествах). *Пусть (A, R) – конечное упорядоченное множество, x и y – различные элементы множества A и xRy. Существует цепочка $z_1, z_2, \dots, z_n$ между x и y, в которой $z_k R^* z_{k+1}$ для каждого $k = 1,2, \dots, n-1$.*

Доказательство. Среди всех цепочек между x и y выберем цепочку наибольшей длины, пусть это $z_1, z_2, \dots, z_n$. Она обладает требуемым свойством – каждый ее элемент, кроме последнего, непосредственно предшествует следующему элементу. Действительно, если неверно, что $z_k R^* z_{k+1}$, то существует промежуточный элемент u: $z_k R u$ и $u R z_{k+1}$. Но тогда получаем более длинную цепочку $z_1, z_2, \dots, z_k, u, z_{k+1}, \dots, z_n$ между x и y. ■

Граф отношения R^* называется *диаграммой Хассе* отношения R. Из доказанной теоремы следует, что диаграмма Хассе дает полное описание отношения R: элемент x предшествует элементу y тогда и только тогда, когда из x можно попасть в y, двигаясь вдоль стрелок диаграммы. Обычно вершины на диаграмме располагают так, чтобы меньший элемент находился ниже большего. Тогда отношение между элементами можно изображать линией, а не стрелкой. На рисунке 2.3 показана диаграмма Хассе частично упорядоченного множества $(2^{\{a,b,c\}}, \subseteq)$.

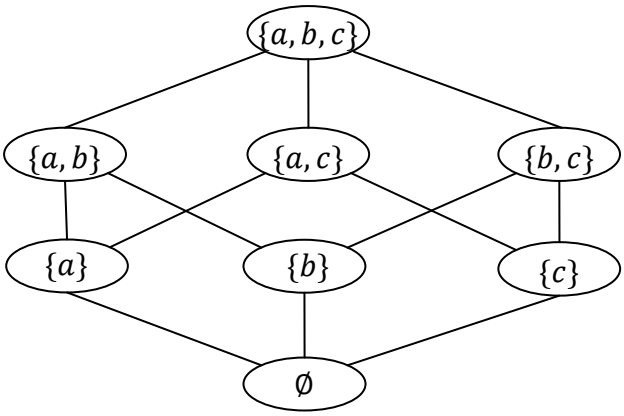


Рис.2.3. Диаграмма Хассе упорядоченного множества $(2^{\{a,b,c\}}, \subseteq)$

Элемент x называется *максимальным элементом* упорядоченного множества (A, R), если не существует такого $y \in A$, что $y \neq x$ и xRy, то есть не существует большего элемента. *Минимальный элемент* – тот, для которого не существует меньшего.

В упорядоченном множестве $(\mathbb{N}, \leq)$ имеется один минимальный элемент и ни одного максимального, а в упорядоченном множестве $(\mathbb{Z}, \leq)$ нет ни максимальных, ни минимальных. В любом конечном упорядоченном множестве имеются и минимальные и максимальные элементы.

Элемент x называется *наибольшим элементом* упорядоченного множества (A, R), если для каждого $y \in A$ выполняется yRx, и *наименьшим*, если для каждого $y \in A$ выполняется xRy.

На рисунке 2.4 показана диаграмма Хассе упорядоченного множества $(\{1,2,3,4,5,6\}, |)$. Видно, что в этом множестве имеется один минимальный элемент, он же наименьший, и три максимальных элемента, а наибольшего элемента нет.

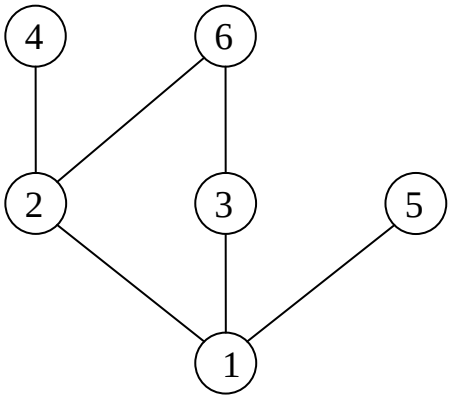


Рис.2.4. Диаграмма Хассе отношения делимости

2.4. Более общие виды отношений

Пусть A и B – множества. Любое подмножество множества $A \times B$ называется *отношением между* A и B. Например, отношение принадлежности $\in$ есть отношение между элементами и множествами, то есть между U и 2^U.

Еще более общим является понятие отношения между несколькими множествами. Пусть $A_1, A_2, \dots, A_n$ – множества. Любое подмножество множества $A_1 \times A_2 \times \dots \times A_n$ называется *n-местным* или *n-арным* отношением. При $n = 2$ это *бинарное* отношение, при $n = 3$ – *тернарное* и т. д. Многоместные отношения лежат в основе реляционных баз данных.

2.5. Функциональные отношения и функции

Пусть A и B – множества. Отношение f между A и B называют *функциональным отношением,* если для каждого $x \in A$ имеется ровно один $y \in B$, находящийся в отношении f с x. В этом случае пишут $y = f(x)$ и говорят, что задана *функция* f, определения на множестве A (*область определения* функции) и принимающая значения в множестве B. Функцию называют также *отображением* множества A в множество B. Символически это записывается так: $f: A \to B$ (читается: f отображает A в B). Если $y = f(x)$, то говорят, что элемент y есть *образ* элемента x при отображении f, а элемент x – *прообраз* элемента y. Если $X \subseteq A$, то множество всех образов элементов из X обозначается через $f(X)$.

Если для функции $f: A \to B$ при $x_1 \neq x_2$ всегда $f(x_1) \neq f(x_2)$, то говорят, что эта функция является *взаимно однозначным отображением* множества A в множество B. Взаимно однозначное отображение называют также *инъективным отображением* или коротко *инъекцией.*

Если $f: A \to B$ *и* $f(A) = B$, то говорят, что функция f *отображает* множество A *на* множество B. В этом случае функцию f называют также *сюръективным отображением* или *сюръекцией.*

Взаимно однозначное отображение множества A на множество B, то есть отображение, являющееся одновременно инъективным и сюръективным, называют *биективным отображением* или *биекцией*.

Для конечных множеств A и B справедливы следующие очевидные утверждения.

Если существует инъекция $f: A \to B$, *то* $|A| \leq |B|$.

Если существует сюръекция $f: A \to B$, *то* $|A| \geq |B|$.

Если существует биекция $f: A \to B$, *то* $|A| = |B|$.

Если существует биекция $f: A \to B$, то говорят, что имеется *взаимно однозначное соответствие* между множествами A и B. В этом случае существует *обратная функция* $f^{-1}: B \to A$ такая, что $f^{-1}\big(f(x)\big) = x$ для всех $x \in A$.

На рисунке 2.5 графически представлены функциональное и не функциональное отношения, а на рисунке 2.6 – функции разных типов.

Рис.2.5. Функциональное (слева) и не функциональное (справа) отношения

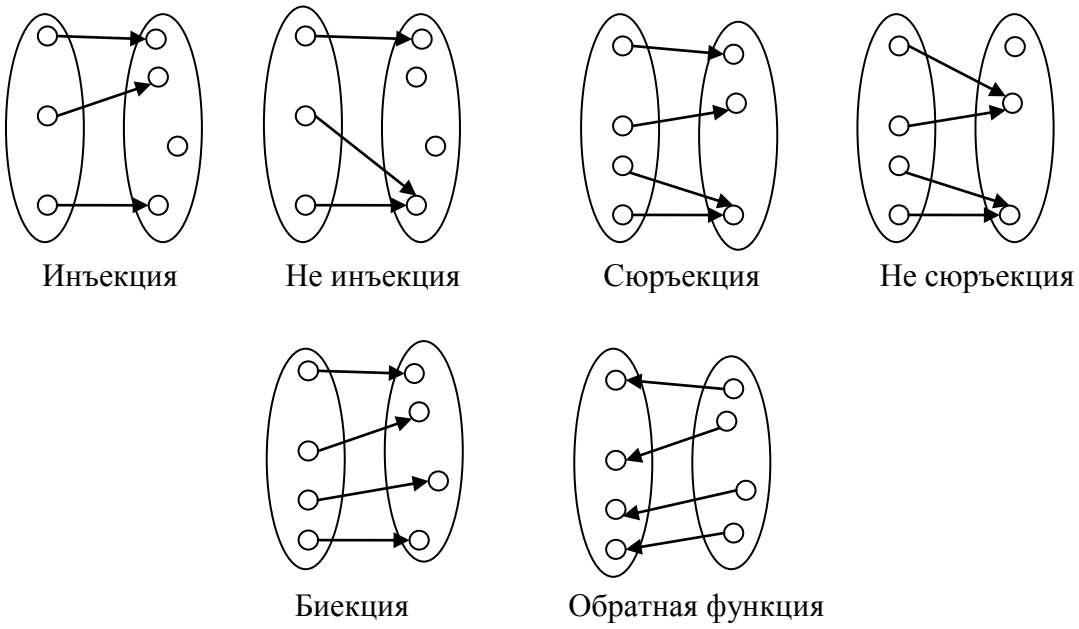


Рис. 2.6. Разные типы функций

2.6. Сравнение бесконечных множеств. Счетные и несчетные множества

Понятия инъекции и биекции дают возможность количественно сравнивать бесконечные множества. Множества A и B называют *равномощными,* если существует биекция $f: A \to B$. В этом случае пишут

$|A| = |B|$. Отношение равномощности рефлексивно, так как отображение $f(x) = x$ на любом множестве является биекцией этого множества на себя. Оно симметрично, так как если $f: A \to B$ – биекция, то имеется и биекция $f^{-1}: B \to A$. Оно транзитивно: если есть биекции $f: A \to B$ и $g: B \to C$, то функция h, определяемая равенством $h(x) = g(f(x))$, является биекцией множества A на множество C. Следовательно, равномощность является отношением эквивалентности на множестве 2^U для любого универса U.

Если существует инъекция $f: A \to B$, то говорят, что мощность множества A не больше мощности множества B: $|\mathrm{A}| \leq |\mathrm{B}|$. Если $|A| \leq |B|$ и множества A и B не равномощны, то есть существует инъекция, но не существует биекции из A в B, то мощность множества A меньше мощности множества B : $|A| < |B|$. Отношение $\leq$ между мощностями множеств обладает следующими свойствами:

- Если $|\mathrm{A}| \leq |\mathrm{B}|$ и $|B| \leq |C|$, то $|A| \leq |C|$.
- Если $|\mathrm{A}| \leq |\mathrm{B}|$ и $|\mathrm{B}| \leq |\mathrm{A}|$, то $|\mathrm{A}| = |\mathrm{B}|$.

Первое доказывается так же, как транзитивность отношения равномощности. Доказательство второго более сложное и здесь не приводится (оно известно как теорема Шрёдера-Бернштейна).

На рисунке 2.7 показано, что следующие множества точек равномощны: два отрезка любой длины, отрезок и дуга, дуга и прямая линия.

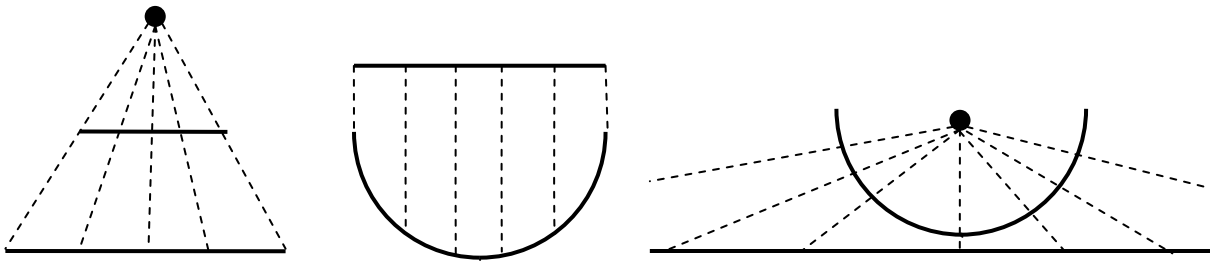


Рис.2.7. Равномощность отрезков разной длины (слева), отрезка и дуги (в центре), дуги и прямой (справа). Пунктирные линии показывают, как устанавливается взаимно однозначное соответствие

Множество называется *счетным*, если оно равномощно множеству натуральных чисел $\mathbb{N}$. Если множество A счетно, то существует биекция $f: \mathbb{N} \to A$ и элементы множества A можно расположить в последовательность: $A = \{f(1), f(2), f(3) \ldots\}$. Обратно, если элементы множества A образуют бесконечную последовательность: $A = \{a_1, a_2, a_3, \ldots\}$, то отображение f, задаваемое равенством $f(a_i) = i$, есть биекция множества A на $\mathbb{N}$. Следовательно, счетность множества равносильна возможности расположить его элементы в бесконечную последовательность

Примеры.

1. Множество $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ счетно. Действительно, имеется биекция множества $\mathbb{N}_0$ на множество $\mathbb{N}$ – функция $f(x) = x + 1$.

2. Множество $\mathbb{Z}$ всех целых чисел счетно. Действительно, целые числа можно расположить в последовательность: $0,1,-1,2,-2,3,-3,\ldots$

3. Множество $\mathbb{N}^2$ счетно. В самом деле, элементами этого множества являются пары (i,j), где i и j – натуральные числа. Пар, у которых $i+j=k$, имеется ровно $k-1$: это пары $(1,k-1)$, $(2,k-2)$,…,$k-1,1)$. Учитывая это, можно расположить все пары в последовательность следующим образом: упорядочим их по возрастанию суммы $i+j$, а при одинаковых суммах – по возрастанию первых элементов пар. Получим последовательность $(1,1)$, $(1,2)$, $(2,1)$, $(1,3)$, $(2,2)$, $(3,1)$,… . В ней первой стоит единственная пара с суммой 2, за ней следуют две пары с суммой 3, затем три пары с суммой 4, и т.д. Ясно, что каждая пара встретится в этой последовательности, причем ровно один раз.

4. Множество $\mathbb{Q}$ всех рациональных чисел счетно. Рассмотрим сначала множество $\mathbb{Q}^+$, состоящее из всех положительных рациональных чисел. Каждое такое число однозначно представляется несократимой дробью $\frac{a}{b}$ и ему можно поставить в соответствие пару $(a,b) \in \mathbb{N}^2$. Таким образом, имеется инъекция из $\mathbb{Q}^+$ в $\mathbb{N}^2$, поэтому $|\mathbb{Q}^+| \le |\mathbb{N}^2| = |\mathbb{N}|$. С другой стороны, множество $\mathbb{N}$ является подмножеством множества $\mathbb{Q}^+$, значит, $|\mathbb{N}| \le |\mathbb{Q}^+|$. Из двух неравенств следует равенство $|\mathbb{Q}^+| = |\mathbb{N}|$. Следовательно, множество $\mathbb{Q}^+$ счетно и его элементы можно расположить в последовательность: $\mathbb{Q}^+ = \{a_1, a_2, a_3, \ldots\}$. Вставим в эту последовательность после каждого числа то же самое число со знаком минус, а в начале ее добавим число 0. В результате получится последовательность $\{0, a_1, -a_1, a_2, -a_2, a_3, -a_3, \ldots\}$, содержащая все рациональные числа.

Следующая теорема показывает, что счетные множества – это в известном смысле «самые малые» бесконечные множества.

Теорема 2.3. *Любое бесконечное множество имеет счетное подмножество.*

Доказательство. В бесконечном множестве A выберем какой-нибудь элемент x_1, в множестве $A - \{x_1\}$ – элемент x_2, в множестве $A - \{x_1, x_2\}$ – элемент x_3, и т.д. Поскольку A бесконечно, после выбора любого конечного числа элементов останется непустое подмножество и в нем можно выбрать следующий элемент. Таким образом, получаем счетное подмножество $\{x_1, x_2, x_3, \ldots\}$ множества A. ■

Бесконечное множество, не являющееся счетным, называется *несчетным*.

Теорема 2.4. *Множество* $\mathbb{R}$ *всех действительных чисел несчетно.*

Доказательство. Докажем, что несчетно множество вещественных чисел в интервале $[0,1)$. Каждое такое число представляется бесконечной десятичной дробью вида $0, a_1 a_2 \ldots$, где a_1, a_2 – десятичные цифры. Для некоторых чисел имеется два представления, например, $\frac{1}{10} = 0{,}1000 \ldots = 0{,}0999 \ldots$. Чтобы представление было однозначным, договоримся в таких случаях рассматривать только ту запись, которая содержит бесконечную последовательность нулей. Допустим, что это множество чисел счетно, то есть его можно расположить в бесконечную последовательность: $[0,1) = \{x_1, x_2, \ldots\}$. Пусть $x_n = 0, x_{n,1} x_{n,2} \ldots$ – десятичная запись числа x_n, $n = 1,2, \ldots$. Составим десятичную запись числа $y = 0, y_1 y_2 \ldots$ следующим образом. В качестве y_1 возьмем любую десятичную цифру, отличную от 9 и от $x_{1,1}$, в качестве y_2 – любую цифру, отличную от 9 и от $x_{2,2}$, … , в качестве y_n – любую цифру, отличную от 9 и от $x_{n,n}$, … . Тогда число y отличается от каждого из чисел x_n хотя бы одной цифрой и, значит, не входит в последовательность $\{x_1, x_2, \ldots\}$. ■

Про множество, равномощное множеству $\mathbb{R}$, говорят, что оно имеет *мощность континуума*. Множество точек любого отрезка (ненулевой длины), множество точек всей прямой, множество точек всей плоскости – все они имеют такую мощность. Мощность континуума имеет также множество $2^{\mathbb{N}}$ всех подмножеств множества $\mathbb{N}$.

Одна из самых знаменитых математических проблем связана с вопросом о том, существуют ли несчетные множества, мощность которых меньше мощности континуума. Континуум-гипотеза Кантора утверждает, что таких промежуточных мощностей не существует, то есть мощность континуума – следующая за счетной. Установлено, что эта гипотеза не может быть ни доказана, ни опровергнута в современной аксиоматической теории множеств.

Следующая теорема показывает, что наибольшей мощности не существует.

Теорема 2.5 (теорема Кантора). *Для любого множества* A *выполняется неравенство* $|2^A| > |A|$.

Доказательство. Для любого множества A отображение, ставящее каждому элементу $x \in A$ множество $\{x\}$, является инъекцией A *в* 2^A. Следовательно, $|2^A| \geq |A|$. Допустим, существует биекция $f: A \to 2^A$. Рассмотрим подмножество $M \subseteq A$, состоящее из всех тех элементов x, для которых выполняется $x \notin f(x)$. Так как f – биекция, то существует такой элемент $m \in A$, что $f(m) = M$. Но тогда получается, что если $m \in f(m) = M$, то m не должен принадлежать M по определению этого множества. Если же $m \notin f(m)$, то, опять по определению множества M, элемент m должен

ему принадлежать. В любом случае получается противоречие, поэтому такой биекции f существовать не может. ■

Глава 3. Комбинаторика

3.1. Принципы подсчета

Комбинаторика (комбинаторный анализ) – раздел дискретной математики, в котором изучаются объекты, составленные из элементов конечных множеств. Одним из основных видов комбинаторных задач являются перечислительные задачи. В этих задачах требуется подсчитать число объектов, обладающих теми или иными свойствами. При решении многих комбинаторных задач используются следующие простые правила.

Правило равенства. *Если между двумя множествами имеется взаимно однозначное соответствие (то есть существует биекция одного из этих множеств на другое), то эти множества равномощны.*

Правило суммы. *Если конечные множества A и B не пересекаются, то* $|A \cup B| = |A| + |B|$.

Правило произведения. *Для любых конечных множеств* A *и* B *выполняется равенство* $|A \times B| = |A||B|$.

Правило суммы очевидно. Правило произведения можно аргументировать так: декартово произведение $A \times B$ состоит из пар вида (a, b), где $a \in A$, $b \in B$. Элемент a в такой паре может принимать $|A|$ различных значений, а при каждом значении a элемент b может принимать $|B|$ различных значений. Всего получается $|A||B|$ различных пар.

Правила суммы и произведения можно обобщить на любое число множеств.

Общее правило суммы. *Если никакие два из конечных множеств* $A_1, A_2, \dots, A_n$ *не пересекаются, то* $|A_1 \cup A_2 \cup \dots \cup A_n| = |A_1| + |A_2| + \cdots + |A_n|$.

Общее правило произведения. *Для любых конечных множеств* $A_1, A_2, \dots, A_n$ *выполняется равенство* $|A_1 \times A_2 \times \dots \times A_n| = |A_1||A_2| \dots |A_n|$.

Общее правило произведения легко доказать с помощью индукции по числу сомножителей n. Случай $n = 2$ рассмотрен выше. В общем случае множество $A_1 \times A_2 \times \dots \times A_n$ состоит из последовательностей вида $(a_1, a_2, \dots, a_n)$, где $a_i \in A_i$ для $i = 1,2, \dots, n$. Такую последовательность можно рассматривать, как пару, в которой первый элемент – это последовательность $(a_1, a_2, \dots, a_{n-1})$, принадлежащая множеству $A_1 \times A_2 \times \dots \times A_{n-1}$, а второй элемент a_n. Первый элемент по предположению

индукции принимает $|A_1||A_2|\ldots|A_{n-1}|$ значений, а второй $|A_n|$ значений. По правилу произведения для двух множеств получается $|A_1||A_2|\ldots|A_n|$ пар.

В случае, когда все сомножители одинаковые, из общего правила произведения получаем $|A^n| = |A|^n$.

3.2. Наборы и слова

Термин *набор* означает конечную последовательность элементов. Понятие набора отличается от понятия множества тем, что

- в наборе могут быть одинаковые элементы,
- порядок расположения элементов в наборе важен (наборы, состоящие из одинаковых элементов, расположенных в разном порядке, считаются различными).

Чтобы отличать наборы от множеств, будем применять круглые скобки для записи наборов: $(a_1, a_2, \ldots, a_n)$. Число элементов в наборе называется длиной набора.

Пример: (a, a, b, c, c, c) , (c, a, c, b, c, a) , (a, a, b, b, c, c) – разные наборы длины 6, составленные из элементов множества $\{a, b, c\}$.

Задача. Сколько различных наборов длины n можно составить из элементов множества мощности k?

Решение. Наборы длины n, составленные из элементов множества A, – это элементы декартовой степени A^n . Ответ дает общее правило произведения: число таких наборов равно k^n.

Пусть $A = \{a_1, a_2, \ldots, a_k\}$ – алфавит, то есть элементы множества A – это буквы. Из букв можно составлять слова: любая последовательность вида $a_{i_1} a_{i_2} \ldots a_{i_n}$ является словом длины n в алфавите A.

Задача. Сколько различных слов длины n можно составить из букв алфавита мощности k?

Решение. Слово – это упорядоченная последовательность букв, причем одна и та же буква может встречаться в слове несколько раз. Так какая же разница между словами и наборами? Чисто синтаксическая – в записи слов элементы (буквы) не разделяются запятыми и нет скобок. Принципиальной разницы нет, слова – это те же наборы. Поэтому и ответ в задаче о числе слов будет тот же самый: k^n.

Задача. Сколько существует различных бинарных отношений на множестве из n элементов?

Решение. Мы видели, что бинарное отношение на множестве из n элементов можно задать с помощью таблицы размера $n \times n$, заполненной нулями и единицами. Соответствие между отношениями и таблицами взаимно однозначное, так что число отношений равно числу способов заполнить такую таблицу. Если выписать строки таблицы одну за другой, получится слово длины n^2 в алфавите $\{0,1\}$. Таких слов имеется 2^{n^2} . Это и есть число отношений.

3.3. Лексикографический порядок

Для ускорения поиска слов в словарях их располагают в алфавитном порядке. В математике этот порядок называется *лексикографическим* и определяется следующим образом.

Сначала вводится отношение линейного порядка на множестве букв данного алфавита A. Для этого отношения будем использовать символ $\leq$, а для соответствующего строгого порядка – символ $<$ ($a < b$ означает, что $a \leq b$ и $a \neq b$). Обычно это тот порядок, в котором принято перечислять буквы алфавита. Например, в руссом алфавите А < Б < В < ⋯.

Определим сначала лексикографический порядок на множестве слов одинаковой длины. Пусть $\alpha = \alpha_1\alpha_2 \ldots \alpha_n$ и $\beta = \beta_1\beta_2 \ldots \beta_n$ – слова в алфавите A . Говорят, что слово α *лексикографически меньше* слова β (или α *лексикографически предшествует* β), если существует такое p, что $\alpha_1 = \beta_1$, ..., $\alpha_{p-1} = \beta_{p-1}$, $\alpha_p < \beta_p$.

Это проверяется так: два слова сравниваются буква за буквой слева направо. Сначала сравниваются первые буквы, если они совпадают, то сравниваются вторые и т.д. до тех пор, пока не обнаружатся первые слева несовпадающие буквы. Отношение между словами определяется отношением между этими несовпадающими буквами.

Будем использовать тот же символ $<$ для записи того, что одно слово лескикографически меньше другого и $\leq$ для нестрогой формы, т.е. с добавлением возможности равенства. Очевидно, отношение $\leq$ рефлексивно и антисимметрично. Докажем, что оно транзитивно.

В самом деле, пусть $\alpha \leq \beta$ и $\beta \leq \gamma$, где $\alpha = \alpha_1\alpha_2 \ldots \alpha_n$, $\beta = \beta_1\beta_2 \ldots \beta_n$, $\gamma = \gamma_1\gamma_2 \ldots \gamma_n$. Покажем, что $\alpha \leq \gamma$. Если $\alpha = \beta$ или $\beta = \gamma$ это очевидно. Допустим, $\alpha < \beta$ и $\beta < \gamma$. Тогда существуют такое p , что $\alpha_1 = \beta_1$, ..., $\alpha_{p-1} = \beta_{p-1}$, $\alpha_p < \beta_p$, и такое q , что $\beta_1 = \gamma_1$, …, $\beta_{q-1} = \gamma_{q-1}$, $\beta_q < \gamma_q$. Рассмотрим два случая: $p \leq q$ и $p > q$.

Если $p \le q$, то $\alpha_1 = \beta_1 = \gamma_1$, …, $\alpha_{p-1} = \beta_{p-1} = \gamma_{p-1}$, $\alpha_p < \beta_p \le \gamma_p$, следовательно, $\alpha < \gamma$. Если $p > q$, аналогично: $\alpha_1 = \beta_1 = \gamma_1$, …, $\alpha_{q-1} = \beta_{q-1} = \gamma_{q-1}$, $\alpha_q = \beta_q < \gamma_q$ и опять $\alpha < \gamma$.

Таким образом, при любом n отношение $\le$ является отношением порядка на множестве всех слов длины n в алфавите A. Очевидно также, что любые два слова лексикографически сравнимы, т.е. этот порядок – линейный.

Лексикографический порядок можно распространить на бесконечное множество всех слов в данном алфавите. Достаточно дополнить приведенное определение следующим пунктом: если слово α является началом слова β, то α лексикографически предшествует β.

3.4. Перестановки

Перестановкой из n различных элементов называется набор, в котором каждый из этих элементов встречается точно один раз. Удобно считать, что переставляемыми элементами являются числа $1,2,\dots,n$. Имеется одна перестановка из одного элемента и две перестановки из двух элементов: (1,2) и (2,1). Все перестановки из трех элементов можно получить, добавляя к каждой перестановке из двух элементов третий элемент всеми возможными способами. Имеется ровно три способа вставить элемент 3 в перестановку (1,2): (3,1,2), (1,3,2), (1,2,3). То же относится и к перестановке (2,1). Таким образом, получается $3 \cdot 2 = 6$ перестановок из трех элементов. В свою очередь, в каждую перестановку из трех элементов четвертый элемент можно вставить четырьмя способами и поэтому число перестановок из четырех элементов равно $4 \cdot 6 = 24$. Вообще, в каждую перестановку из элементов $1,2,\dots,n-1$ элемент n можно вставить n способами, поэтому число перестановок из n элементов в n раз больше числа перестановок из $n-1$ элемента. Отсюда следует, что число перестановок из n элементов равно

$$n(n-1)\dots 3 \cdot 2 = n!$$

Для строгого доказательства можно применить индукцию по n.

Другой способ вычислить число перестановок состоит в том, что мы рассматриваем перестановку $(p_1, p_2, \dots, p_n)$ как результат последовательного выбора элементов $p_1, p_2, \dots, p_n$ и подсчитываем число вариантов выбора на каждом шаге. Сначала выбираем элемент p_1. Это можно сделать n способами. После того, как p_1 выбран, для выбора элемента p_2 остается $n-1$ вариант, так как p_2 должен отличаться от p_1. Следовательно, пару (p_1, p_2) можно выбрать $n(n-1)$ способами. После того, как выбрана эта пара, для выбора элемента p_3 остается $n-2$ варианта – он должен отличаться и от p_1, и от p_2. Значит, тройку (p_1, p_2, p_3) можно выбрать

$n(n-1)(n-2)$ способами. В итоге приходим к той же формуле для числа перестановок.

3.5. Последовательный выбор

Принцип последовательного выбора элементов, с помощью которого только что была выведена формула для числа перестановок, применим к решению многих комбинаторных задач. Может показаться, что здесь применяется правило произведения, на самом деле это не так. Первый элемент p_1 мы выбираем из множества $\{1,2,\dots,n\}$. А вот множество, из которого выбирается элемент p_2, зависит от того, какой выбран элемент p_1: если $p_1 = 1$, то p_2 нужно выбирать из множества $\{2,3,\dots,n\}$, а если $p_1 = 2$, то из множества $\{1,3,\dots,n\}$ и т.д. Значит, здесь не идет речь о декартовом произведении некоторых фиксированных множеств и правило произведения, как оно сформулировано выше, не работает. На самом деле при подсчете таким последовательным способом важно не то, из какого множества делается выбор на каждом шаге, а то, что мощность этого множества не зависит от выбора, сделанного на предыдущих шагах. Сформулируем этот принцип подсчета в общем виде.

Теорема 3.1 (о последовательном выборе). *Пусть набор $(x_1, x_2, \dots, x_k)$ формируется в результате последовательного выбора элементов $x_1, x_2, \dots, x_k$, причем*

– элемент x_1 можно выбрать n_1 способами,

– при любом x_1 элемент x_2 можно выбрать n_2 способами,

– при любых x_1 и x_2 элемент x_3 можно выбрать n_3 способами,

$\vdots$

– при любых x_1, x_2, …, x_{k-1} элемент x_k можно выбрать n_k способами.

Тогда весь набор можно выбрать $n_1 n_2 \dots n_k$ способами.

Фактически это дальнейшее обобщение правила произведения. Доказательство этого утверждения точно такое же, как доказательство правила произведения для k множеств: вначале рассматривается случай $k = 2$, затем проводится индукция по k.

3.6. Размещения

Размещением из n по k называется набор, состоящий из k различных элементов, выбранных из n-элементного множества. Таким образом, перестановка – это размещение из n по n. Размещения из n по k называют также *k-перестановками из n элементов*. Обозначим число размещений из n по k через $P(n,k)$. Это число легко подсчитать с помощью правила последовательного выбора. Первый элемент размещения можно выбрать n способами, второй $n-1$ способом, ..., k -тый $n-(k-1)$ способами. Следовательно, $P(n,k)=n(n-1)\dots(n-k+1)$. Если правую часть умножить и разделить на $(n-k)!$, то в числителе образуется $n!$ и получаем формулу

$$P(n,k)=\frac{n!}{(n-k)!}.$$

3.7. Сочетания

Сочетанием из n по k называют k-элементное подмножество n-элементного множества. Число сочетаний из n по k обозначают C_n^k или $\binom{n}{k}$ (читается «из n по k»). Будем пользоваться вторым обозначением.

Приведем таблицу сочетаний из четырех элементов a,b,c,d.

$k=$	0	1	2	3	4
	$\emptyset$	$\{a\}$	$\{a,b\}$	$\{a,b,c\}$	$\{a,b,c,d\}$
		$\{b\}$	$\{a,c\}$	$\{a,b,d\}$	
		$\{c\}$	$\{a,d\}$	$\{a,c,d\}$	
		$\{d\}$	$\{b,c\}$	$\{b,c,d\}$	
			$\{b,d\}$		
			$\{c,d\}$		

Подсчитаем число сочетаний из n по k в общем случае. Если взять какое-нибудь сочетание из n по k и расположить его элементы в каком-нибудь порядке, то получится размещение из n по k. Число способов

упорядочить k элементов сочетания равно числу перестановок из k элементов, то есть $k!$. Значит, из одного сочетания, упорядочивая его элементы разными способами, можно получить $k!$ размещений. Понятно, что каждое размещение может быть получено таким образом из некоторого сочетания, причем только из одного. Значит, имеет место равенство

$$\binom{n}{k} k! = P(n,k).$$

Отсюда, учитывая, что $P(n,k) = \frac{n!}{(n-k)!}$, получаем формулу для числа сочетаний:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}\,. \qquad (1)$$

Задача. Сколько имеется слов длины n в алфавите $\{a,b\}$, в которых буква a встречается ровно k раз?

Решение. Пронумеруем позиции букв в слове числами 1, 2, …, n. Любое слово в алфавите $\{a,b\}$ можно задать так: указать номера позиций, в которых располагается буква a. В остальных позициях должна находится буква b. Если буква a встречается ровно k раз, то для задания такого слова нужно выбрать k позиций из n. Следовательно, имеется взаимно однозначное соответствие между словами длины n в алфавите $\{a,b\}$, в которых буква a встречается ровно k раз, и k-элементными подмножествами множества $\{1,2,\dots,n\}$. По правилу равенства число слов равно числу подмножеств, то есть $\binom{n}{k}$.

3.8. Бином Ньютона. Биномиальные коэффициенты

Вспомним школьную формулу:

$$(a+b)^2 = a^2 + 2ab + b^2.$$

Откуда взялся коэффициент 2? Произведение $(a+b)(a+b)$ раскрываем с помощью дистрибутивного закона, не группируя сомножители и слагаемые:

$$(a+b)(a+b) = a(a+b) + b(a+b) = aa + ab + ba + bb.$$

Слагаемые в правой части – это всевозможные слова длины 2 в алфавите $\{a,b\}$. Имеется 2 слова, содержащие каждую из букв a и b по одному разу, отсюда и коэффициент 2 перед произведением ab. Аналогично для третьей степени:

$$(a+b)^3 = (a+b)^2(a+b) = (aa+ab+ba+bb)(a+b) =$$

$$= aaa + aab + aba + abb + baa + bab + bba + bbb.$$

Слагаемые в правой части – всевозможные слова длины 3. Слов, содержащих ровно две буквы a, имеется ровно $\binom{3}{2} = 3$, а слов, содержащих ровно одну букву a, $\binom{3}{1} = 3$. Поэтому после приведения подобных коэффициенты при a^2b и ab^2 будут равны 3 и получаем известную формулу

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3.$$

В общем случае, раскрывая бином $(a+b)^n$ без группирования сомножителей и слагаемых, получим сумму, слагаемыми в которой являются всевозможные слова длины n в алфавите $\{a, b\}$. Это легко доказать индукцией по n:

$$(a+b)^n = (a+b)^{n-1}(a+b) = (\text{сумма всех слов длины } n-1)(a+b).$$

Когда раскроем скобки в последнем выражении, из каждого слова длины $n-1$ получатся два слова длины n добавлением в конце слова букв a и b. Таким образом, получатся все слова длины n.

Теперь, группируя сомножители, получим слагаемые вида $a^k b^{n-k}$, $k = 0,1,\dots,n$. Такое слагаемое с данным значением k входит в сумму столько раз, сколько имеется слов длины n, в которые буква a входит ровно k раз, то есть $\binom{n}{k}$ раз. Таким образом, группируя слагаемые, получим сумму

$$\binom{n}{0}a^0b^n + \binom{n}{1}a^1b^{n-1} + \binom{n}{2}a^2b^{n-2} + \cdots + \binom{n}{n}a^nb^0 = \sum_{k=0}^{n}\binom{n}{k}a^k\,b^{n-k}.$$

Тем самым доказана знаменитая формула бинома Ньютона.

Теорема 3.2 (бином Ньютона).

$$(a+b)^n = \sum_{k=0}^{n}\binom{n}{k}a^kb^{n-k}.$$

Таким образом, числа $\binom{n}{k}$ выступают в качестве коэффициентов в разложении бинома $(a+b)^n$ по степеням a и b. За это их называют *биномиальными коэффициентами.*

Отметим некоторые свойства биномиальных коэффициентов.

1. $\binom{n}{0} = 1$ при любом n.

Это получается, если в формуле (1) положить $k = 0$ и учесть, что $0! = 1$ по определению. Но и не пользуясь формулой (1) можно получить это равенство, если вспомнить смысл величины $\binom{n}{0}$ – это число подмножеств мощности 0 у множества из n элементов. Подмножество мощности 0 – это пустое множество, а оно единственно.

2. $\binom{n}{1} = n.$

Комбинаторный смысл этого равенства так же прозрачен, как и предыдущего – у множества из n элементов имеется ровно n одноэлементных подмножеств.

3. $\binom{n}{k} = \binom{n}{n-k}.$

Это получается, если в формуле (1) подставить $n - k$ вместо k. Но и здесь имеется простое комбинаторное доказательство: у подмножества мощности k дополнительное подмножество имеет мощность $n - k$. Таким образом, имеется взаимно однозначное соответствие между подмножествами мощности k и подмножествами мощности $n - k$. Отсюда и следует данное равенство.

4. $\binom{n}{k} = \frac{n(n-1)\ldots(n-k+1)}{k!}.$

Получается сокращением на $(n-k)!$ в правой части формулы (1). Это выражение удобно применять для вычисления величины $\binom{n}{k}$ при малых k, например,

$$\binom{n}{2} = \frac{n(n-1)}{2}.$$

5. $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.$

Это равенство легко выводится из формулы (1):

$$\binom{n-1}{k} + \binom{n-1}{k-1} = \frac{(n-1)!}{k!(n-k-1)!} + \frac{(n-1)!}{(k-1)!(n-k)!} =$$

$$= \frac{(n-1)!}{(k-1)!(n-k-1)!}\left(\frac{1}{k} + \frac{1}{n-k}\right) = \frac{(n-1)!}{(k-1)!(n-k-1)!}\frac{n}{k(n-k)} =$$

$$= \frac{n!}{k!(n-k)!} = \binom{n}{k}.$$

Последнее тождество используется для построения таблицы биномиальных коэффициентов, известной как *треугольник Паскаля*. В этой таблице биномиальные коэффициенты располагаются следующим образом:

$$\binom{0}{0}$$

$$\binom{1}{0} \quad \binom{1}{1}$$

$$\binom{2}{0} \quad \binom{2}{1} \quad \binom{2}{2}$$

$$\binom{3}{0} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{3}$$

$$. \quad . \quad . \quad . \quad . \quad . \quad . \quad .$$

Крайние элементы в каждой строке равны 1. Каждый из остальных элементов вычисляется как сумма двух расположенных над ним слева и справа элементов предыдущей строки:

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}$$

Таким образом, можно заполнять строчку за строчкой, используя только операцию сложения:

$$1$$

$$1 \quad 1$$

$$1 \quad 2 \quad 1$$

$$1 \quad 3 \quad 3 \quad 1$$

$$1 \quad 4 \quad 6 \quad 4 \quad 1$$

$$. \quad . \quad . \quad . \quad . \quad . \quad .$$

Многие тождества для биномиальных коэффициентов можно получить из бинома Ньютона. Приведем два из них.

6. Если в формуле бинома положить $a = b = 1$, получится равенство

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n.$$

Оно тоже имеет простой комбинаторный смысл. В левой части перечисляются подмножества мощности $0,1,2,\ldots,n$. Сумма должна быть равна числу всех подмножеств n-элементного множества. Но именно это число и написано в правой части.

7. Если положить $a=-1, b=1$, получим

$$\sum_{k=0}^{n}(-1)^k\binom{n}{k}=0.$$

Если все слагаемые со знаком минус перенести в правую часть, то получится равенство

$$\binom{n}{0}+\binom{n}{2}+\binom{n}{4}+\cdots=\binom{n}{1}+\binom{n}{3}+\binom{n}{5}+\cdots,$$

которое словами можно выразить так: у каждого конечного множества число подмножеств четной мощности равно числу подмножеств нечетной мощности.

3.9. Упорядоченные разбиения

Каким числом способов можно множество из n элементов разбить на k частей? Вопрос сформулирован не совсем корректно – нужно еще уточнить, считаем ли мы различными разбиения, отличающиеся только порядком частей. Например,

$$\{1,2,3,4,5,6\}=\{1,6\}\cup\{2\}\cup\{3,4,5\}$$

и

$$\{1,2,3,4,5,6\}=\{2\}\cup\{3,4,5\}\cup\{1,6\}$$

– это одно и то же разбиение или разные? Если такие разбиения считаются разными, то есть порядок частей важен, то говорят, что рассматриваются *упорядоченные разбиения*. Если же разбиения, отличающиеся только порядком частей, считаются одинаковыми, то имеются в виду *неупорядоченные разбиения*. Сейчас займемся подсчетом упорядоченных разбиений, неупорядоченные будут рассмотрены позже.

Итак, сколько существует упорядоченных разбиений множества из n элементов на k частей? Еще один пункт нужно уточнить: могут ли некоторые части разбиения быть пустыми множествами? Сейчас будем считать, что пустые части допускаются.

Пусть $a_1, a_2, \ldots, a_n$ – элементы разбиваемого множества. Части разбиения нумеруем числами от 1 до k. Разбиение можно построить так.

Берем элемент a_1 и назначаем ему часть, которой он будет принадлежать, скажем, часть с номером i_1. Затем так же поступаем с элементом a_2, назначая ему часть с номером i_2, и т.д. В результате получается набор $(i_1, i_2, \dots, i_n)$, в котором для каждого элемента указано, какой части разбиения он принадлежит. Элементы набора принадлежат множеству $\{1,2, \dots, k\}$. Ясно, что имеется взаимно однозначное соответствие между такими наборами и разбиениями. Значит, число разбиений равно числу наборов.

Теорема 3.3. *Число упорядоченных разбиений множества из n элементов на k частей, среди которых могут быть пустые, равно k^n.*

Рассмотрим теперь упорядоченные разбиения с заданными размерами частей. Теперь кроме параметров n и k заданы еще числа $n_1, n_2, \dots, n_k$, причем $n_1 + n_2 + \cdots + n_k = n$. Требуется подсчитать число таких разбиений множества из n элементов на части $P_1, \dots, P_k$, у которых $|P_1| = n_1, \dots, |P_k| = n_k$.

Эта задача легко решается с помощью последовательного выбора. Часть P_1 должна быть подмножеством мощности n_1 множества мощности n, следовательно, ее можно выбрать $\binom{n}{n_1}$ способами. После того, как часть P_1 выбрана, в множестве осталось $n - n_1$ элементов и из них нужно выбрать n_2 элементов, чтобы образовать часть P_2. Это можно сделать $\binom{n-n_1}{n_2}$ способами. Часть P_3 можно выбрать $\binom{n-n_1-n_2}{n_3}$ способами и т.д. По правилу последовательного выбора, перемножая эти числа, получим искомое число разбиений:

$$\binom{n}{n_1}\binom{n-n_1}{n_2}\binom{n-n_1-n_2}{n_3}\cdots\binom{n-n_1-\cdots-n_{k-1}}{n_k} =$$

$$= \frac{n!}{n_1!(n-n_1)!}\frac{(n-n_1)!}{n_2!(n-n_1-n_2)!} \cdots \frac{(n-n_1-\cdots-n_{k-1})!}{n_k!(n-n_1-\cdots-n_{k-1}-n_k)!} =$$

$$= \frac{n!}{n_1!n_2!\dots n_k!}.$$

Это число обозначают через $\binom{n}{n_1, n_2, \dots, n_k}$ и называют *полиномиальным коэффициентом* (почему, выяснится немного позже).

Теорема 3.4. *Число упорядоченных разбиений множества мощности n на k частей мощностей $n_1, n_2, \dots, n_k$ равно*

$$\binom{n}{n_1, n_2, \dots, n_k} = \frac{n!}{n_1!n_2!\dots n_k!}.$$

Задача. Сколько имеется слов длины n в алфавите $A = \{a_1, a_2, \dots, a_k\}$, в которых буква a_1 встречается n_1 раз, буква $a_2 - n_2$ раза, …, буква $a_k - n_k$ раз $(n_1 + n_2 + \cdots + n_k = n)$?

Решение. Пронумеруем позиции букв в слове числами 1, 2, …, n. Чтобы построить слово с заданными параметрами, нужно выбрать n_1 позиций для буквы a_1, n_2 позиций для буквы a_2 и т.д. Пусть P_i – множество позиций, выбранных для буквы a_i, $i = 1,2, \dots, k$. Тогда семейство множеств $\{P_1, P_2, \dots, P_k\}$ является разбиением множества $\{1,2, \dots, n\}$, причем части разбиения имеют мощности $n_1, n_2, \dots, n_k$. Следовательно, искомое число слов равно $\binom{n}{n_1, n_2, \dots, n_k}$.

3.10. Полиномиальная теорема

Полиномиальная теорема – это обобщение бинома Ньютона.

Теорема 3.5.

$$(a_1 + a_2 + \cdots + a_k)^n = \sum_{n_1+n_2+\cdots+n_k=\mathrm{n}} \binom{n}{n_1, n_2, \dots, n_k} {a_1}^{n_1} {a_2}^{n_2} \dots {a_k}^{n_k}.$$

Доказательство этой формулы можно провести по тому же плану, что и приведенное выше доказательство бинома Ньютона. Раскрывая скобки, не группируя при этом сомножителей, получаем сумму, слагаемыми в которой выступают всевозможные слова длины n в алфавите $\{a_1, a_2, \dots, a_k\}$. Группируя затем сомножители, получаем слагаемые вида ${a_1}^{n_1} {a_2}^{n_2} \dots {a_k}^{n_k}$, причем такое слагаемое появляется столько раз, сколько имеется слов с соответствующим составом букв, то есть $\binom{n}{n_1, n_2, \dots, n_k}$ раз.

3.11. Сочетания с повторениями

Сочетания с повторениями – это мультимножества, составленные из элементов данного множества. Общее число вхождений элементов в мультимножество будем называть *размером* этого мультимножества. Например, $\{a, a, b, b, b, c\}$ – мультимножество размера 6. Сочетания с повторениями из n по k – это мультимножества размера k, составленные из элементов множества мощности n.

Из трех элементов a, b, c можно составить 10 мультимножеств размера 3:

$\{a, a, a\}$ $\{a, a, c\}$ $\{a, b, c\}$ $\{b, b, b\}$ $\{b, c, c\}$
$\{a, a, b\}$ $\{a, b, b\}$ $\{a, c, c\}$ $\{b, b, c\}$ $\{c, c, c\}$

Подсчитаем число сочетаний с повторениями из n по k в общем случае.

Пусть $A = \{a_1, a_2, \dots, a_n\}$. Мультимножеству размера k, состоящему из элементов множества A, поставим в соответствие слово в алфавите $\{0,1\}$ следующим образом. Это слово будет состоять из n групп нулей, разделенных единицами (число таких разделяющих единиц равно $n-1$): число нулей в первой группе равно числу вхождений элемента a_1 в данное мультимножество, во второй – числу вхождений a_2 и т.д.

Например, пусть $k = 9, n = 5$. Мультимножеству

$$\{a_1, a_1, a_1, a_1, a_2, a_2, a_4, a_4, a_4\}$$

ставится в соответствие слово

$$0000100110001.$$

В общем случае получится слово из k нулей и $n-1$ единиц. Ясно, что каждое слово с такими параметрами соответствует некоторому мультимножеству. Таким образом, имеется биекция между множеством всех сочетаний с повторениями из n по k и множеством всех слов длины $n+k-1$, состоящих из k нулей и $n-1$ единиц. Значит, число сочетаний с повторениями равно числу таких слов, то есть $\binom{n+k-1}{k}$.

Теорема 3.6. *Число сочетаний с повторениями из n по k равно* $\binom{n+k-1}{k}$.

3.12. Формула включений-исключений

Правило суммы гласит, что

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

для любых непересекающихся конечных множеств A и B. Если множества пересекаются, равенство будет неверным, так как элементы, принадлежащие обоим множествам, оказываются сосчитанными дважды в правой части. Но это легко исправить – нужно вычесть из суммы мощностей число элементов, сосчитанных дважды, то есть мощность пересечения множеств. Получается равенство

$$|A \cup B| = |A| + |B| - |A \cap B|,$$

верное для любых конечных множеств.

Аналогичное равенство справедливо для объединения трех множеств:

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

Сначала складываются мощности трех множеств, затем вычитаются мощности всевозможных пересечений двух множеств и прибавляется мощность пересечения всех трех множеств. Чтобы убедиться в справедливости этого равенства для любых трех конечных множеств, достаточно проверить, что каждый элемент множества $A \cup B \cup C$ сосчитан в правой части ровно один раз. При этом нужно учесть, что элемент может принадлежать одному, двум или всем трем множествам A, B, C и рассмотреть все эти возможности. Если элемент принадлежит только одному из трех множеств, скажем, A, то он учтен только в слагаемом $|A|$, следовательно, сосчитан один раз. Если элемент принадлежит в точности двум множествам, скажем, A и B, то он учтен в слагаемых $|A|$ и $|B|$ со знаком плюс и в слагаемом $|A \cap B|$ со знаком минус, значит, общий вклад этого элемента равен $1 + 1 - 1 = 1$. Элемент, принадлежащий всем трем множествам, присутствует во всех слагаемых и его вклад равен $1 + 1 + 1 - 1 - 1 - 1 + 1 = 1$. Таким образом, вклад каждого элемента из множества $A \cup B \cup C$ в правую часть равенства равен 1, следовательно, вся сумма равна числу элементов в этом множестве.

Можно написать подобное равенство для четырех множеств:

$$|A \cup B \cup C \cup D| = |A| + |B| + |C| + |D| -$$
$$-|A \cap B| - |A \cap C| - |A \cap D| - |B \cap C| - |B \cap D| - |C \cap D| +$$
$$+|A \cap B \cap C| + |A \cap B \cap D| + |A \cap C \cap D| + |B \cap C \cap D| -$$
$$-|A \cap B \cap C \cap D|$$

и доказать его аналогичными рассуждениями. Заметим, что слагаемые в правой части соответствуют всевозможным пересечениям множеств из семейства $\{A, B, C, D\}$, причем пересечения нечетного числа множеств входят со знаком плюс, а пересечения четного числа множеств – со знаком минус.

Рассмотрим общий случай. Пусть $A_1, A_2, \dots, A_n$ – конечные множества. Пересечение любых t из этих множеств назовем *t-пересечением*. Обозначим через S_t сумму мощностей всех t-пересечений этих n множеств:

$$S_t = \sum_{\{i_1, i_2, \dots, i_t\} \subseteq \{1,2,\dots,n\}} |A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_t}|.$$

Запись $\{i_1, i_2, \dots, i_t\} \subseteq \{1,2, \dots, n\}$, расположенная ниже знака суммы Σ, указывает, что суммирование производится по всем t-элементным подмножествам множества $\{1,2, \dots, n\}$. Для каждого такого подмножества $\{i_1, i_2, \dots, i_t\}$ вычисляется мощность t-пересечения $|A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_t}|$ и все эти мощности складываются. В частности,

$S_1 = |A_1| + |A_2| + \cdots + |A_n|$,

$S_2 = |A_1 \cap A_2| + |A_1 \cap A_3| + \cdots + |A_{n-1} \cap A_n|$ и т.д.

Теорема 3.7 (формула включений-исключений).

$$|A_1 \cup A_2 \cup \ldots \cup A_n| = S_1 - S_2 + S_3 - \cdots + (-1)^{n-1} S_n.$$

Доказательство. Возьмем любой элемент $x \in A_1 \cup A_2 \cup \ldots \cup A_n$ и обозначим через $s(x)$ вклад, вносимый этим элементом в сумму в правой части утверждения теоремы. Всякий раз, когда элемент x входит в пересечение некоторых из множеств $A_1, A_2, \ldots, A_n$, к этой сумме (и к $s(x)$) прибавляется единица, если это пересечение нечетного числа множеств, если же четного, то единица вычитается. Покажем, что $s(x) = 1$ для каждого $x \in A_1 \cup A_2 \cup \ldots \cup A_n$, отсюда будет следовать утверждение теоремы.

Допустим, что элемент x принадлежит в точности m из множеств $A_1, A_2, \ldots, A_n$. Тогда имеется

ровно m 1-пересечений (т.е. множеств A_i), содержащих x;

ровно $\binom{m}{2}$ 2-пересечений, содержащих x;

ровно $\binom{m}{3}$ 3-пересечений, содержащих x;

…

Вообще, для любого $t = 1,2, \ldots, m$ имеется ровно $\binom{m}{t}$ t-пересечений, содержащих x (вспомним, что $\binom{m}{1} = m$), а для $t > m$ таких t-пересечений нет. Следовательно,

$$s(x) = \binom{m}{1} - \binom{m}{2} + \binom{m}{3} - \cdots + (-1)^{m-1} \binom{m}{m}.$$

Одно из следствий бинома Ньютона гласит, что

$$\binom{m}{0} - \binom{m}{1} + \binom{m}{2} - \binom{m}{3} + \cdots + (-1)^m \binom{m}{m} = 0.$$

Отсюда

$$\binom{m}{1} - \binom{m}{2} + \binom{m}{3} - \cdots + (-1)^{m-1} \binom{m}{m} = \binom{m}{0} = 1,$$

следовательно, $s(x) = 1$. ■

Пример – задача о беспорядках. Пусть $p = (p_1, p_2, \ldots, p_n)$ – перестановка элементов $\{1,2, \ldots, n\}$. Если $p_i = i$ для некоторого i, то элемент i называется *неподвижной точкой* перестановки p. Перестановка, у которой нет неподвижных точек, называется *беспорядком*. Иначе говоря, в беспорядке ни один элемент не стоит на своем месте. Число беспорядков из n элементов обозначим через d_n. Легко проверить, например, что имеется ровно два беспорядка из трех элементов: (2,3,1) и (3,1,2), так что $d_3 = 2$.

Применим метод включения-исключения для подсчета числа беспорядков при произвольном n.

Обозначим через A_i множество всех таких перестановок n элементов, у которых i является неподвижной точкой. Тогда $A_1 \cup A_2 \cup \ldots \cup A_n$ – множество всех перестановок, имеющих неподвижные точки. Так как всего имеется $n!$ перестановок из n элементов, то

$$d_n = n! - |A_1 \cup A_2 \cup \ldots \cup A_n|.$$

Для подсчета мощности объединения множеств используем формулу включений-исключений:

$$|A_1 \cup A_2 \cup \ldots \cup A_n| = S_1 - S_2 + S_3 - \cdots - (-1)^{n-1} S_n,$$

где S_t – сумма мощностей всевозможных t-пересечений множеств $A_1, A_2, \ldots, A_t$. Каждое такое t-пересечение $A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_t}$ состоит из всех перестановок, у которых элементы $i_1, i_2, \ldots, i_t$ неподвижны. Число таких перестановок равно числу перестановок остальных $n - t$ элементов, то есть $(n-t)!$. Значит, в сумме S_t каждое слагаемое равно $(n-t)!$. Число слагаемых в этой сумме равно числу t-элементных подмножеств множества из n элементов, то есть $\binom{n}{t}$. Следовательно,

$$S_t = (n-t)!\binom{n}{t} = \frac{n!}{t!},$$

и получаем формулу для числа беспорядков:

$$d_n = n! - \frac{n!}{1!} + \frac{n!}{2!} - \cdots + (-1)^n \frac{n!}{n!} = n!\,(1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!}).$$

3.13. Неупорядоченные разбиения

В качестве еще одного применения метода включений-исключений выведем формулу для числа неупорядоченных разбиений.

На первый взгляд неупорядоченные и упорядоченные разбиения соотносятся между собой так же, как сочетания с размещениями. Из каждого сочетания из n по k, упорядочивая разными способами его элементы, можно получить $k!$ размещений из n по k. На этом основан вывод формулы для числа сочетаний: нам известно число размещений и известно, что их в $k!$ раз больше, чем сочетаний. Нельзя ли этот прием использовать для подсчета числа неупорядоченных разбиений? Мы знаем, что число упорядоченных разбиений множества из n элементов на k частей равно k^n. Каждое упорядоченное разбиение получается из неупорядоченного расстановкой его

частей в определенном порядке. Имеется $k!$ способов упорядочить k частей. Разве не так?

Нет, не так, и это видно из следующего примера. Рассмотрим разбиение множества из 5 элементов на 4 части:

$$\{1,2,3,4,5\} = \emptyset \cup \{2,4\} \cup \{1,3,5\} \cup \emptyset.$$

Первая и четвертая части одинаковы и при их перестановке получается то же самое упорядоченное разбиение. Поэтому перестановкой частей этого разбиения можно получить не $4! = 24$, а только 12 различных упорядоченных разбиений.

Понятно, что в этом примере все дело в том, что имеются две одинаковые части. Понятно также, что одинаковыми могут быть только пустые части. Если найти число упорядоченных разбиений множества из n элементов на k непустых частей, то для подсчета числа неупорядоченных разбиений достаточно будет разделить этот результат на $k!$.

Задача. Сколько имеется упорядоченных разбиений множества из n элементов на k непустых частей?

Решение. Применим метод включений-исключений. Пусть $P_1, P_2, \dots, P_k$ – части упорядоченного разбиения. Обозначим через A_i множество всех разбиений, у которых $P_i = \emptyset$. Тогда $A_1 \cup A_2 \cup \dots \cup A_k$ – множество разбиений, у которых хотя бы одна часть пустая, а число разбиений, не имеющих пустых частей, равно

$$k^n - |A_1 \cup A_2 \cup \dots \cup A_k|.$$

Для подсчета мощности объединения множеств используем формулу включений-исключений:

$$|A_1 \cup A_2 \cup \dots \cup A_k| = S_1 - S_2 + S_3 - \dots + (-1)^{k-1} S_k,$$

где S_t – сумма мощностей всевозможных t-пересечений множеств $A_1, A_2, \dots, A_k$. Каждое такое t-пересечение $A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_t}$ состоит из всех разбиений, у которых данные t частей пустые: $P_{i_1} = P_{i_2} = \dots = P_{i_t} = \emptyset$. Таких разбиений имеется ровно столько, сколько упорядоченных разбиений множества из n элементов на $k - t$ частей, то есть $(k-t)^n$. Следовательно, в сумме S_t каждое слагаемое равно $(k-t)^n$. Число слагаемых в этой сумме равно числу t-элементных подмножеств множества из k элементов, то есть $\binom{k}{t}$. Таким образом, $S_t = \binom{k}{t}(k-t)^n$, а искомое число разбиений, не имеющих пустых частей, равно

$$k^n - \binom{k}{1}(k-1)^n + \binom{k}{2}(k-2)^n - \dots + (-1)^k \binom{k}{k}(k-k)^n =.$$

$$= \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n.$$

Теперь можно получить формулу для числа неупорядоченных разбиений множества из n элементов на k непустых частей. Это число называется *числом Стирлинга второго рода* и обозначается через $S(n,k)$. Неупорядоченное разбиение с k непустыми частями можно упорядочить ровно $k!$ способами, поэтому

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i \binom{k}{i}(k-i)^n = \sum_{i=0}^{k}(-1)^i \frac{(k-i)^n}{i!(k-i)!}.$$

Число всех разбиений множества из n элементов на непустые части (с любым числом частей) называется *числом Белла* и обозначается через B_n. Таким образом,

$$B_n = \sum_{k=1}^{n} S(n,k) = \sum_{k=1}^{n}\sum_{i=0}^{k}(-1)^i \frac{(k-i)^n}{i!(k-i)!}.$$

Из теоремы о факторизации следует, что B_n – это число различных отношений эквивалентности, которые можно определить на множестве мощности n.

3.14. Функции

Многие комбинаторные задачи могут быть сформулированы на языке отображений: требуется подсчитать число функций, обладающих теми или иными свойствами. Рассмотрим несколько простых задач этого типа. Будем рассматривать функции, отображающие множество $\{1,2,\dots,n\}$ в множество $\{1,2,\dots,m\}$. Множество всех таких функций обозначим через $F_{n,m}$.

Задача. Сколько всего функций имеется в множестве $F_{n,m}$?

Решение. Каждая функция $f \in F_{n,m}$ может быть задана набором своих значений $(f(1), f(2), \dots, f(n))$. Каждый элемент такого набора может принимать любое из m значений. Соответствие между функциями и наборами значений взаимно однозначно. Следовательно, число функций равно числу наборов, то есть m^n.

Задача. Сколько биекций в множестве $F_{n,m}$?

Решение**.** Биекции существуют только при $n = m$. Если $f \in F_{n,n}$ – биекция, то набор значений $(f(1), f(2), \dots, f(n))$ является перестановкой элементов множества $\{1,2,\dots,n\}$. Обратно, всякая такая перестановка

определяет биекцию. Значит, число биекций равно числу перестановок, то есть $n!$.

Задача. Сколько в множестве $F_{n,m}$ инъективных функций?

Решение. Инъективные функции в $F_{n,m}$ существуют только при $n \leq m$. Функция $f \in F_{n,m}$ является инъекцией тогда и только тогда, когда в наборе $(f(1), f(2), \dots, f(n))$ все элементы различны. Значит, этот набор должен быть размещением из m по n. Число инъективных функций равно числу таких размещений, то есть $\frac{m!}{(m-n)!}$.

Задача. Сколько в множестве $F_{n,m}$ сюръективных функций?

Решение. Для существования сюръекций должно выполняться неравенство $n \geq m$. Пусть $f \in F_{n,m}$ – сюръекция. Для каждого $y \in \{1,2,\dots,m\}$ множество всех прообразов элемента y при отображении f обозначается через $f^{-1}(y)$, то есть $f^{-1}(y) = \{x: f(x) = y\}$. Так как f сюръекция, то все эти множества непустые. Набор множеств $(f^{-1}(1), f^{-1}(2), \dots, f^{-1}(m))$ является упорядоченным разбиением множества $\{1,2,\dots,n\}$ на m непустых частей. Обратно, каждое такое разбиение определяет сюръекцию. Следовательно, число сюръекций равно числу разбиений, то есть $\sum_{i=0}^{m}(-1)^i \binom{m}{i} (m-i)^n$.

Задача. Сколько в множестве $F_{n,m}$ строго возрастающих функций?

Решение. Для того, чтобы задать монотонно возрастающую функцию $f \in F_{n,m}$, достаточно выбрать некоторое -элементное подмножество множества $\{1,2,\dots,m\}$ в качестве множества ее значений. Затем упорядочиваем выбранные элементы по возрастанию: $a_1 < a_2 < \dots < a_n$ и полагаем $f(1) = a_1,\ f(2) = a_2, \dots, f(n) = a_n$. Следовательно, число возрастающих функций равно числу -элементных подмножеств множества мощности m, то есть числу сочетаний из m по n.

Задача. Сколько в множестве $F_{n,m}$ неубывающих функций?

Решение. Рассуждаем, как в предыдущем случае, но теперь нужно выбрать в качестве множества значений функции мультимножество размера n, составленное из элементов множества $\{1,2,\dots,m\}$. Число функций равно числу таких мультимножеств, то есть числу сочетаний с повторениями из m по n, а оно равно $\binom{m+n-1}{n}$.

Глава 4. Линейные рекуррентные уравнения

4.1. Рекуррентные уравнения

Пусть $x_0, x_1, x_2, \ldots$ – бесконечная последовательность чисел, в которой несколько начальных элементов (*начальные значения*) $x_0, x_1, \ldots, x_k$ заданы, а каждый из последующих вычисляется по предыдущим в соответствии с некоторым правилом. Если это правило задано в виде уравнения, то оно называется *рекуррентным уравнением*. Решением такого уравнения является формула, выражающая x_n как функцию от n и от начальных значений.

Примеры.

1. Рассмотрим уравнение $x_n = x_{n-1} + 2$ с начальным значением $x_0 = 0$. Очевидно, что решением является $x_n = 2n$. Если изменить начальное значение, скажем, положить $x_0 = 1$, то и решение будет другое: $x_n = 2n + 1$.

2. Уравнение $x_n = 2x_{n-1}$, начальное значение $x_0 = 1$. Как и в предыдущем примере, решение очевидно: $x_n = 2^n$.

3. Уравнение $x_n = nx_{n-1}$, начальное значение $x_0 = 1$. Можно догадаться, что решением является $x_n = n!$. Это легко проверить. При $n = 0$ имеем $0! = 1 = x_0$. При $n > 0$ подставляем в уравнение $n!$ вместо x_n и $(n-1)!$ вместо x_{n-1} и убеждаемся, что равенство справедливо: $n! = n \cdot (n-1)!$

4. Уравнение $x_n = 2x_{n-1} - x_{n-2}$, начальные значения $x_0 = 0, x_1 = 1$. Здесь нужны два начальных значения, поскольку в уравнении элемент последовательности x_n определяется по двум предыдущим элементам. Зная x_0 и x_1, можем последовательно вычислять следующие элементы: $x_2 = 2 \cdot 1 - 0 = 2$, $x_3 = 2 \cdot 2 - 1 = 3$, $x_4 = 2 \cdot 3 - 2 = 4$, … . Наблюдая несколько первых значений, можно предположить, что решением является $x_n = n$. Это предположение справедливо при $n = 0$ и $n = 1$, его справедливость при $n > 1$ подтверждается подстановкой в уравнение: $n = 2(n-1) - (n-2)$. Далее мы увидим, как можно решать подобные уравнения, не опираясь на догадки.

Рекуррентное уравнение вида

$$x_n = a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_k x_{n-k} + b\,,$$

где $a_1, \ldots, a_k, b$ – константы, называется *линейным рекуррентным уравнением порядка k с постоянными коэффициентами*. Если $b = 0$, оно называется *однородным*. Рассмотрим линейные рекуррентные уравнения порядка 1 и 2.

4.2. Линейные рекуррентные уравнения первого порядка

Линейное рекуррентное уравнение первого порядка имеет вид

$$x_n = ax_{n-1} + b.$$

Если задано начальное значение x_0, то можно вычислить любой элемент последовательности x_n: $x_1 = ax_0 + b$, $x_2 = ax_1 + b = a^2x_0 + ab + b$ и т.д. Найдем общую формулу для x_n. Рассмотрим сначала два частных случая.

1. $a = 1$. Уравнение имеет вид

$$x_n = x_{n-1} + b.$$

Решение очевидно:

$$x_n = x_0 + nb.$$

Это не что иное, как арифметическая прогрессия.

2. $b = 0$. Уравнение имеет вид

$$x_n = ax_{n-1}.$$

Решение тоже очевидно:

$$x_n = a^n x_0.$$

Это геометрическая прогрессия.

Теперь рассмотрим общий случай. Сделаем замену

$$x_n = y_n + s,$$

где y_n – новая неизвестная, s – константа, значение которой определим позже. Подставляя это в исходное уравнение, получаем $y_n + s = ay_{n-1} + as + b$, то есть

$$y_n = ay_{n-1} + (a-1)s + b.$$

Теперь выберем s так, чтобы $(a-1)s + b = 0$, то есть $s = \frac{b}{1-a}$. Это можно сделать, если $a \neq 1$. Но случай $a = 1$ был рассмотрен раньше и теперь мы можем считать, что $a \neq 1$. Уравнение приобретает вид

$$y_n = ay_{n-1}.$$

Этот тип уравнений был уже рассмотрен, решением является

$$y_n = a^n y_0.$$

Так как $y_n = x_n - s$, получаем $x_n - s = a^n(x_0 - s)$ и остается подставить выражение для s:

$$x_n = a^n\left(x_0 - \frac{b}{1-a}\right) + \frac{b}{1-a}.$$

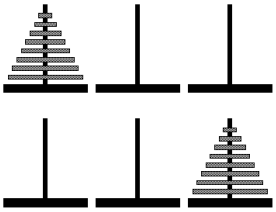
Это и есть решение уравнения при $a \neq 1$.

Замечание 1. Цель замены $x_n = y_n + s$ состоит в сведении неоднородного уравнения к однородному, которое затем легко решается.

Замечание 2. Решение уравнения имеет вид $x_n = \mathrm{c}_1 a^n + \mathrm{c}_2$, где a, c_1, c_2 – константы. Зависимость от n здесь выражена степенной функцией a^n. Это наблюдение оказывается полезным при переходе к уравнениям более высокого порядка, как будет видно далее.

Рассмотрим пример.

Ханойская башня. Три стержня установлены вертикально. На один из них нанизаны n дисков разного диаметра, диаметр убывает снизу вверх. Нужно переместить все диски на другой стержень. Разрешается перекладывать по одному диску, при этом нельзя класть больший диск на меньший. На рисунке показано начальное и финальное расположение дисков. Какое наименьшее число шагов требуется для перемещения всех дисков? (Шаг – перекладывание одного диска). Эту задачу придумал французский математик Люка в XIX в., ее называют задачей о Ханойской башне.



Решение. Обозначим через t_n наименьшее число шагов, требующееся для перемещения всех дисков с одного стержня на другой. Очевидно, $t_0 = 0$ (нет дисков – нет и перекладываний), $t_1 = 1$, $t_2 = 3$.

В общем случае $n-1$ меньших дисков должны быть перемещены на другой стержень прежде, чем мы сможем переложить самый большой диск. Это требует t_{n-1} шагов. Затем перекладывается большой диск. После этого нужно снова переместить $n-1$ меньших дисков так, чтобы уложить их поверх большого. Для этого опять требуется t_{n-1} шагов. Итак, для перемещения n дисков необходимо не менее $2t_{n-1}+1$ шагов. Но этого количества и достаточно, так как выше фактически описан план решения задачи: перекладываем $n-1$ меньших дисков, затем большой, затем снова $n-1$ меньших. Получаем рекуррентное уравнение

$$t_n = 2t_{n-1} + 1$$

с начальным значением $t_0 = 0$. Для перехода к однородному уравнению делаем замену $t_n = y_n + s$. Получаем $y_n + s = 2y_{n-1} + 2s + 1$. Полагаем

$s = -1$. Уравнение приобретает вид $y_n = 2y_{n-1}$, его решение $y_n = 2^n y_0$. Переходим к старым переменным: $y_n = t_n + 1$ и получаем ответ

$$t_n = 2^n - 1.$$

4.3. Линейные рекуррентные уравнения второго порядка

Как и уравнение первого порядка, неоднородное уравнение второго порядка можно с помощью подходящей замены свести к однородному. Поэтому мы будем рассматривать только однородное уравнение второго порядка, общий вид которого

$$x_n = a_1 x_{n-1} + a_2 x_{n-2}. \qquad (2)$$

Необходимо задать еще два начальных значения x_0 и x_1. Тогда, пользуясь равенством (2), можно последовательно вычислять элементы $x_2, x_3, \ldots$:

$x_2 = a_1 x_1 + a_2 x_0$

и т.д. Нашей целью опять является нахождение общей формулы для x_n.

Сначала займемся уравнением (2) безотносительно к начальным значениям. То есть считаем, что начальные значения не заданы, а под решением понимаем любую последовательность $\{x_n\}$, удовлетворяющую уравнению (2). Отметим два важных свойства решений.

(A). Если последовательность $\{x_n\}$ является решением уравнения (2) и d – любая константа, то последовательность $\{dx_n\}$ – тоже решение этого уравнения.

Чтобы в этом убедиться, достаточно подставить элементы последовательности $\{dx_n\}$ в уравнение: на d можно сократить и получится равенство для последовательности $\{x_n\}$.

(B). Если последовательности $\{x'_n\}$ и $\{x''_n\}$ являются решениями уравнения (2), то последовательность $\{x'_n + x''_n\}$ – тоже решение этого уравнения.

Для доказательства достаточно написать равенство (2) для каждой из последовательностей $\{x'_n\}$ и $\{x''_n\}$, а затем сложить эти два равенства, получится равенство (2) для последовательности $\{x'_n + x''_n\}$.

Теперь будем искать решение уравнения (2) в виде $x_n = \alpha^n$. Эта идея навеяна видом решения уравнения первого порядка (см. выше замечание 2).

Подставляя это выражение в уравнение (2) и сокращая на α^{n-2}, получаем квадратное уравнение с неизвестной α:

$$\alpha^2 - a_1\alpha - a_2 = 0,$$

оно называется *характеристическим уравнением*. Дальнейшие действия зависят от того, сколько корней имеет это уравнение – два или один. Рассмотрим оба случая.

1. Характеристическое уравнение имеет два различных корня α_1 и α_2. Тогда каждая из последовательностей $\{\alpha_1^n\}$ и $\{\alpha_2^n\}$ является решением уравнения (2). В силу свойств (А) и (В) при любых c_1 и c_2 последовательность

$$c_1\alpha_1^n + c_2\alpha_2^n \qquad (3)$$

тоже будет решением. Это общий вид решения уравнения (2).

Допустим теперь, что заданы начальные значения x_0 и x_1. Нельзя ли подобрать константы c_1 и c_2 так, чтобы последовательность (3) имела именно такие начальные значения? То есть при $n = 0$ и $n = 1$ элементы этой последовательности должны совпадать с x_0 и x_1:

$$c_1 + c_2 = x_0,$$

$$c_1\alpha_1 + c_2\alpha_2 = x_1.$$

Мы получили систему двух линейных уравнений с двумя неизвестными c_1 и c_2. Определитель этой системы

$\begin{vmatrix} 1 & 1 \\ \alpha_1 & \alpha_2 \end{vmatrix} = \alpha_2 - \alpha_1 \neq 0$, так как $\alpha_1 \neq \alpha_2$. Следовательно, существует единственное решение c_1, c_2 и мы получаем решение уравнения (2) с начальными значениями x_0, x_1.

2. Характеристическое уравнение имеет один корень α (т.е. $\alpha_1 = \alpha_2$). Тогда дискриминант этого уравнения равен 0, отсюда следует, что $a_2 = -a_1^2/4$ и $\alpha = a_1/2$. Используя эти соотношения, нетрудно прямой подстановкой показать, что в этом случае последовательность $\{n\alpha^n\}$ также будет решением уравнения (2). Комбинируя его с решением $\{\alpha^n\}$ на основе свойств (А) и (В), получаем общее решение

$$c_1\alpha^n + c_2 n\alpha^n.$$

Для вычисления констант c_1 и c_2 опять используются начальные значения:

$$c_1 = x_0,$$

$$c_1\alpha + c_2\alpha = x_1.$$

Рассмотрим пример.

Числа Фибоначчи. Найдем решение уравнения $f_n = f_{n-1} + f_{n-2}$ с начальными значениями $f_0 = 0$, $f_1 = 1$. Элементы этой последовательности называются *числами Фибоначчи*.

Решение. Характеристическое уравнение

$$\alpha^2 - \alpha - 1 = 0$$

имеет два корня $\alpha_1 = \left(1 + \sqrt{5}\right)/2$, $\alpha_2 = \left(1 - \sqrt{5}\right)/2$. Составляем уравнения для констант c_1 и c_2:

$$c_1 + c_2 = 0,$$

$$\alpha_1 c_1 + \alpha_2 c_2 = 1.$$

Отсюда находим $c_1 = 1/\sqrt{5}$, $c_2 = -1/\sqrt{5}$ и получаем формулу для чисел Фибоначчи (она известна как формула Бине):

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Глава 5. Графы

5.1. Основные понятия теории графов

С понятием графа мы уже встречались, когда рассматривали бинарные отношения. Напомним, что граф отношения – это диаграмма, наглядно изображающая отношение на конечном множестве с помощью стрелок, соединяющих элементы множества. При этом собственно графические подробности несущественны – неважно, какими значками изображены элементы множества, неважно, как выглядят стрелки, важно лишь, какие элементы и в каких направлениях эти стрелки соединяют. Поэтому в математическом определении понятия графа нет ничего графического или геометрического, а говорится лишь о неких элементах и их парах.

Граф состоит из двух множеств – конечного множества V, элементы которого называются *вершинами*, и множества E, состоящего из пар вершин, эти пары называются *ребрами*. Это записывают так: $G = (V, E)$, прочитать эту запись можно так: «граф G с множеством вершин V и множеством ребер E».

Мы уже сталкивались с тем, что пары бывают упорядоченные и неупорядоченные. Разница в том считаем ли мы пары (a, b) и (b, a) различными или одинаковыми. В приведенном определении об этом ничего не говорится и это требует уточнения. Различают два основных вида графов.

Ориентированный граф – ребрами являются упорядоченные пары вершин (*ориентированные ребра*).

Неориентированный граф – ребрами являются неупорядоченные пары вершин (*неориентированные ребра*).

В комбинаторике мы использовали круглые и фигурные скобки, чтобы различать упорядоченные совокупности (наборы) и неупорядоченные (множества). В теории графов сложилась традиция в обоих случаях употреблять круглые скобки. Поэтому по записи пары (a, b) не ясно, ориентированное это ребро или неориентированное. Обычно, начиная разговор о графах, заранее предупреждают, о каких графах пойдет речь.

Ребро типа (a, a) называют *петлей*.

Неориентированный граф, не имеющий петель, называется *обыкновенным графом*. Это один из наиболее распространенных видов графов. В дальнейшем, если не оговаривается иное, под графом будем понимать именно обыкновенный граф.

Для обозначения числа вершин и числа ребер графа будем обычно использовать буквы n и m.

Говорят, что ребро (a, b) *соединяет* вершины a и b, а вершины a и b являются *концами* этого ребра. Если в графе есть ребро (a, b), то говорят, что вершины a и b в нем *смежны*. Заметим, что в графе может быть не более одного ребра, соединяющего данную пару вершин.

Мультиграф – обобщение понятия графа. В мультиграфе могут быть *кратные ребра*, то есть несколько ребер, соединяющих одну и ту же пару вершин. В мультиграфе ребра – это не пары вершин, а самостоятельные объекты. При этом для каждого ребра должна быть указана пара вершин, которые это ребро соединяет.

Нетрудно подсчитать число графов с фиксированным множеством вершин. Обозначим через g_n число (обыкновенных) графов с множеством вершин $\{1,2, \dots, n\}$.

Теорема 5.1.

$$g_n = 2^{\frac{n(n-1)}{2}}.$$

Доказательство. Графы с одним и тем же множеством вершин различаются только множествами ребер. Каждое ребро – это неупорядоченная пара вершин. Множество всех таких пар состоит из $\binom{n}{2} = \frac{n(n-1)}{2}$ элементов. Значит, у него имеется $2^{\frac{n(n-1)}{2}}$ подмножеств, каждое из которых задает некоторый граф. ■

Пусть $e = (a, b)$ – ребро некоторого графа. Говорят, что ребро e *инцидентно* каждой из вершин a и b, а каждая из этих вершин инцидентна ребру e.

Число ребер, инцидентных вершине x в графе, называется *степенью* вершины x и обозначается через $\deg(x)$.

Если сложить степени всех вершин графа, то каждое ребро внесет в эту сумму вклад, равный 2. Следовательно, сумма степеней всех вершин равна удвоенному числу ребер графа. Это утверждение называют теоремой о рукопожатиях.

Теорема 5.2 (о рукопожатиях). *Для любого графа выполняется равенство*

$$\sum_{x \in V} \deg(x) = 2m.$$

Из этой теоремы следует, что в любом графе число вершин нечетной степени четно.

Существует много способов представить граф, назовем только самые распространенные.

1. Перечисление элементов. Исходя из определения, для того, чтобы задать граф, достаточно перечислить его вершины и ребра (т.е. пары вершин). Пусть, например, $V = \{a, b, c, d, e, f\}$, $E = \{(a, f), (a, d), (b, c), (c, d), (c, f)\}$. Тем самым задан граф с 6 вершинами и 5 ребрами.

2. Изображение. Если граф не очень велик, его можно нарисовать. Вершины изображают какими-нибудь значками (кружками, прямоугольниками и т.п.), ребра – в неориентированном графе ребра линиями, в ориентированном стрелками. На рисунке 5.1 показан граф, заданный выше перечислением вершин и ребер.

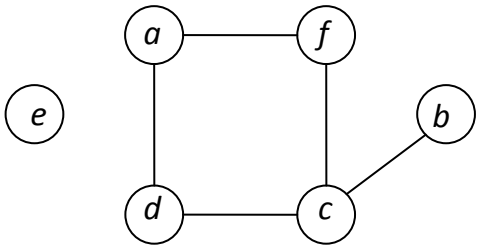


Рис. 5.1. Изображение графа

3. Матрица смежности. Это квадратная матрица порядка n. Для ее построения вершины графа нумеруются числами от 1 до n. Элемент матрицы, стоящий на пересечении строки с номером i и столбца с номером j, равен 1, если вершины с номерами i и j смежны, он равен 0, если эти вершины не смежны. Вот матрица смежности описанного выше графа (вершины пронумерованы в алфавитном порядке):

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Отметим две особенности матрицы смежности обыкновенного графа:

1) на главной диагонали стоят нули (нет петель);

2) матрица симметрична относительно главной диагонали (граф неориентированный).

4. Списки смежности. Этот способ часто используется для компьютерного представления графов. Состоит он в том, что для каждой вершины задается список всех смежных с ней вершин. Для рассматриваемого графа это может выглядеть так (пишется номер вершины и после двоеточия перечисляются номера смежных с ней вершин):

1: 4, 5
2: 3
3: 2, 4, 6
4: 1,3
5:
6: 1, 3

В структурах данных, применяемых в программировании, списки смежности могут быть реализованы как массив линейных списков.

Граф $G' = (V', E')$ называется *подграфом* графа $G = (V, E)$, если $V' \subseteq V$, $E' \subseteq E$. Всякий подграф может быть получен из графа удалением некоторых вершин и ребер (при удалении вершины удаляются и все инцидентные ей ребра).

Дополнением (*дополнительным графом*) к графу $G = (V, E)$ называется граф $\bar{G}$, у которого множество вершин то же, что у графа G, а множество ребер является дополнением множества E до множества всех неупорядоченных пар различных вершин. Иначе говоря, две различные вершины смежны в графе $\bar{G}$ тогда и только тогда, когда они не смежны в графе G. На рисунке 5.2 показаны граф, его подграф и дополнительный к нему граф.

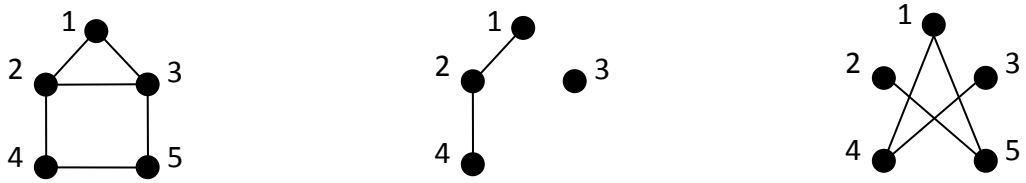


Рис. 5.2. Граф (слева), его подграф (в центре) и дополнительный граф (справа)

Некоторые часто встречающиеся графы имеют собственные названия и обозначения. Назовем наиболее важные из них

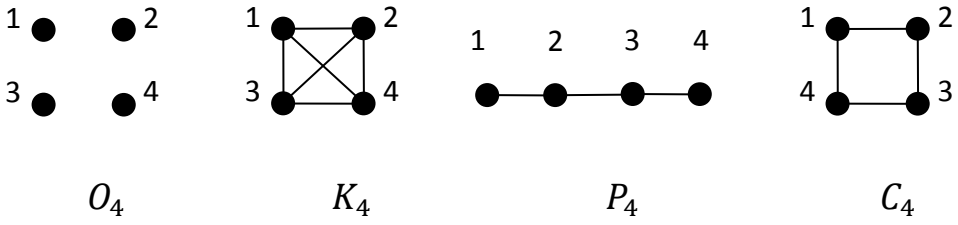
Пустой граф – граф, не содержащий ни одного ребра. Пустой граф с множеством вершин $\{1,2,\ldots,n\}$ обозначается O_n.

Полный граф – граф, в котором каждые две вершины смежны. Полный граф с множеством вершин $\{1,2,\ldots,n\}$ обозначается K_n.

Путь P_n имеет множество вершин $\{1,2,\dots,n\}$, ребрами его являются пары $(i, i+1)$, $i = 1,2,\dots n-1$.

Цикл C_n получается из графа P_n добавлением ребра $(1, n)$.

Все эти графы при $n = 4$ показаны на рисунке 5.3.



O_4 K_4 P_4 C_4

Рис. 5.3. Пустой граф, полный граф, путь и цикл

Следующие утверждения очевидны.

- $K_n = \overline{O_n}$.
- У всякого графа имеется пустой подграф.
- Всякий граф является подграфом полного графа.
- Граф P_n является подграфом графа C_n.
- Для всякого графа выполняется равенство $\bar{\bar{G}} = G$.
- Если граф G_1 является подграфом графа G_2, а граф G_2 – подграфом графа G_3, то G_1 – подграф графа G_3 (то есть отношение «быть подграфом» транзитивно).

5.2. Изоморфизм

На рисунке 5.4 изображены два графа с одним и тем же множеством вершин $\{a, b, c, d\}$. При внимательном рассмотрении можно обнаружить, что это разные графы – в графе, нарисованном слева, имеется ребро (a, d), в правом графе такого ребра нет. В то же время, если не обращать внимания на наименования вершин, то эти графы явно одинаково устроены: каждый из них – цикл из четырех вершин. Во многих случаях при исследовании

G_1 G_2

Рис. 5.4. Изоморфные графы

строения графов имена или номера вершин не играют роли, и такие графы, один из которых получается из другого переименованием вершин, удобнее было бы считать одинаковыми. Для того чтобы это можно было делать «на законном основании», вводится понятие изоморфизма графов. Если говорить не совсем формально, то два графа считаются изоморфными, если один из них можно превратить в другой переименованием вершин. Придание точного смысла понятию «переименование» приводит к следующему определению изоморфизма.

Определение. Графы $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называются *изоморфными*, если существует такая биекция f множества V_1 на множество V_2, что $(a, b) \in E_1$ тогда и только тогда, когда $(f(a), f(b)) \in E_2$. Отображение f в этом случае называется *изоморфизмом* графа G_1 на граф G_2.

Тот факт, что графы G_1 и G_2 изоморфны, записывается так: $G_1 \cong G_2$.

Для графов, изображенных на рисунке 4, изоморфизмом является, например, отображение, задаваемое таблицей:

x (вершина графа G_1)	a	b	c	d
$f(x)$ (вершина графа G_2)	a	b	d	c

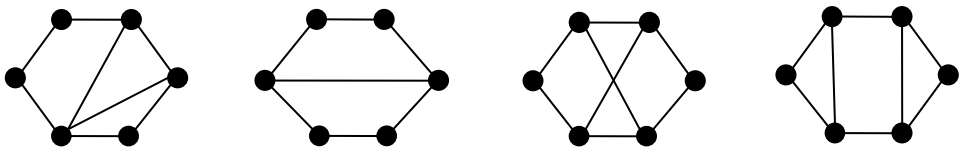
Заметим, что в этом примере есть и другие изоморфизмы первого графа на второй.

Сформулированное определение изоморфизма годится и для ориентированных графов, нужно только обе упоминаемые в нем пары вершин считать упорядоченными.

Изоморфизм – бинарное отношение на множестве графов. Очевидно, это отношение рефлексивно, симметрично и транзитивно, то есть является отношением эквивалентности. Классы эквивалентности называются *абстрактными графами*. Когда говорят, что рассматриваются абстрактные графы, это означает, что изоморфные графы считаются одинаковыми. Абстрактный граф можно представлять себе как граф, у которого стерты имена (пометки) вершин, поэтому абстрактные графы иногда называют также *непомеченными* графами.

Узнать, изоморфны ли два графа, бывает непросто. Если буквально следовать определению, то нужно перебрать все биекции множества вершин одного из них на множество вершин другого и для каждой из этих биекций проверить, является ли она изоморфизмом. Для n вершин имеется $n!$ биекций и эта работа становится практически невыполнимой уже для не очень

больших n (например, $20! > 2 \cdot 10^{18}$). Однако во многих случаях бывает довольно легко установить, что два данных графа неизоморфны. Рассмотрим, например, графы, изображенные на рисунке 5.5.



G_1 G_2 G_3 G_4

Рис. 5.5. Неизоморфные графы

Так как при изоморфизме пара смежных вершин переходит в пару смежных, а пара несмежных – в пару несмежных, то ясно, что число ребер у двух изоморфных графов должно быть одинаковым. Поэтому сразу можно сказать, что графы G_1 и G_2, у которых разное количество ребер, неизоморфны.

Характеристики графов (не обязательно количественные), которые сохраняются при изоморфизме, называются *инвариантами*. Иначе говоря, инвариант – это свойство графа, которое не зависит от того, как называются его вершины. Простейшие инварианты – число вершин и число ребер.

У графов G_1 и G_3 на рисунке 5.5 одинаковое число ребер, но они тоже неизоморфны. Это можно установить, сравнивая степени вершин. Очевидно, при изоморфизме каждая вершина переходит в вершину той же степени. Но у графа G_1 есть вершина степени 4, а у графа G_3 такой вершины нет. Можно ли сказать, что степень вершины является инвариантом? Нет, инвариант – это характеристика всего графа, а не одной вершины. А вот наличие у графа вершины данной степени – это инвариант. Число вершин данной степени – инвариант. Полную информацию о степенях дает *набор степеней* – последовательность степеней всех вершин графа, выписанных в неубывающем порядке. У графа G_1 набор степеней (2,2,2,3,3,4), а у графа G_3– (2,2,3,3,3,3).

С графами G_3 и G_4 дело обстоит немного сложнее – у них и наборы степеней одинаковы. Все же и эти графы неизоморфны: можно заметить, что в графе G_4 есть полный подграф из трех вершин, а в графе G_3 таких подграфов нет. Ясно, что при изоморфизме каждый подграф одного графа переходит в изоморфный ему подграф другого. Значит, наличие подграфа определенного вида является инвариантом.

Если удается установить, что для двух исследуемых графов некоторый инвариант принимает разные значения, то эти графы неизоморфны. Нельзя ли придумать такой инвариант или систему инвариантов, что совпадение всех этих инвариантов у двух исследуемых графов гарантировало бы, что эти графы изоморфны? Можно, и такие полные системы инвариантов известны. К сожалению, пока от них мало пользы, так как все известные полные

системы инвариантов требуют большого объема вычислений. Поиск быстро вычисляемых полных систем инвариантов ведется давно и пока без особых успехов.

Для того чтобы доказать, что два графа изоморфны, необходимо предъявить соответствующую биекцию. Например, графы, показанные на рисунке 5.6, изоморфны. Это доказывает таблица, представляющая изоморфизм между ними.

x	1	2	3	4	5	6
$f(x)$	4	6	1	5	3	2

Рис. 5.6. Изоморфные графы и изоморфизм между ними

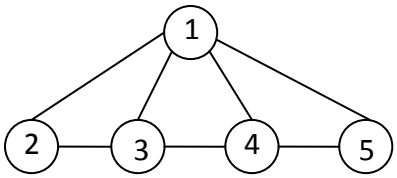
5.3. Пути и циклы

Путь и цикл в предыдущем разделе были определены как графы специального вида. Но в теории графов эти слова употребляются еще и в другом смысле.

Путь в графе – это последовательность вершин $x_1, x_2, \dots, x_k$, такая, что при каждом $i = 1,2, \dots, k-1$ пара (x_i, x_{i+1}) является ребром графа, причем все эти ребра различны. Эти $k-1$ ребер называются ребрами пути, а число $k-1$ – длиной пути. Путь *проходит* через вершины $x_1, x_2, \dots, x_k$ и *соединяет* вершины x_1 и x_k. Путь называется *простым*, если через каждую вершину он проходит не больше одного раза, то есть вершины $x_1, x_2, \dots, x_k$ все различны.

Цикл – это путь $x_1, x_2, \dots, x_k$, в котором $x_k = x_1$. Цикл *простой*, если вершины $x_1, x_2, \dots, x_{k-1}$ все различны.

На рисунке 5.7 показаны последовательности вершин разных типов.



2, 3, 4, 5, 1, 4, 3 – не путь (дважды проходит по ребру (3,4));
3, 1, 4, 5, 1, 2 – путь (не простой); 2, 3, 1, 4, 5 – простой путь;
2, 3, 1, 4, 5, 1, 2 – цикл (не простой); 2, 3, 4, 5, 1, 2 – простой цикл.

Рис. 5.7. Пути и циклы в графе

Теорема 5.3 (о существовании цикла). *Если в графе n вершин, m ребер и $m \geq n$, то в этом графе есть цикл.*

Доказательство. Допустим, что утверждение теоремы неверно, тогда существует граф с $m \geq n$, не имеющий циклов. Среди всех таких графов выберем граф G с наименьшим числом вершин.

В графе G не может быть вершин степени 0 и 1. Действительно, если есть такая вершина, то, удалив ее, получим граф, в котором число ребер по-прежнему не меньше числа вершин (число вершин уменьшилось на 1, а число ребер либо не изменилось, либо тоже уменьшилось на 1) и нет циклов. Но это противоречит выбору графа G, как графа с наименьшим числом вершин, имеющего такие свойства.

Итак, в графе G степень каждой вершины больше или равна 2. Выберем в этом графе простой путь наибольшей длины. Пусть это путь $P = x_1, x_2, \dots, x_k$. Степень вершины x_k не меньше 2, значит, кроме вершины x_{k-1}, в графе есть еще по крайней мере одна вершина, смежная с x_k. Пусть y – такая вершина. Если предположить, что y не принадлежит множеству $\{x_1, x_2, \dots, x_k\}$, то получается простой путь $x_1, x_2, \dots, x_k, y$, длина которого больше длины пути P. Значит, $y \in \{x_1, x_2, \dots, x_k\}$, то есть $y = x_i$ при некотором i, причем $i < k - 1$. Но тогда $y, x_{i+1}, x_{i+2}, \dots, x_k, y$ – цикл. ■

Граф называется *связным*, если для любых двух вершин в нем имеется путь, соединяющий эти вершины.

Если граф не связен, то он распадается на связные части, называемые *компонентами связности*. Компонента связности – это максимальный по включению связный подграф, то есть связный подграф, не являющийся частью другого связного подграфа. Граф на рисунке 5.8 имеет четыре компоненты связности.

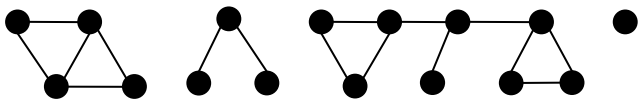


Рис. 5.8. Несвязный граф

Теорема 5.4 (о числе ребер в связном графе). *Если граф с n вершинами и m ребрами связен, то $m \geq n - 1$.*

Доказательство. Каждый граф с n вершинами можно построить, взяв пустой граф с n вершинами и добавляя к нему ребра. У пустого графа n компонент связности, при добавлении ребер их число будет уменьшаться. Допустим, мы уже построили некоторый граф G и добавляем к нему еще одно ребро. Как может измениться при этом число компонент связности?

Если две вершины этого нового ребра принадлежат одной компоненте связности графа G, то число компонент не изменится. Если же они принадлежат разным компонентам, то после добавления такого ребра эти две компоненты объединятся в одну и общее число компонент уменьшится на единицу. Итак, при добавлении нового ребра число компонент связности либо не изменяется, либо уменьшается на единицу. Значит, для превращения пустого графа в связный граф, то есть в граф с одной компонентой, число добавленных ребер должно быть не меньше $n - 1$. ■

Вершина графа называется *точкой сочленения* или *шарниром*, если в результате ее удаления число компонент связности увеличивается.

Ребро, при удалении которого увеличивается число компонент связности, называется *перешейком*.

У графа, изображенного на рисунке 8, имеется четыре шарнира и пять перешейков.

Теорема 5.5 (о перешейках). *Ребро является перешейком в том и только том случае, если в графе нет цикла, содержащего это ребро.*

Доказательство. Если ребро (a, b) принадлежит какому-нибудь циклу, то после удаления этого ребра вершины a и b оставшаяся часть этого цикла образует путь, соединяющий эти вершины. Поэтому число компонент связности не увеличится и это ребро – не перешеек.

Обратно, если ребро (a, b) не перешеек, то после его удаления вершины a и b останутся в одной компоненте связности. Значит, в графе с удаленным ребром имеется путь, соединяющий эти вершины. При добавлении к этому пути ребра (a, b) образуется цикл, содержащий это ребро. ■

5.4. Расстояния. Метрические характеристики

Расстояние между двумя вершинами в графе определяется как наименьшая длина пути, соединяющего эти вершины. Расстояние между вершинами a и b обозначается через $d(a, b)$. Если граф связный, то эта величина определена для любой пары вершин. Далее в этом разделе предполагаем, что рассматриваемый граф связен.

Отметим некоторые свойства расстояния в графе.

$d(a, b) = 0$ тогда и только тогда, когда $a = b$.

$d(a, b) = d(b, a)$.

$d(a,c) \leq d(a,b) + d(b,c)$.

Первые два свойства очевидны. Третье, оно называется *неравенством треугольника*, легко доказать: в объединении кратчайшего пути между a и b и кратчайшего пути между b и c содержится некоторый путь между a и c, но он не может быть короче кратчайшего пути между a и c.

В математике функцию двух переменных, обладающую свойствами 1) – 3), называют *метрикой или расстоянием*, а множество, на котором она определена – *метрическим пространством*. Наиболее известный пример – евклидово расстояние и евклидово пространство. Таким образом, всякий связный граф можно рассматривать как метрическое пространство.

Характеристики графа, связанные с расстоянием, называют *метрическими характеристиками*. Одна из них – *диаметр* графа, он определяется как наибольшее расстояние между вершинами. Диаметр графа G обозначается $\mathrm{diam}(G)$:

$$\mathrm{diam}(G) = \max_{x \in V} \max_{y \in V} d(x,y).$$

Наименьший возможный диаметр равен 1, такой диаметр имеет только полный граф. Наибольший возможный диаметр у связного графа с n вершинами равен $n-1$, таков диаметр графа P_n.

Эксцентриситет вершины – это расстояние от нее до самой удаленной вершины. Эксцентриситет вершины x обозначается $\mathrm{ecc}(x)$:

$$\mathrm{ecc}(x) = \max_{y \in V} d(x,y).$$

Диаметр графа, таким образом, – это максимальный из эксцентриситетов его вершин:

$$\mathrm{diam}(G) = \max_{x \in V} \mathrm{ecc}(x).$$

Минимальный из эксцентриситетов вершин называется *радиусом* графа и обозначается $\mathrm{rad}(G)$:

$$\mathrm{rad}(G) = \min_{x \in V} \mathrm{ecc}(x).$$

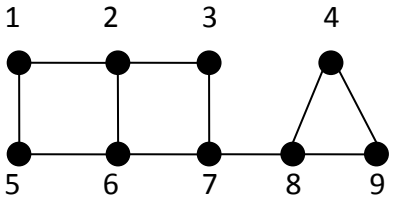
У полного графа эксцентриситеты всех вершин равны 1, значит, и его радиус равен 1. У графа P_n радиус равен $\frac{n}{2}$ при четном n и $\frac{n-1}{2}$ при нечетном. Используя обозначение $\lfloor x \rfloor$ для целой части числа x, можно написать: $\mathrm{rad}(P_n) = \left\lfloor \frac{n}{2} \right\rfloor$.

Вершина графа, имеющая минимальный эксцентриситет, называется *центральной*. Иначе говоря, центральная вершина – это такая, эксцентриситет которой равен радиусу графа. *Центр* графа – множество всех

его центральных вершин, он обозначается $C(G)$. У полного графа все вершины центральные, а у графа P_n одна или две центральные вершины, в зависимости от четности числа n.

На рисунке 5.9 показаны граф и таблица эксцентриситетов его вершин.

Радиус этого графа равен 3, диаметр равен 5, центр состоит из вершин 3, 6, 7.



x	1	2	3	4	5	6	7	8	9
ecc(x)	5	4	3	5	4	3	3	4	5

Рис. 5.9. Граф и эксцентриситеты его вершин

Теорема 5.6 (о радиусе и диаметре). *Для любого связного графа G выполняются неравенства*

$$\mathrm{rad}(G) \le \mathrm{diam}(G) \le 2\mathrm{rad}(G).$$

Доказательство. Левое неравенство очевидно, так как радиус – это минимальный эксцентриситет, а диаметр – максимальный. Докажем правое неравенство.

Пусть x и y – вершины, расстояние между которыми максимально в графе, то есть $d(x,y) = \mathrm{diam}(G)$. Пусть z – центральная вершина, то есть $\mathrm{ecc}(z) = \mathrm{rad}(G)$. Применим неравенство треугольника:

$\mathrm{diam}(G) = d(x,y) \le d(x,z) + d(z,y) \le 2\mathrm{ecc}(z) = 2\mathrm{rad}(G)$. ■

5.5. Эйлеровы циклы и пути

В 1736 году Леонард Эйлер сообщил в письме о своем решении задачи о Кенигсбергских мостах. Это событие считается моментом рождения теории графов, хотя называться теорией графов она стала двумя столетиями позже.

В городе Кенигсберге (ныне Калининград) было семь мостов через реку Преголь, соединяющих разные части суши (два берега и два острова). Схема расположения мостов показана на рисунке 5.10 слева, буквами обозначены участки суши. В популярной в то время головоломке спрашивалось, можно ли придумать такой маршрут, чтобы обойти все мосты, пройдя по каждому один раз, и вернуться в исходную точку. Эйлер

доказал, что это невозможно, и дал общее правило, с помощью которого можно для любой системы мостов определить, существует ли такой обход.

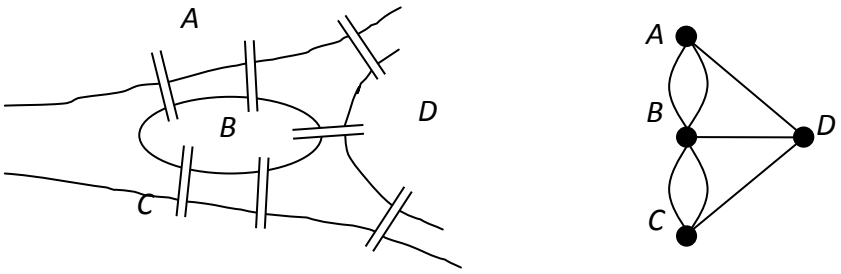


Рис. 5.10. Схема кенигсбергских мостов и ее представление мультиграфом

Задача о кенигсбергских мостах легко переводится на язык теории графов. Поставим в соответствие каждой части суши вершину графа, а каждому мосту – ребро. Получается мультиграф, изображенный на рисунке 5.10 справа. Нужно обойти все ребра этого мультиграфа, пройдя по каждому один раз, и вернуться в стартовую вершину. Рассмотрим эту задачу в общем случае, только для обыкновенных графов (решение, которое будет получено, можно распространить и на мультиграфы, но для этого нужно определить понятия пути и цикла в мультиграфе, чего мы делать не будем).

Определение. Цикл, проходящий через все ребра графа, называется *эйлеровым циклом*.

Цикл, проходя через вершину, каждый раз использует два ребра – по одному входит, по другому выходит. Если он проходит через вершину k раз, то будут использованы $2k$ инцидентных этой вершине ребер. Если этот цикл эйлеров, то все ребра будут пройдены, значит, степень вершины – четное число. Это относится к каждой вершине. Значит, необходимое условие существования эйлерова цикла – четность степеней всех вершин графа (заметим, что в мультиграфе задачи о кенигсбергских мостах степени всех вершин нечетны).

Теорема 5.7 (об эйлеровом цикле). *Эйлеров цикл в связном графе существует тогда и только тогда, когда степени всех вершин этого графа четны*.

Доказательство. О необходимости условия четности уже было сказано. Докажем его достаточность.

Пусть G – связный граф с четными степенями всех вершин. Выберем в этом графе самый длинный путь $P = x_1, x_2, \dots, x_k$. Покажем, что этот путь – цикл и что он содержит все ребра графа, т.е. P – эйлеров цикл.

Допустим, что P – не цикл, т.е. $x_1 \neq x_k$. Путь P проходит через все ребра, инцидентные вершине x_k, иначе он не был бы самым длинным путем

(если есть не пройденное ребро (x_k, y), то вершину y можно добавить к пути). Если вершина x_k встречается в пути P ровно s раз, то $s-1$ раз при ее прохождении используются два инцидентных ей ребра, а последний раз – одно. Значит, всего имеется $2(s-1)+1$ ребер, инцидентных этой вершине. Но это противоречит четности ее степени. Значит, P – цикл, $P = x_1, x_2, \dots, x_{k-1}, x_1$.

Допустим, что P проходит не через все ребра графа. Так как граф связен, то имеется не пройденное ребро, инцидентное какой-нибудь вершине из P. Пусть это ребро (x_i, z) (вершина z может принадлежать, а может не принадлежать пути). Но тогда можно построить более длинный путь:

$$x_i, x_{i+1}, \dots, x_{k-1}, x_1, x_2, \dots, x_i, z,$$

а это противоречит выбору пути P. ■

Если не требовать возвращения в исходный пункт, то получается задача об эйлеровом пути.

Определение. Путь, проходящий через все ребра графа, называется *эйлеровым путем*.

Теорема 5.8 (об эйлеровом пути). *Эйлеров путь в связном графе существует тогда и только тогда, когда в нем имеется не более двух вершин нечетной степени*.

Доказательство. Необходимость доказывается так же, как для эйлерова цикла. Если имеется эйлеров путь, то всякая вершина, кроме начальной и конечной вершин пути, должна иметь четную степень. Докажем достаточность условия.

Если нет вершин нечетной степени, то по предыдущей теореме существует эйлеров цикл, а цикл – это частный вид пути.

Не может быть в графе только одна вершина нечетной степени, так как сумма степеней должна быть четной (теорема о рукопожатиях).

Остается рассмотреть случай, когда в связном графе G есть ровно две вершины нечетных степеней, a и b. Построим новый граф H, добавив к графу G новую вершину c и ребра (a, c) и (b, c). В графе H степени всех вершин четны. Значит, в нем существует эйлеров цикл. Пусть $x_1, x_2, \dots, x_{k-1}, x_1$ – такой цикл. Существует единственное i, при котором $x_i = c$. Тогда путь $x_{i+1}, x_{i+2}, \dots, x_{k-1}, x_1, x_2, \dots, x_{i-1}$ является эйлеровым путем в графе G. ■

5.6. Деревья

Связный граф, не имеющий циклов, называется *деревом***.**

Если в графе нет циклов, то каждая его компонента связности является деревом. Такой граф называют *лесом*.

На рисунке 5.11 показаны все непомеченные деревья с числом вершин, не превосходящим 5.

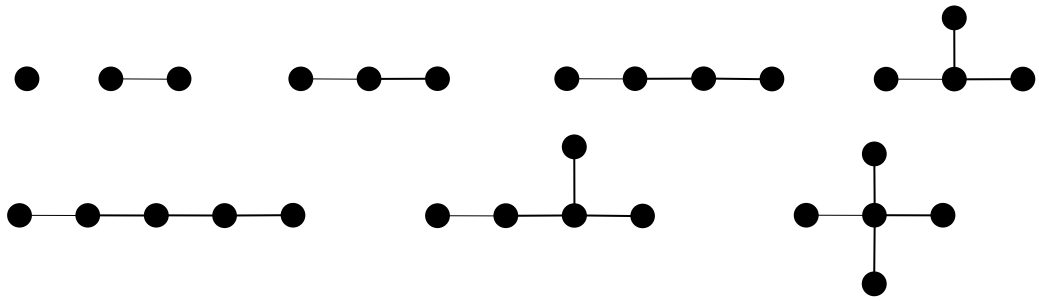


Рис. 5.11. Все деревья с числом вершин 1, 2, 3, 4, 5

Напомним, что мы договорились обозначать число вершин графа буквой n, а число ребер буквой m.

Теорема 5.9 (о свойствах деревьев).

1) *Для любого дерева справедливо равенство* $m = n - 1$.

2) *Если в графе нет циклов и* $m = n - 1$*, то этот граф – дерево.*

3) *Если граф связен и* $m = n - 1$*, то этот граф – дерево.*

Доказательство. Согласно теореме 5.3, если $m \geq n$, то в графе есть цикл. Так как в дереве циклов нет, то должно выполняться неравенство $m \leq n - 1$. Так как дерево – связный граф, то по теореме 5.4 выполняется неравенство $m \geq n - 1$. Из этих двух неравенств следует, что $m = n - 1$. Первое утверждение теоремы доказано.

Докажем второе утверждение. Пусть G – граф без циклов, с n вершинами и m ребрами, причем $m = n - 1$. Нужно доказать, что граф G связен. Допустим, что это не так. Возьмем две вершины из разных компонент связности и добавим к графу G ребро, соединяющее эти вершины. Ясно, что при добавлении такого ребра циклов не появится. Но в новом графе n вершин и n ребер, по теореме 5.3 в нем должен быть цикл. Противоречие. Значит, G – связный граф, то есть дерево.

Докажем третье утверждение. Пусть G – связный граф с n вершинами и m ребрами, причем $m = n - 1$. Нужно доказать, что в этом графе нет циклов. Допустим, что циклы есть. Возьмем любое ребро, принадлежащее какому-нибудь циклу, и удалим из графа. Граф останется связным. Но в новом графе

n вершин и $n-2$ ребра, по теореме 5.4 этот граф не может быть связным. Противоречие. Значит, в графе G нет циклов и этот граф – дерево. ■

Теорема 5.10. *В любом дереве для любых двух вершин существует единственный путь, соединяющий эти вершины.*

Доказательство. Так как дерево – связный граф, то путь между любыми двумя вершинами существует. Предположим, что в некотором дереве есть два различных пути, P_1 и P_2, соединяющих вершины a и b. Начальные отрезки этих путей совпадают (оба пути начинаются в одной вершине a). Пусть x – последняя вершина этого совпадающего начального участка, а далее в пути P_1 следует вершина y_1, в пути P_2 – вершина y_2. Рассмотрим ребро (x, y_1). Если удалить его из графа, он останется связным, так как в нем останется путь, соединяющий вершины x и y_1 – следуем вдоль пути P_1 от вершины y_1 до первой вершины, принадлежащей пути P_2 (такая существует, так как оба пути заканчиваются в одной вершине b), а затем вдоль пути P_2 в обратном направлении до вершины x. Значит, ребро (x, y_1) не является перешейком. По теореме 5.5 оно принадлежит какому-нибудь циклу. Следовательно, в графе есть цикл. Противоречие, так как наш граф – дерево. ■

Вершина степени 1 в дереве называется *листом*. Отметим два факта, связанных с листьями.

Утверждение 5.1. *В дереве с $n > 1$ для любой вершины любая наиболее удаленная от нее вершина является листом.*

Действительно, пусть a – вершина дерева, b – одна из наиболее удаленных от нее вершин. Допустим, b не лист, тогда имеется вершина c, смежная с b и не принадлежащая единственному пути между a и b. Но тогда $d(a,c) = d(a,b) + 1$, а это противоречит тому, что b – наиболее удаленная от a вершина.

Утверждение 5.2. *В любом дереве с $n > 1$ имеется не менее двух листьев.*

В самом деле, пусть a – вершина дерева, b – наиболее удаленная от нее вершина. Как доказано выше, b – лист. Пусть c – вершина, наиболее удаленная от b. Тогда c – тоже лист.

При любом $n > 1$ существует дерево, имеющее ровно два листа – это граф P_n.

В графе может быть сколько угодно центральных вершин. Есть графы, у которых все вершины центральные, например, K_n или C_n при любом n. Для деревьев имеется гораздо более узкий диапазон возможностей.

Теорема 5.11 (о центре дерева). *Центр любого дерева состоит из одной вершины или из двух смежных вершин.*

Доказательство. Для деревьев с одной и двумя вершинами это, очевидно, верно. Далее проводим индукцию по числу вершин.

Пусть T – дерево с вершинами, $n \geq 3$. Если a – лист, b – вершина, смежная с a, то b не является листом (иначе было бы $n = 2$) и все пути, соединяющие вершину a с другими вершинами, проходят через вершину b. Поэтому для любой вершины x, отличной от a, выполняется равенство $d(b,x) = d(a,x) - 1$. Отсюда следует, что $ecc(b) = ecc(a) - 1$. Значит, ни один лист не является центральной вершиной, так как смежная с ним вершина имеет меньший эксцентриситет.

Удалим из дерева T все листья. Так как при удалении листа связность не нарушается, то в результате получится дерево T'. Ни одна центральная вершина дерева T не является листом, поэтому все они сохранятся в T'. Из утверждения 5.1 следует, что при одновременном удалении всех листьев эксцентриситет каждой из оставшихся вершин уменьшается на 1. Значит, вершины с наименьшим эксцентриситетом в дереве T останутся вершинами с наименьшим эксцентриситетом и в дереве T', поэтому $C(T) = C(T')$. По предположению индукции $C(T')$ состоит из одной вершины или из двух смежных вершин. ■

Таким образом, все деревья делятся на два класса: деревья с одной центральной вершиной и деревья с двумя центральными вершинами. Оба класса непустые – граф P_n при нечетном n имеет одну центральную вершину, а при четном две.

Любой способ представления графов, конечно, годится и для представления деревьев. Но есть и специфические способы представления именно деревьев. Самым компактным представлением помеченного дерева является его *код Прюфера.*

Код Прюфера строится для дерева с множеством вершин $\{1,2,\dots,n\}$. Строится он следующим образом. В дереве T находим наименьший лист a_1 и смежную с ним вершину b_1. Вершину b_1 запоминаем, а вершину a_1 удаляем из дерева. В полученном дереве T_1 снова отыскиваем наименьший лист a_2 и смежную с ним вершину b_2. Запоминаем b_2, удаляем a_2, получается дерево T_2. Такие действия повторяем до тех пор, пока не дойдем до дерева T_{n-2}, состоящего из двух вершин. К этому моменту мы запомнили вершины $b_1, b_2, \dots, b_{n-2}$. Этот набор чисел и есть код Прюфера дерева T, он обозначается $p(T)$:

$$p(T) = (b_1, b_2, \dots, b_{n-2}).$$

Отметим, что каждый элемент кода Прюфера – это целое число в диапазоне от 1 до n.

На рисунке 5.12 показан пример, иллюстрирующий процесс построения кода Прюфера.

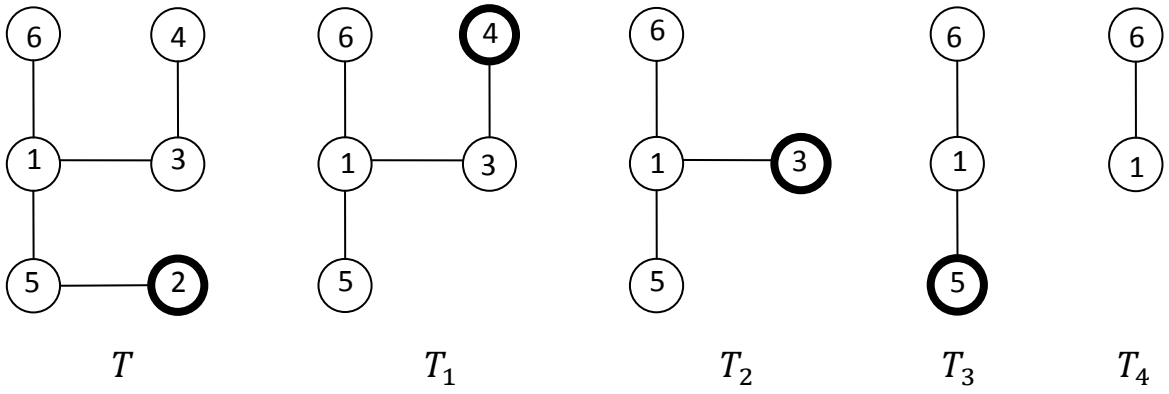


Рис.12. Построение кода Прюфера. Результат: $p(T) = (5,3,1,1)$.

По коду Прюфера можно однозначно восстановить дерево. Число вершин узнать легко – нужно к числу элементов кода прибавить 2. Нетрудно узнать и степени вершин.

Лемма. *Степень вершины в дереве T на 1 больше числа вхождений этой вершины в $p(T)$.*

Доказательство. Пусть a – вершина степени k в дереве T, покажем, что a входит в $p(T)$ ровно $k-1$ раз. В процессе построения кода вершина a либо будет удалена из дерева, либо останется в числе двух последних вершин. В любом случае она в какой-то момент превратится в лист. Для того, чтобы она превратилась в лист, должны быть удалены $k-1$ из смежных с ней вершин. Каждый раз, когда удаляется вершина, смежная с вершиной a, сама вершина a добавляется к коду. Значит, она будет добавлена к коду $k-1$ раз, после этого станет листом и больше добавляться к коду не будет. ■

Для того, чтобы восстановить дерево, нужно узнать его ребра (вершины нам известны). Всего в дереве $n-1$ ребро. Из них $n-2$ ребра – это пары (a_i, b_i), $i = 1,2,\dots,n-2$, где $a_1,\dots,a_{n-2}$ – это листья, последовательно удаляемые при построении кода Прюфера, $b_1,\dots,b_{n-2}$ – смежные с ними вершины. Последние известны из кода Прюфера. Остается найти $a_1,\dots,a_{n-2}$. Кроме того, есть еще ребро, принадлежащее дереву T_{n-2}, на котором завершается процесс построения кода. Значит, нужно еще найти две вершины этого ребра. Продемонстрируем на примере, как можно решить эти задачи.

Пусть задан код Прюфера $p(T) = (5,1,4,5,4)$. Процесс восстановления дерева отражен в таблице:

i	1	2	3	4	5	6	7	a_i	b_i
1	2	1	1	3	3	1	1	2	5
2	2	0	1	3	2	1	1	3	1
3	1	0	0	3	2	1	1	1	4
4	0	0	0	2	2	1	1	6	5
5	0	0	0	2	1	0	1	5	4
6	0	0	0	1	0	0	1	4	7

Код состоит из пяти чисел, следовательно, дерево имеет 7 вершин. Их номера выписаны в верхней строке таблицы. Следующая строка содержит степени вершин, вычисленные по коду с помощью леммы. При кодировании число a_1 определялось как наименьший лист в дереве. Листья – это вершины степени 1. Из таблицы видно, что наименьшим листом является вершина 2. Значит, $a_1 = 2$. Вписываем это значение в соответствующую клетку таблицы, а в соседнюю клетку – число $b_1 = 5$ из кода. Это дает ребро (2,5). Далее в процессе кодирования удаляется вершина 2, при этом степень вершины 5 уменьшается на 1. Это отражено в следующей строке таблицы, представляющей степени вершин дерева T_1. Удаленные вершины в таблице отмечаются нулем. Далее процесс повторяется, каждая следующая строка представляет набор степеней вершин дерева, полученного после удаления очередного листа. В последней строке остались две единицы, они указывают на две оставшиеся вершины, образующие последнее ребро. Вписываем эти вершины в два последних столбца. Теперь два последних столбца таблицы представляют все ребра дерева. Само дерево изображено на рисунке 5.13.

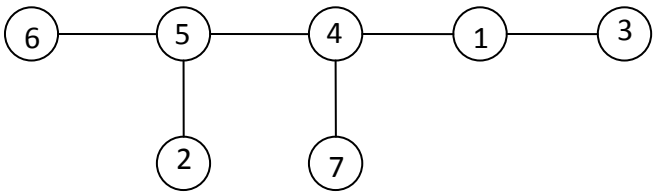


Рис. 5.13. Дерево с кодом Прюфера $p(T) = (5,1,4,5,4)$

Дадим общее описание алгоритмов построения кода Прюфера и его декодирования.

Алгоритм построения кода Прюфера.

Вход: дерево T с множеством вершин $\{1,2,\dots,n\}$.

Выход: набор чисел $p = (b_1, b_2, \dots, b_{n-2})$.

Процедура:

1 для i от 1 до $n-2$ выполнить

2 найти в дереве T наименьший лист a;
3 найти вершину b, смежную с a;
4 удалить a из T;
5 положить $b_i = b$.

Алгоритм восстановления дерева по коду Прюфера.

Вход: набор $p = (b_1, b_2, \dots, b_{n-2})$ чисел из множества $\{1,2, \dots, n\}$.

Выход: множество E, состоящее из пар элементов множества $\{1,2, \dots, n\}$.

Процедура:

1 для i от 1 до n положить $\deg(i) = 1$;
2 для i от 1 до $n-2$ положить $\deg(b_i) = \deg(b_i) + 1$;
3 для i от 1 до $n-2$ выполнить
4 найти наименьшее a с $\deg(a) = 1$;
5 добавить к E пару (a, b_i) ;
6 положить $\deg(a) = 0$;
7 положить $\deg(b_i) = \deg(b_i) - 1$
8 найти a и b с $\deg(a) = \deg(b) = 1$ и добавить к множеству E пару (a, b).

Тот факт, что по коду Прюфера можно восстановить дерево, еще не означает, что этот код устанавливает взаимно однозначное соответствие между множеством всех деревьев с множеством вершин $\{1,2, \dots, n\}$ и множеством наборов длины $n-2$ целых чисел из множества $\{1,2, \dots, n\}$, т.е. множеством $\{1,2, \dots, n\}^{n-2}$. Этот факт означает инъективность отображения деревьев в наборы, но нужно еще доказать, что оно сюръективно, т.е., что каждый такой числовой набор является кодом некоторого дерева.

Оно действительно сюръективно, в этом можно убедиться следующим образом. Алгоритм восстановления дерева можно применить к любому набору из множества $\{1,2, \dots, n\}^{n-2}$. В любом случае результатом будет множество пар E, которое можно рассматривать как множество ребер некоторого графа. Но всегда ли этот граф будет деревом? В этом графе $n-1$ ребро. Если мы докажем, что в нем нет циклов, то из теоремы 5.9 будет следовать, что он является деревом.

Пусть $(a_1, b_1), (a_2, b_2), \dots, (a_{n-1}, b_{n-1})$ – пары, последовательно добавляемые к множеству E в процессе работы алгоритма декодирования. Обозначим через G_i граф с множеством вершин $\{1,2, \dots, n\}$ и множеством ребер $\{(a_i, b_i), (a_{i+1}, b_{i+1}), \dots, (a_{n-1}, b_{n-1})\}$. Граф G_i получается из графа G_{i+1} добавлением ребра (a_i, b_i). Когда это ребро добавляется к множеству E в процессе работы алгоритма, значение $\deg(a_i)$ изменяется с 1 на 0. Значит, в графе G_i степень вершины a_i равна 1, а в графе G_{i+1} она равна 0 (нетрудно

доказать, что в начале i –того повторения основного цикла массив deg содержит степени вершин графа G_i). Цикл не может проходить через вершину степени 1. Следовательно, если в графе G_{i+1} циклов нет, то их нет и в графе G_i. В графе G_{n-1} циклов нет. Проводя «обратную индукцию» по i, убеждаемся, что в графе G_1 , который является результатом работы алгоритма, циклов нет.

Таким образом, функция $P(T)$ является биекцией множества деревьев с множеством вершин $\{1,2,\dots,n\}$ на множество наборов $\{1,2,\dots,n\}^{n-2}$. Число наборов нам известно, по правилу равенства таково же и число деревьев. Получаем следующий результат, известный как *формула Кэли*.

Теорема 5.12 (формула Кэли). *Число деревьев с множеством вершин* $\{1,2,\dots,n\}$ *равно* n^{n-2}.

Корневым деревом называется дерево с особо выделенной вершиной, называемой *корнем*.

Деревья, встречающиеся в приложениях – это чаще всего именно корневые деревья. Схема подчиненности в организации, структура папок и файлов на компьютерном диске, различные системы классификации – примеры корневых деревьев. Корневые деревья лежат в основе многих эффективных структур хранения, поиска, сортировки данных.

Если в корневом дереве вершина b лежит на пути, соединяющем вершину a с корнем, то вершину b называют *предком* вершины a, а вершину a – *потомком* вершины b. Если при этом a и b смежны, то говорят, что вершина b – *отец* вершины a, а вершина a – *сын* вершины b (иногда используют термины «родитель» и «ребенок»). Пример корневого дерева показан на рисунке 5.14, корень расположен сверху. В этом дереве вершина 4 является предком вершин 5, 6, 7 и отцом вершины 5, у вершины 3 три сына, у вершины 5 – два, у вершины 8 сыновей нет.

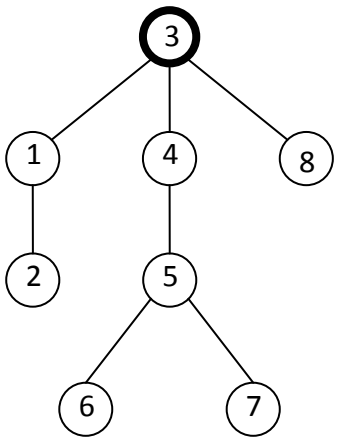


Рис. 5.14. Корневое дерево

В корневом дереве у каждой вершины, кроме корня, имеется единственный отец. Это дает компактный и удобный во многих применениях способ задания корневых деревьев с помощью таблицы, указывающей для каждой вершины ее отца. Иногда считают, что отцом корня является сама эта вершина. Для дерева на рисунке 5.14 получается такая таблица:

Вершина	1	2	3	4	5	6	7	8
Отец	3	1	3	3	4	5	5	3

В помеченном дереве с n вершинами корень можно выбрать n способами, поэтому число помеченных корневых деревьев в n раз больше числа помеченных деревьев, то есть равно n^{n-1}.

Каркас (применяются также термины *остов*, *остовное дерево*) связного графа – это дерево, являющееся подграфом этого графа и содержащего все его вершины. На рисунке 5.15 показан граф и его каркасы.

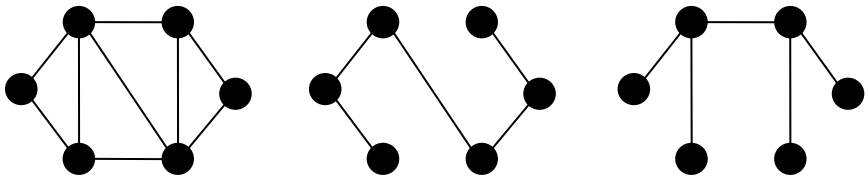


Рис. 5.15. Граф (слева) и два его каркаса

У всякого связного графа имеется хотя бы один каркас. Действительно, если в связном графе нет циклов, то этот граф – дерево и он сам является своим каркасом. Если же циклы есть, то удалив цикловое ребро (ребро, принадлежащее какому-нибудь циклу), получим связный подграф. Продолжая удалять цикловые ребра, в конце концов придем к связному подграфу, не имеющему циклов, то есть дереву. Оно и будет каркасом исходного графа.

Если в графе есть циклы, то у него больше одного каркаса. Определить точное число каркасов связного графа позволяет так называемая матричная теорема Кирхгофа. Приведем ее без доказательства. Для графа G с множеством вершин $\{1,2,\dots,n\}$ определим матрицу Кирхгофа $K(G)$ – квадратную матрицу порядка n, элементы которой определяются следующим образом:

$$K_{ij} = \begin{cases} -1, \text{если вершины } i \text{ и } j \text{ смежны,} \\ 0, \text{если вершины } i \text{ и } j \text{ не смежны и } i \neq j, \\ \deg(i), \text{ если } i = j. \end{cases}$$

Иначе говоря, $K(G)$ получается из матрицы смежности, если заменить все 1 на -1, а вместо нулей на главной диагонали поставить степени вершин. Заметим, что матрица $K(G)$ – вырожденная, так как сумма элементов каждой строки равна 0, то есть столбцы линейно зависимы.

Теорема 5.13 (матричная теорема Кирхгофа). *Если G – связный граф с не менее чем двумя вершинами, то алгебраические дополнения всех элементов матрицы $K(G)$ равны между собой и равны числу каркасов графа G.*

5.7. Двудольные графы

Граф называется *двудольным***,** если множество его вершин можно так разбить на две части, что каждое ребро соединяет вершины из разных частей. Эти подмножества называются *долями*. На рисунке 5.16 показан пример двудольного графа.

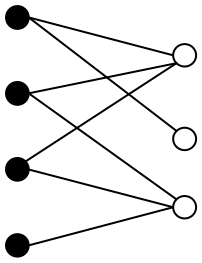


Рис. 5.16. Двудольный граф

Двудольные графы часто встречаются в приложениях – с их помощью моделируются отношения между объектами двух типов. Таковы, например, отношения «продукт x используется в приготовлении блюда y» или «работник x владеет профессией y».

Путь P_n при любом n является двудольным графом: одна доля состоит из вершин с четными номерами, другая – с нечетными. Цикл C_3 – пример графа, не являющегося двудольным: при любом разбиении множества его вершин на два подмножества в одном из этих подмножеств окажутся две смежных вершины. Цикл C_4 – двудольный граф – это показано на рисунке 5.17.

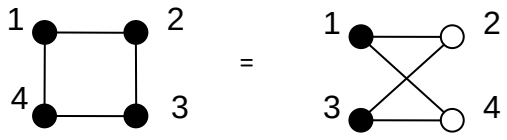


Рис. 5.17. C_4 как двудольный граф

Рассмотрим цикл C_5 и попытаемся распределить его вершины по двум долям. Поместим вершину 1 в первую долю. Тогда вершина 2 должна быть во второй доле, так как она смежна с вершиной 1. Вершина 3, смежная с вершиной 2, должна быть в первой доле, вершина 4, смежная с вершиной 3 –

во второй, а вершина 5, смежная с вершиной 4 – в первой. Но теперь ребро (1,5) соединяет вершины одной доли (рисунок 5.18)., чего быть не должно, если граф двудольный. Процесс распределения вершин по долям происходит однозначно, и вершины 1 и 5 неизбежно оказываются в одной доле. Следовательно, граф C_5 не двудольный.

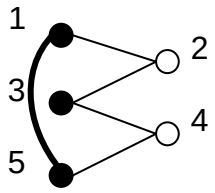


Рис. 5.18. Граф C_5 не является двудольным

Эти рассуждения можно распространить на циклы с любым числом вершин и становится ясно, что цикл C_n – двудольный граф при четном n и не двудольный при нечетном (рисунок 5.19).

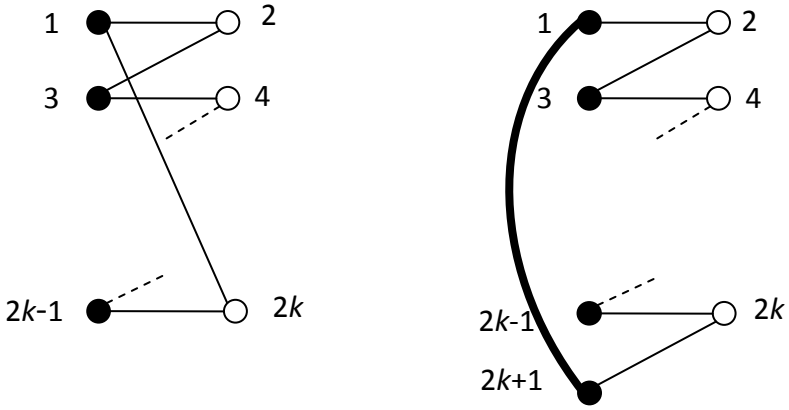


Рис. 5.19. Четный и нечетный циклы

Оказывается, присутствие в графе нечетных циклов (т.е. циклов нечетной длины) – единственная причина, по которой граф не является двудольным.

Теорема 5.14 (теорема Кёнига). *Граф является двудольным тогда и только тогда, когда он не содержит нечетных циклов в качестве подграфов.*

Доказательство. Мы уже видели, что это условие необходимо – нечетный цикл не является двудольным графом, следовательно, любой граф, содержащий такой цикл, – не двудольный. Докажем достаточность.

Очевидно, что граф двудольный тогда и только тогда, когда каждая его компонента связности – двудольный граф. Поэтому достаточно рассмотреть связные графы.

Пусть G – связный граф, не имеющий нечетных циклов. Возьмем какую-нибудь вершину a в этом графе и разделим все вершины графа на два подмножества: V_1 – множество всех вершин, расстояние от которых до вершины a нечетно, V_2 – множество вершин с четным расстоянием до a. Покажем, что нет ребер, соединяющих вершины из одного подмножества. Отсюда будет следовать, что G – двудольный граф с долями V_1 и V_2.

Допустим, что (b, c) – такое ребро. Расстояния от вершины a до вершин b и c могут быть различными или одинаковыми. Рассмотрим обе возможности.

1) $d(a,b) \neq d(a,c)$. Пусть $d(a,b) < d(a,c)$. Если $x_1, x_2, \dots, x_k$ – кратчайший путь от a до b, то, очевидно, $x_1, x_2, \dots, x_k, c$ – кратчайший путь от a до c. Значит, $d(a,c) = d(a,b) + 1$. Но это противоречит тому, что числа $d(a,b)$ и $d(a,c)$ имеют одинаковую четность (вершины b и c принадлежат одному подмножеству).

2) $d(a,b) = d(a,c)$. Пусть $x_1, x_2, \dots, x_k$ и $y_1, y_2, \dots, y_k$ – кратчайшие пути от вершины a до вершин b и c, то есть $x_1 = y_1 = a$, $x_k = b$, $y_k = c$. Эти два пути начинаются в одной вершине, а заканчиваются в разных. Пусть $x_i = y_i$ – последняя вершина, принадлежащая обоим путям, то есть $x_j \neq y_j$ при $j > i$ (рисунок 5.20). Но тогда $x_i, x_{i+1}, \dots, x_k, y_k, y_{k-1}, \dots, y_i$ – цикл длины $2(k-i)+1$, чего не может быть, так как в графе нет нечетных циклов. ■

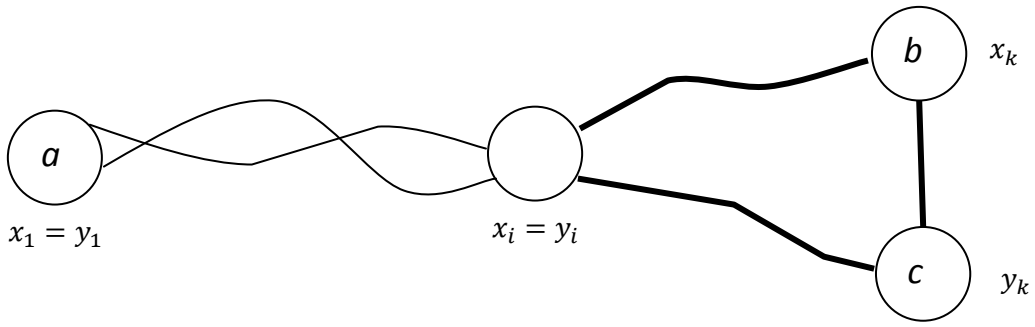


Рис. 5.20. К доказательству теоремы 5.14

Двудольный граф называется *полным двудольным графом*, если в нем каждая вершина одной доли смежна с каждой вершиной другой доли. Полный двудольный граф, в котором одна доля состоит из p вершин, а другая – из q вершин, обозначается $K_{p,q}$. На рисунке 5.21 показан граф $K_{2,3}$.

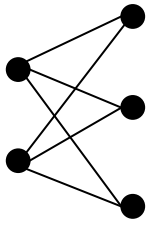
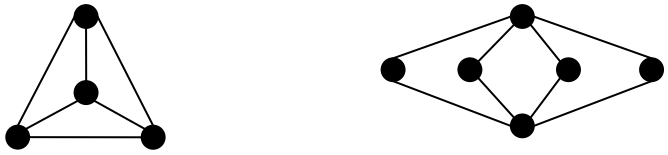


Рис. 5.21. Граф $K_{2,3}$

5.8. Планарные графы

Мы уже не раз пользовались возможностью изобразить граф в виде фигуры на плоскости. Некоторые графы можно нарисовать так, чтобы линии, представляющие ребра, не пересекались. Такое изображение называют *плоским графом*. Более точно, плоский граф – это граф, вершинами которого являются точки плоскости, а ребра представлены линиями, соединяющими смежные вершины, при этом никакие два ребра не должны иметь общих точек, кроме инцидентной им обоим вершины. Граф называется *планарным***,** если он изоморфен плоскому графу. Этот плоский граф называют также *плоской укладкой* планарного графа. На рисунке 5.22 показаны плоские укладки графов K_4 и $K_{2,4}$. А вот у графов K_5 и $K_{3,3}$ плоских укладок не существует – вскоре будет доказано, что эти графы не планарны.



K_4 $K_{2,4}$

Рис. 5.22. Плоские укладки двух графов

Если плоскость разрезать по ребрам плоского графа, она распадется на связные части, которые называют *гранями*. Всегда имеется одна неограниченная *внешняя* грань, все остальные грани называются *внутренними*. Если в плоском графе нет циклов, то у него имеется только одна грань. Если же циклы есть, то граница каждой грани содержит цикл, но не обязательно является циклом. На рисунке 5.23 показан плоский граф с пятью пронумерованными гранями. Граница грани с номером 2 состоит из двух циклов, а граница грани с номером 3 кроме цикла длины 5 включает еще дерево из трех ребер.

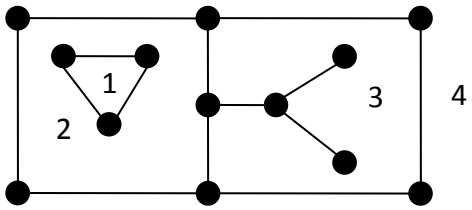


Рис. 5.23. Плоский граф и его грани

Следующая теорема дает формулу для числа граней, называемую *формулой Эйлера.*

Теорема 5.15 (формула Эйлера). Количество граней в любой плоской укладке планарного графа, имеющего n вершин, m ребер и k компонент связности, равно $m - n + k + 1$.

Доказательство. Докажем сначала утверждение теоремы при $k = 1$. Проведем индукцию по числу ребер.

Согласно теоремам 5.4 и 5.9 наименьшее число ребер в связном графе с n *вершинами равно* $n - 1$ и всякий связный граф с таким числом ребер является деревом. Очевидно, любое дерево – планарный граф, в любой плоской укладке которого имеется единственная грань, и формула верна: $n - 1 - n + 1 + 1 = 1$.

Допустим, в связном плоском графе $m > n - 1$. Тогда по теореме 5.3 в графе есть цикл. Возьмем какое-нибудь ребро e, принадлежащее некоторому простому циклу. Это ребро содержится в границе двух граней, одна из которых целиком лежит внутри цикла, другая – снаружи. Если удалить ребро e из графа, то эти две грани сольются в одну. Граф, полученный удалением ребра e, очевидно, будет плоским и связным, в нем на одно ребро и на одну грань меньше, чем в исходном графе, а число вершин осталось прежним. По предположению индукции в новом графе имеется ровно $m - 1 - n + 2 = m - n + 1$ грань, значит, в исходном было $m - n + 2$ грани. Таким образом, формула верна для любого связного плоского графа.

Если граф несвязен, то в компоненте связности, имеющей n_i вершин и m_i ребер, как доказано выше, будет $m_i - n_i + 1$ внутренняя грань. Суммируя по всем компонентам и прибавляя 1 для учета внешней грани, убеждаемся в справедливости формулы в общем случае. ■

Следствие 1. *Если в планарном графе* n *вершин,* $n \geq 3$, *и* m *ребер, то* $m \leq 3n - 6$.

Доказательство. Если в графе нет циклов, то $m \leq n - 1$ и неравенство выполняется при $n \geq 3$. Рассмотрим плоский граф с f гранями, в котором имеются циклы. Занумеруем грани числами от 1 до f и обозначим через a_i количество ребер, принадлежащих границе грани с номером i. Так как граница каждой грани содержит цикл, то $a_i \geq 3$ для каждого i, следовательно, $\sum_{i=1}^{f} a_i \geq 3f$. С другой стороны, каждое ребро принадлежит границе не более чем двух граней, поэтому $\sum_{i=1}^{f} a_i \leq 2m$. Из этих двух неравенств следует, что $3f \leq 2m$. Применяя формулу Эйлера, получаем

$m \leq 3n - 3k - 3 \leq 3n - 6$. ■

Следствие 1 дает необходимое условие планарности, которое в некоторых случаях позволяет установить, что граф не является планарным. Рассмотрим, например, полный граф K_5. У него $n = 5$, $m = 10$ и неравенство из следствия 1 не выполняется. Значит, этот граф не планарный.

В то же время существуют графы, не являющиеся планарными, для которых неравенство следствия 1 выполняется. Пример – полный двудольный граф $K_{3,3}$. У него 6 вершин и 9 ребер. Неравенство выполняется, но мы сейчас установим, что он не планарный. Заметим, что в этом графе нет циклов длины 3 (так как он двудольный, в нем вообще нет нечетных циклов). Поэтому граница каждой грани содержит не менее четырех ребер. Повторяя рассуждения из доказательства следствия 1, но используя неравенство $a_i \geq 4$ вместо $a_i \geq 3$ получаем следующий результат.

Следствие 2. *Если в планарном графе* n *вершин,* $n \geq 3$, m *ребер и нет циклов длины* 3, *то* $m \leq 2n - 4$.

Для графа $K_{3,3}$ неравенство следствия 2 не выполняется, и это доказывает, что он непланарен.

Известно несколько критериев планарности, сформулируем без доказательства два из них.

Операция *подразбиения ребра* (a, b) состоит в том, что это ребро удаляется из графа, добавляются новая вершина c и ребра (a, c), (b, c). Это можно трактовать как замену ребра путем длины 2. Многократное применение операции подразбиения равносильно замене некоторых ребер графа путями произвольной длины. Полученный в результате граф назовем *подразбиением* исходного графа. На рисунке 5.24 изображено подразбиение графа K_4.

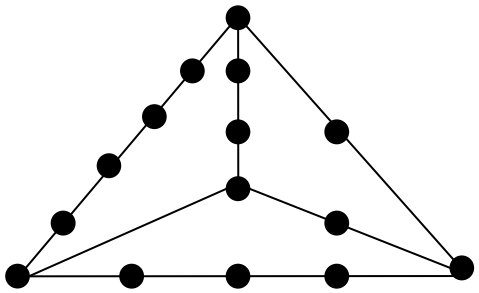


Рис. 5.24. Подразбиение графа K_4

Теорема 5.16 (критерий Понтрягина–Куратовского). *Граф планарен тогда и только тогда, когда у него нет подграфа, являющегося подразбиением графа* K_5 *или* $K_{3,3}$.

Операция *стягивания* ребра (a, b) определяется следующим образом. Вершины a и b удаляются из графа, к нему добавляется новая вершина c и она соединяется ребром с каждой вершиной, с которой была смежна хотя бы одна из вершин a, b (рисунок 5.25). Граф G называется *стягиваемым* к графу H, если H можно получить из G последовательностью стягиваний ребер.

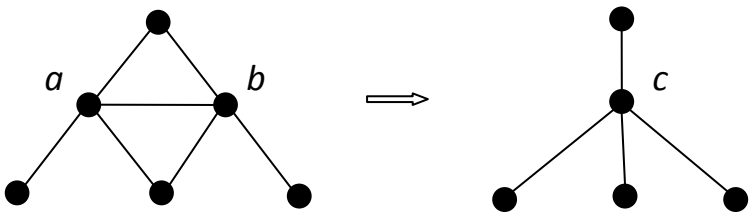


Рис. 5.25. Стягивание ребра

Теорема 5.17 (критерий Вагнера). *Граф планарен тогда и только тогда, когда у него нет подграфов, стягиваемых к графу* K_5 *или* $K_{3,3}$..

В качестве примера рассмотрим граф, изображенный на рисунке 5.26, его называют графом Петерсена. В нем нет подграфа, являющегося подразбиением графа K_5. К такому выводу легко придти, если заметить, что в графе K_5 каждая вершина имеет степень 4. Значит, и в любом его подразбиении будет 5 вершин степени 4, а в графе Петерсена степень каждой вершины равна 3. В то же время легко видеть, что граф Петерсена можно превратить в K_5 стягиванием пяти выделенных на рисунке ребер.

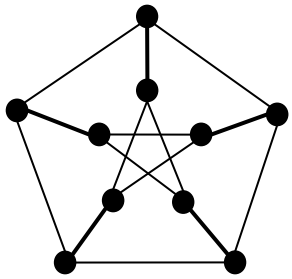


Рис. 5.26. Граф Петерсена

Глава 6. Логические функции. Алгебра логики

6.1. Булевы функции. Существенные и фиктивные переменные

Функция $f(x_1, x_2, \dots, x_n)$, у которой каждая переменная принимает значения из множества $\{0,1\}$ и сама функция принимает значения в этом множестве, называется *логической функцией* или *булевой функцией*. Областью определения такой функции является множество $\{0,1\}^n$, а областью значений – множество $\{0,1\}$, т.е. $f: \{0,1\}^n \to \{0,1\}$.

Логическую функцию можно задать таблицей. В такой таблице для каждого набора значений переменных указывается значение функции на этом наборе. Наборы значений переменных обычно перечисляют в лексикографическом порядке. Пример таблицы, задающей функцию трех переменных:

x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Для функции от n переменных в таблице будет 2^n строк (не считая строки заголовка, где указаны имена переменных и имя функции). Таблицы разных функций от одних и тех же переменных отличаются только последним столбцом – столбцом значений функции. Выписывая этот столбец в виде горизонтальной строки, получаем *набор значений* или *вектор значений* функции. Набор значений функции f обозначается через $\tilde{f}$. Для функции, заданной выше таблицей, $\tilde{f} = 10101100$. Функция определяется набором значений и каждый набор длины 2^n из элементов 0, 1, задает некоторую функцию от n переменных. Следовательно, булевых функций от

n переменных имеется ровно столько, сколько существует таких наборов, т.е. 2^{2^n}.

Теорема 6.1. *Существует ровно* 2^{2^n} *булевых функций от* n *переменных.*

Название «логические функции» подразумевает, что такие двузначные функции имеют какое-то отношение к логике. В классической логике рассматриваются высказывания – утверждения, которые могут быть истинными или ложными. Если слова «истина» и «ложь» заменить символами 1 и 0, то логическая функция может описывать зависимость истинности или ложности некоторого высказывания от истинности или ложности других высказываний. Это объясняет происхождение термина, на самом деле логические функции могут использоваться в математических моделях везде, где какие-нибудь параметры принимают лишь два возможных значения, а такое встречается часто.

Рассмотрим функцию, заданную таблицей:

x_1	x_2	x_3	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Можно заметить, что на любых двух наборах, отличающихся только значениями переменной x_3, значения этой функции совпадают:

$$f(0,0,0) = f(0,0,1) = 1,$$

$$f(0,1,0) = f(0,1,1) = 0,$$

$$f(1,0,0) = f(1,0,1) = 1,$$

$$f(1,1,0) = f(1,1,1) = 1.$$

Это говорит о том, что данная функция по существу не зависит от переменной x_3. Такую переменную называют фиктивной. Сформулируем общее определение фиктивной переменной.

Переменная x_k функции $f(x_1, x_2, \dots, x_n)$, $1 \le k \le n$, называется *фиктивной*, если при любых значениях остальных переменных выполняется равенство $f(x_1, \dots, x_{k-1}, 0, x_{k+1}, \dots, x_k) = f(x_1, \dots, x_{k-1}, 1, x_{k+1}, \dots, x_k)$.

Переменная, не являющаяся фиктивной, называется *существенной*. Таким образом, переменная x_k существенная, если существуют такие значения $a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_n$ остальных переменных, что значения $f(a_1, \dots, a_{k-1}, 0, a_{k+1}, \dots, a_n)$ и $f(a_1, \dots, a_{k-1}, 1, a_{k+1}, \dots, a_n)$ различны.

В рассмотренном примере переменная x_1 существенная, так как
$f(0,1,0) \neq f(1,1,0)$.
Переменная x_2 тоже существенная, так как
$f(0,0,0) \neq f(0,1,0)$.

Если у некоторой функции имеется фиктивная переменная, то ее можно превратить в функцию от меньшего числа переменных, удалив фиктивную переменную. В табличном представлении функции операция удаления фиктивной переменной состоит в следующем. Из таблицы удаляются все строки, в которых эта переменная принимает значение 1 и удаляется столбец, соответствующий этой переменной.

Можно выполнить и обратную операцию – ввести фиктивную переменную.

В рассмотренном примере после удаления фиктивной переменной x_3 получаем функцию, задаваемую таблицей

x_1	x_2	f'
0	0	1
0	1	0
1	0	1
1	1	1

Строго говоря, f и f' – разные функции, у них разные области определения ($\{0,1\}^3$ и $\{0,1\}^2$). Но они, конечно, тесно связаны друг с другом – каждая из них однозначно определяет другую. Если из двух функций в результате удаления всех фиктивных переменных получаются одинаковые функции, то исходные функции называются *эквивалентными*.

6.2. Элементарные функции. Формулы. Алгебра логики

Некоторые часто встречающиеся логические функции называют *элементарными*. Прежде всего к ним относятся все четыре функции одной переменной. Они показаны в таблице:

x	0	1	x	$\bar{x}$
0	0	1	0	1
1	0	1	1	0

Первые два столбца этой таблицы представляют *константы* – константу 0 и константу 1. У этих функций единственная переменная является фиктивной. Следующий столбец таблицы представляет функцию x, она называется *тождественной функцией.* В последнем столбце представлена функция, называемая *отрицанием*. Обозначение $\bar{x}$ читается «не x». Иногда отрицание обозначается как $\neg x$. В логике речь идет об отрицании некоторого высказывания. Отрицание высказывания А – это утверждение, что А ложно. Высказывание «не А» истинно, если высказывание А ложно и ложно, если А истинно. Это и отражено в таблице значений данной функции.

Рассмотрим теперь наиболее важные функции двух переменных.

Конъюнкция обозначается символом & (он называется *амперсанд*, иногда вместо него используют символ ∧) и задается таблицей

x_1	x_2	$x_1 \& x_2$
0	0	0
0	1	0
1	0	0
1	1	1

Запись $x_1 \& x_2$ читается как «x_1 и x_2». В логической интерпретации речь идет о составном высказывании, полученном соединением двух высказываний с помощью союза «и». Это высказывание истинно тогда и только тогда, когда истинны оба составляющих его высказывания. Например, «$3 > 2$ и $2 \times 2 = 4$» – истинное высказывание, а «$3 < 2$ и $2 \times 2 = 4$» – ложное.

Если рассматривать 0 и 1 как обычные числа, то конъюнкция – это просто умножение. Поэтому ее также называют *логическим умножением*, в

качестве знака операции иногда используют точку, а чаще этот знак вовсе опускают, т.е. пишут x_1x_2 вместо $x_1 \& x_2$.

Дизъюнкция обозначается знаком ∨ и задается таблицей

x_1	x_2	$x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

Запись $x_1 \vee x_2$ читается как «x_1 или x_2». Логическая интерпретация: составное высказывание, истинное тогда и только тогда, когда истинно хотя бы одно из составляющих его высказываний. Например, оба высказывания «$3 > 2$ или $2 \times 2 = 4$» и «$3 < 2$ или $2 \times 2 = 4$» истинны, а «$3 < 2$ или $2 \times 2 = 5$» ложно. Дизъюнкцию иногда называют логическим сложением.

Сумма по модулю 2 обозначается знаком $\oplus$ и задается таблицей

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

В логике эту операцию называют исключающим «или»: речь идет о составном высказывании, которое истинно тогда и только тогда, когда истинно одно из составляющих высказываний, но не оба.

Импликация обозначается стрелкой $\rightarrow$ и задается таблицей

x_1	x_2	$x_1 \rightarrow x_1$
0	0	1
0	1	1
1	0	0
1	1	1

Запись $x_1 \to x_2$ читается «если x_1, то x_2» или «из x_1 следует x_2». В логической интерпретации импликация «из А следует В» – это высказывание, утверждающее, что если истинно высказывание А, то истинно и высказывание В. При этом не предполагается какой-либо смысловой связи между высказываниями А и В. Единственный случай, когда импликация ложна – это когда А истинно, а В ложно (из истинности А не следует истинность В). Во всех остальных случаях импликация считается истинной. В соответствии с этим, например, высказывание «если $2 \times 2 = 4$, то $1 > 3$» ложно, а высказывания «если $2 \times 2 = 4$, то $1 < 3$», «если $2 \times 2 = 5$, то $1 > 3$» «если $2 \times 2 = 5$, то $1 < 3$» следует считать истинными.

Эквиваленция обозначается знаком ~ (используются также символы ≡ и ↔) и задается таблицей

x_1	x_2	$x_1 \sim x_2$
0	0	1
0	1	0
1	0	0
1	1	1

Запись $x_1 \sim x_2$ читается «x_1 эквивалентно x_2». В логической интерпретации имеется в виду утверждение, что два данных высказывания либо оба истинны, либо оба ложны. Нетрудно видеть, что эквиваленция является отрицанием суммы по модулю 2: $x_1 \sim x_2 = \overline{x_1 \oplus x_2}$.

Штрих Шеффера. Функция обозначается вертикальной чертой | и задается таблицей

x_1	x_2	$x_1 \mid x_2$
0	0	1
0	1	1
1	0	1
1	1	0

Очевидно, эта функция есть отрицание конъюнкции: $x_1 \mid x_2 = \overline{x_1 \& x_2}$.

Стрелка Пирса. Обозначается стрелкой ↓ и задается таблицей

x_1	x_2	$x_1 \downarrow x_2$
0	0	1
0	1	0
1	0	0
1	1	0

Очевидно, эта функция – отрицание дизъюнкции: $x_1 \downarrow x_2 = \overline{x_1 \vee x_2}$.

Описанные выше элементарные функции можно рассматривать как операции над элементами из множества $\{0,1\}$. Изучение свойств таких операций и формул, которые можно составлять на их основе, составляет предмет *алгебры логики*. С помощью формул можно представлять логические функции и такое представление иногда привлекательнее табличного, оно более гибкое и нередко более компактное.

В формулах алгебры логики используются обозначения переменных (латинские буквы с индексами или без них), символы констант 0 и 1, обозначение операции отрицания (надстрочная черта или знак ¬), знаки других операций (элементарных функций) &, ∨ и т. д., скобки для указания порядка действий. Можно, написать, например, такую формулу:

$$(\bar{x}\&y) \vee ((y \oplus 1)\&(z \to 0)).$$

Эта формула задает некоторую функцию $f(x,y,z)$. Фактически она описывает алгоритм вычисления значений этой функции при заданных значениях переменных: нужно просто выполнить все действия в порядке, указанном скобками. Например, чтобы вычислить $f(0,0,0)$, нужно последовательно найти $\bar{0} = 1$, $1\&0 = 0$, $0 \oplus 1 = 1$, $0 \to 0 = 1$, $1\&1 = 1$, $f(0,0,0) = 0 \vee 1 = 1$.

Выше уже говорилось, что знак конъюнкции принято опускать (как это делают с обычным арифметическим умножением). В дальнейшем так и будем обычно поступать (иногда во избежание неясностей используем точку). Кроме того, принято считать, что операция конъюнкции имеет более высокий приоритет по отношению к другим бинарным операциям (или, как говорят, «связывает сильнее»). Это означает, что конъюнкция должна выполняться раньше других операций (кроме отрицания), если иной порядок действий не указан скобками. С учетом этих соглашений приведенная выше формула может быть записана в следующем виде:

$$\bar{x}y \vee (y \oplus 1)(z \to 0).$$

Две формулы *эквивалентны*, если они представляют эквивалентные функции. Иначе говоря, формулы A и B эквивалентны, если равенство $A = B$ является тождеством, т.е. справедливо при любых значениях входящих в них переменных. Отметим, что при этом некоторые переменные, входящие в одну из этих формул, могут не входить в другую. Например, равенство

$$x \oplus (0 \to y) = \overline{x(z \to 1)}$$

является тождеством. Переменная y в левой части и переменная z в правой части – фиктивные переменные соответствующих функций, обе формулы эквивалентны формуле $\bar{x}$.

6.3. Булевы формулы

Одним из важнейших видов формул алгебры логики являются формулы, в которых используются только операции отрицания, конъюнкции и дизъюнкции (могут появляться также константы 0 и 1). Такие формулы будем называть *булевыми формулами* (Джордж Буль – английский математик XIX в., один из основателей математической логики).

Перечислим ряд тождеств, выражающих важнейшие свойства трех операций. Они объединены в группы однотипных тождеств.

1. $0 \cdot x = 0; \quad 1 \cdot x = x; \quad 0 \vee x = x; \quad 1 \vee x = 1.$

2. $x \cdot x = x; \quad x \cdot \bar{x} = 0; \quad x \vee x = x; \quad x \vee \bar{x} = 1.$

3. Закон двойного отрицания:

$$\bar{\bar{x}} = x.$$

4. Коммутативность:

$$xy = yx;$$

$$x \vee y = y \vee x.$$

5. Ассоциативность:

$$x(yz) = (xy)z\ ;$$

$$x \vee (y \vee z) = (x \vee y) \vee z.$$

6. Дистрибутивность:

$$x(y \vee z) = xy \vee xz;$$

$$x \vee yz = (x \vee y)(x \vee z).$$

7. Законы де Моргана:

$$\overline{xy} = \bar{x} \vee \bar{y};$$

$$\overline{x \vee y} = \bar{x}\,\bar{y}.$$

Все эти свойства легко проверяются по таблицам.

Благодаря свойству ассоциативности можно писать просто $x_1x_2x_3$, не расставляя скобки, так как порядок действий не влияет на результат. Это относится и к конъюнкции любого числа переменных. В этом случае используется запись

$$x_1x_2 \ldots x_n = \bigwedge_{i=1}^{n} x_i$$

(читается: «конъюнкция x_i по i от 1 до n»).
Аналогично для дизъюнкции:

$$x_1 \vee x_2 \vee \ldots \vee x_n = \bigvee_{i=1}^{n} x_i.$$

Можно сделать два полезных наблюдения:

$x_1x_2 \ldots x_n = 1$ тогда и только тогда, когда $x_1 = x_2 = \cdots = x_n = 1$;

$x_1 \vee x_2 \vee \ldots \vee x_n = 1$ тогда и только тогда, когда хотя бы одна из переменных x_i принимает значение 1.

Законы де Моргана с помощью индукции можно распространить на любое число сомножителей и слагаемых:

$$\overline{\bigwedge_{\imath=1}^{n} x_\imath} = \bigvee_{i=1}^{n} \overline{x_\imath}\ ,$$

$$\overline{\bigvee_{\imath=1}^{n} x_\imath} = \bigwedge_{i=1}^{n} \overline{x_\imath}\ .$$

Отметим, что если в некотором тождестве вместо переменных подставить формулы (вместо всех вхождений одной и той же переменной – одну и ту же формулу), то полученное равенство тоже будет тождеством. Например, к любым формулам A и B можно применить закон де Моргана:

$\overline{AB} = \bar{A} \vee \bar{B}$.

Приведенные тождества могут быть использованы для эквивалентных преобразований формул с целью их упрощения. Например:

$$x(\bar{x} \vee y) \vee \bar{x}y = x \cdot \bar{x} \vee xy \vee \bar{x}y = 0 \vee (x \vee \bar{x})y = 1 \cdot y = y\ .$$

С помощью эквивалентных преобразований можно также выводить новые тождества. Например:

$$x \vee xy = x \cdot 1 \vee xy = x(1 \vee y) = x \cdot 1 = x\ .$$

Это тождество называется законом поглощения, оно весьма полезно для упрощения булевых формул и мы присоединяем его к списку основных тождеств.

8. Закон поглощения:

$$x \vee xy = x\ .$$

Напомним, что функция штрих Шеффера равна отрицанию конъюнкции, а стрелка Пирса – отрицанию дизъюнкции, т.е. для формул $x|y$ и $x \downarrow y$ имеются эквивалентные булевы формулы. Другие элементарные функции также могут быть выражены булевыми формулами:

$$x \oplus y = x\bar{y} \vee \bar{x}y\ ;$$

$$x \sim y = \bar{x}\bar{y} \vee xy\ ;$$

$$x \to y = \bar{x} \vee y\ .$$

Это, конечно, не гарантирует того, что всякая логическая функция может быть выражена булевой формулой, но вскоре мы убедимся, что это действительно так.

6.4. Нормальные формы

Среди булевых формул особенно важную роль играют формулы, в которых нет скобок, а операция отрицания применяется только к отдельным переменным, а не к более сложным выражениям. Такая формула называется *дизъюнктивной нормальной формой*, сокращенно ДНФ. Иначе говоря, ДНФ – это логическая сумма логических произведений переменных и отрицаний переменных. Пример ДНФ:

$$x_1x_2x_3 \vee \overline{x_1}x_4 \vee \overline{x_2}\,x_3\overline{x_4} \vee \overline{x_3}\ .$$

Для более строгого определения введем понятие *простой конъюнкции*. Так называется формула вида $A_1A_2 \ldots A_k$, где каждый сомножитель A_i – это переменная или отрицание переменной, причем каждая переменная входит в произведение не более одного раза (с отрицанием или без него). Например, $\overline{x_1}x_2x_3\overline{x_4}$ и x_2 – простые конъюнкции, а $x_1x_2x_1$, $x_1x_2\overline{x_1}$ и $x_1\overline{x_2x_3}$ таковыми не являются. Дизъюнктивная нормальная форма – это формула вида $K_1 \vee$

$K_2 \vee \ldots \vee K_m$, где каждое слагаемое K_i есть простая конъюнкция. Требуется также, чтобы все эти простые конъюнкции были различны (две простые конъюнкции считаются одинаковыми, если они различаются только порядком сомножителей, т. е. состоят из одних и тех же переменных и каждая из этих переменных либо в обе конъюнкции входит без отрицания, либо в обе с отрицанием). Формулу, состоящую из одного символа 0 (это тоже булева формула) тоже считаем ДНФ.

Конъюнктивная нормальная форма (КНФ) – это формула, имеющая вид логического произведения логических сумм переменных и отрицаний переменных, например:

$$(x_1 \vee x_2 \vee x_3)(\overline{x_1} \vee x_4)(\overline{x_2} \vee x_3 \vee \overline{x_4})\overline{x_3}\ .$$

Точное определение аналогично определению ДНФ – вводится понятие простой дизъюнкции и т. д. Формула, состоящая из одного символа 1, тоже считается КНФ.

Любую булеву формулу можно преобразовать в эквивалентную ей ДНФ. Сначала, применяя законы де Моргана и закон двойного отрицания, добиваемся, чтобы операция отрицания применялась только к отдельным переменным. Затем раскрываем скобки, применяя дистрибутивный закон. В результате получится формула, имеющая вид дизъюнкции конъюнкций переменных и их отрицаний. Если в какую-либо конъюнкцию входят одновременно некоторая переменная и ее отрицание, то эта конъюнкция эквивалентна 0 и может быть удалена из суммы (применяем тождества $x \cdot \bar{x} = 0$, $0 \cdot x = 0$, $0 \vee x = x$). Если в какую-либо из оставшихся конъюнкций какая-либо переменная входит более одного раза, то применяем тождество $x \cdot x = x$. После всех таких преобразований получится логическая сумма, состоящая из простых конъюнкций (или 0). Если в ней есть одинаковые простые конъюнкции, то пользуемся тождеством $x \vee x = x$, которое дает право одно из двух одинаковых слагаемых вычеркнуть из логической суммы. Полученная в результате формула будет дизъюнктивной нормальной формой.

Пример: преобразуем в ДНФ формулу $\overline{x(\bar{y} \vee y\bar{z})} \cdot \overline{(\bar{x}yz)}$.
Применяем законы де Моргана и закон двойного отрицания:

$$\overline{x(\bar{y} \vee y\bar{z})} \cdot \overline{(\bar{x}yz)} = \big(\bar{x} \vee \overline{(\bar{y} \vee y\bar{z})}\big)(x \vee \bar{y} \vee \bar{z}) = \big(\bar{x} \vee y(\bar{y} \vee z)\big)(x \vee \bar{y} \vee \bar{z}).$$

Теперь раскрываем скобки с помощью дистрибутивного закона и удаляем нулевые слагаемые:

$$\big(\bar{x} \vee y(\bar{y} \vee z)\big)(x \vee \bar{y} \vee \bar{z}) = (\bar{x} \vee yz)(x \vee \bar{y} \vee \bar{z}) = \bar{x}\bar{y} \vee \bar{x}\bar{z} \vee xyz.$$

Для каждой булевой формулы *A* можно построить и эквивалентную ей КНФ, например, следующим способом. Построим сначала ДНФ для формулы $\bar{A}$:

$$\bar{A} = K_1 \vee K_2 \vee \ldots \vee K_m,$$

каждое K_i – простая конъюнкция. Возьмем отрицание от обеих частей равенства и применим закон де Моргана:

$$A = \overline{K_1} \cdot \overline{K_2} \cdot \ldots \cdot \overline{K_m}.$$

Применив закон де Моргана еще раз к каждому сомножителю, получим произведение простых дизъюнкций, т.е. КНФ.

Таким образом, справедлива

Теорема 6.2. *Для любой булевой формулы существуют эквивалентные ей ДНФ и КНФ.*

По формуле нетрудно построить таблицу функции, представляемой этой формулой. Всегда ли можно по таблице функции найти представляющую ее булеву формулу? Сейчас мы увидим, что ответ утвердительный – имеется простой способ построения по любой таблице булевой формулы.

Совершенная дизъюнктивная нормальная форма, сокращенно СДНФ, – это ДНФ, у которой каждое слагаемое (простая конъюнкция) содержит все переменные, имеющиеся в формуле (формула, состоящая из одного символа 0, тоже считается СДНФ). Например, формула

$$xy\bar{z} \vee x\bar{y}z \vee \bar{x}\bar{y}z$$

является СДНФ, а формула

$$xy \vee \bar{x}yz \vee z$$

– не СДНФ, так как в первом слагаемом отсутствует переменная z, входящая в формулу, а в последнем слагаемом – переменные x и y. Но ее можно преобразовать в СДНФ, если воспользоваться тождеством $1 = x \vee \bar{x}$, раскрыть скобки и удалить повторяющиеся слагаемые:

$$xy \vee \bar{x}yz \vee z = xy \cdot 1 \vee \bar{x}yz \vee 1 \cdot 1 \cdot z = xy(z \vee \bar{z}) \vee \bar{x}yz \vee (x \vee \bar{x})(y \vee \bar{y})z =$$

$$= xyz \vee xy\bar{z} \vee \bar{x}yz \vee x\bar{y}z \vee \bar{x}\bar{y}z.$$

Понятно, что с помощью такого «внедрения» в простые конъюнкции недостающих переменных можно любую ДНФ преобразовать в СДНФ.

Покажем, как можно построить СДНФ по таблице функции. Введем обозначение

$$x^{\alpha} = \begin{cases} x, & \text{если } \alpha=1, \\ \bar{x}, & \text{если } \alpha=0. \end{cases}$$

x^α можно рассматривать как функцию одной переменной с параметром α. Ее можно также представить формулой:

$$x^\alpha = \alpha x \vee \bar{\alpha}\bar{x}.$$

Легко проверить, что $x^\alpha = 1$ тогда и только тогда, когда $x = \alpha$.

Рассмотрим функцию, определяемую простой конъюнкцией $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$. Конъюнкция принимает значение 1 тогда и только тогда, когда каждый сомножитель равен 1. Значит, $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} = 1$ тогда и только тогда, когда $x_1 = \alpha_1,\ x_2 = \alpha_2,\ \dots,\ x_n = \alpha_n$, т. е. справедливо следующее

Утверждение 1. *Функция* $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ *принимает значение 1 на единственном наборе* $(\alpha_1, \alpha_2, \dots, \alpha_n)$.

Пусть $f(x_1, x_2, \dots, x_n)$ – логическая функция. Возьмем какой-нибудь двоичный набор $(\alpha_1, \alpha_2, \dots, \alpha_n)$, на котором эта функция принимает значение 1 (если таких наборов нет, то функция эквивалентна константе 0, а формула 0 – это по определению тоже СДНФ). Функция $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ тоже равна 1 на этом наборе, а на всех остальных наборах она равна 0 (утверждение 1). Образуем теперь дизъюнкцию всех таких простых конъюнкций, соответствующих наборам, на которых $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1$. Получится формула, которую в общем виде можно записать так:

$$\bigvee_{f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}.$$

Условие под знаком дизъюнкции означает, что дизъюнкция берется по всем наборам, на которых функция f принимает значение 1.

Если функция f на некотором наборе равна 1, то в этой дизъюнкции присутствует слагаемое, тоже равное 1 на этом наборе и, значит, вся дизъюнкция равна 1. Если же функция f на некотором наборе равна 0, то все слагаемые в этой дизъюнкции равны 0 (каждое слагаемое равно 1 только на одном наборе, причем на таком, где $f = 1$), значит, вся дизъюнкция равна 0. Таким образом, данная формула действительно представляет функцию f. Очевидно, она является СДНФ.

Таким образом, для любой функции, не эквивалентной 0, имеет место тождество

$$f(x_1, x_2, \dots, x_n) = \bigvee_{f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}.$$

Для примера рассмотрим функцию f, заданную таблицей

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Имеется три набора, на которых функция равна 1, значит, СДНФ будет состоять из трех слагаемых:

набору (0,0,1) соответствует слагаемое $x^0y^0z^1 = \bar{x}\bar{y}z$;

набору (1,0,1) – слагаемое $x^1y^0z^1 = x\bar{y}z$;

набору (1,1,1) – слагаемое $x^1y^1z^1 = xyz$.

Искомая СДНФ: $\bar{x}\bar{y}z \vee x\bar{y}z \vee xyz$.

Итак, каждая логическая функция может быть представлена булевой формулой и притом весьма специального вида. Есть еще одно свойство совершенных ДНФ, выделяющее их из произвольных ДНФ. Одна и та же функция может быть многими способами представлена в виде ДНФ. Например, следующие три ДНФ эквивалентны, т. е. представляют одну и ту же функцию:

$$xy \vee \bar{x}yz, \quad yz \vee xy\bar{z}, \quad xy \vee yz.$$

СДНФ же, представляющая данную функцию, единственна, как показывает следующая теорема. Две СДНФ считаем одинаковыми, если они различаются только порядком сомножителей и слагаемых, т. е. одну можно превратить в другую, переставляя конъюнкции и сомножители в конъюнкциях.

Теорема 6.3. *Для любой логической функции существует единственная представляющая ее СДНФ.*

Доказательство. Существование СДНФ для любой функции было доказано выше. Докажем единственность.

Мы знаем, что имеется в точности 2^{2^n} логических функций от переменных $x_1, x_2, \dots, x_n$. Подсчитаем число различных СДНФ, содержащих все эти переменные и никаких других. Каждое слагаемое в такой СДНФ – это простая конъюнкция, имеющая вид $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$. Разные конъюнкции различаются наборами $(\alpha_1, \alpha_2, \dots, \alpha_n)$. Значит, всего имеется 2^n разных конъюнкций. Каждая СДНФ определяется некоторым подмножеством множества всех этих конъюнкций. Таких подмножеств имеется 2^{2^n}, значит, ровно столько существует СДНФ от переменных $x_1, x_2, \dots, x_n$. Это число равно числу функций от этих переменных. Но для каждой функции имеется представляющая ее СДНФ, значит, соответствие между функциями и СДНФ взаимно однозначно. ■

Основываясь на этой теореме, можно проверять (или опровергать) тождества в булевой алгебре, не прибегая к таблицам, а пользуясь только эквивалентными преобразованиями формул. Сначала преобразуем обе сравниваемые формулы в ДНФ, затем каждую ДНФ – в СДНФ. Если в одной из формул есть переменная, отсутствующая в другой, «внедряем» ее, как описано выше. Исходные формулы эквивалентны в том и только том случае, если получатся одинаковые СДНФ. Заметим, что эквивалентные преобразования, выполняемые в этой процедуре, используют только приведенные выше основные тождества 1 – 7. Следовательно, этих тождеств достаточно, чтобы вывести любое тождество в булевой алгебре.

По аналогии с совершенной ДНФ можно определить совершенную КНФ (СКНФ). В ней каждый сомножитель (простая дизъюнкция) содержит все переменные, входящие в формулу. Для построения СКНФ напишем сначала тождество, выражающее СДНФ, но только для функции $\bar{f}$:

$$\overline{f(x_1, x_2, \dots, x_n)} = \bigvee_{\overline{f(\alpha_1, \alpha_2, \dots, \alpha_n)}=1} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} = \bigvee_{f(\alpha_1, \alpha_2, \dots, \alpha_n)=0} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}.$$

Теперь возьмем отрицание от обеих частей равенства и применим закон де Моргана:

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{f(\alpha_1, \alpha_2, \dots, \alpha_n)=0} (\overline{x_1^{\alpha_1}} \vee \overline{x_2^{\alpha_2}} \vee \dots \vee \overline{x_n^{\alpha_n}}).$$

Легко проверить тождество

$$\overline{x^{\alpha}} = x^{\bar{\alpha}}.$$

Применяя его, получаем

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{f(\alpha_1, \alpha_2, \dots, \alpha_n)=0} (x_1^{\overline{\alpha_1}} \vee x_2^{\overline{\alpha_2}} \vee \dots \vee x_n^{\overline{\alpha_n}}).$$

Это и есть общая формула для СКНФ.

Главный итог настоящего раздела состоит в том, что для представления любой логической функции достаточно трех операций – отрицания, конъюнкции и дизъюнкции. Константы 0 и 1 тоже могут быть выражены через эти операции: $0 = x \cdot \bar{x}$, $1 = x \vee \bar{x}$. На самом деле достаточно двух операций: так как из законов де Моргана следует, что

$$xy = \overline{\bar{x} \vee \bar{y}},$$

$$x \vee y = \overline{\bar{x}\bar{y}},$$

то конъюнкцию или дизъюнкцию можно исключить – каждая из этих операций выражается через другую и отрицание.

6.5. Полиномы

Кроме булевой алгебры, имеются другие алгебраические системы для описания логических функций. Одной из наиболее известных является алгебра Жегалкина (И.И. Жегалкин (1869–1947) – российский и советский математик). В этой алгебре используются операции конъюнкции и сложения по модулю 2. В этом разделе под суммой везде понимается сумма по модулю 2. Отметим сначала некоторые свойства этой операции.

$x \oplus 0 = x.$

$x \oplus 1 = \bar{x}.$

$x \oplus x = 0.$

Коммутативность: $x \oplus y = y \oplus x.$

Ассоциативность: $x \oplus (y \oplus z) = (x \oplus y) \oplus z.$

Дистрибутивность: $x(y \oplus z) = xy \oplus xz.$

Уравнение $x \oplus a = b$ имеет единственное решение $x = a \oplus b.$

Благодаря свойству ассоциативности можно записывать сумму любого числа слагаемых без использования скобок. В таких случаях применяется запись

$$x_1 \oplus x_2 \oplus \ldots \oplus x_n = \sum_{i=1}^{n} x_i \,.$$

Основным видом формул в алгебре Жегалкина является *полином Жегалкина* (далее просто «полином»). Это формула, имеющая вид суммы

простых конъюнкций (без отрицаний, в алгебре Жегалкина нет операции отрицания). Одним из слагаемых может быть константа 1. Скобок в полиноме быть не должно. Не должно быть одинаковых слагаемых (две конъюнкции считаются одинаковыми, если они различаются только порядком сомножителей). Примеры полиномов:

$$x_1x_2 \oplus x_2x_3 \oplus x_1x_2x_3 \oplus x_1x_2x_3x_4, \quad 1 \oplus x_1 \oplus x_1x_2x_3, \quad x_1 \oplus x_2 \oplus x_3 \oplus x_4, \quad x, \quad 1.$$

Формула, состоящая из одного символа 0, тоже считается полиномом. А эти формулы полиномами не являются:

$$x_1(x_2 \oplus x_3), \quad x_1 \oplus \overline{x_2}, \quad x_1x_2 \oplus x_3 \oplus x_2x_1,$$

хотя каждая из них может быть легко преобразована в полином.

Три операции булевой алгебры можно выразить полиномами: $\bar{x} = x \oplus 1$, x_1x_2 – это уже полином, а для дизъюнкции имеем

$$x_1 \vee x_2 = \overline{\overline{x_1} \cdot \overline{x_2}} = (x_1 \oplus 1)(x_2 \oplus 1) \oplus 1 = x_1x_2 \oplus x_1 \oplus x_2 .$$

Теорема 6.4. *Для каждой логической функции существует единственный представляющий ее полином.*

Доказательство. Как уже известно, каждую функцию можно представить булевой формулой. Возьмем такую формулу и заменим в ней полиномами каждое отрицание и каждую дизъюнкцию, пользуясь тождествами $\bar{x} = x \oplus 1$ и $x_1 \vee x_2 = x_1x_2 \oplus x_1 \oplus x_2$. Получится формула алгебры Жегалкина. Раскроем в ней скобки, применяя дистрибутивный закон. Получится сумма конъюнкций. Каждую конъюнкцию превратим в простую конъюнкцию с помощью тождества $xx = x$. Избавимся от одинаковых слагаемых, используя тождество $x \oplus x = 0$. В результате получится полином, представляющий данную функцию.

Для доказательства взаимной однозначности соответствия между функциями и полиномами применим тот же прием, что и для доказательства аналогичного утверждения относительно СДНФ – сравним количество функций и количество полиномов от переменных $x_1, x_2, \dots, x_n$.

Каждое слагаемое в полиноме имеет вид $x_{i_1}x_{i_2} \dots x_{i_k}$ или 1. Каждая такая конъюнкция определяется подмножеством $\{i_1, i_2, \dots, i_n\}$ множества $\{1,2, \dots n\}$ (пустому подмножеству соответствует слагаемое 1). Значит, множество всех возможных слагаемых содержит столько элементов, сколько имеется подмножеств у n–элементного множества, т. е. 2^n. Чтобы составить полином, нужно выбрать подмножество множества всех возможных слагаемых. Таких подмножеств имеется 2^{2^n}, это и есть число полиномов от n переменных. Как мы знаем, столько же имеется функций от n переменных. Так как для каждой функции существует представляющий ее полином и

число функций равно числу полиномов, то имеется ровно один представляющий полином для каждой функции. ■

В доказательстве теоремы 6.4 описан способ преобразования булевой формулы в полином. Если исходной формулой является СДНФ, то процесс преобразования можно несколько упростить. Вспомним формулу для СДНФ:

$$f(x_1, x_2, \dots, x_n) = \bigvee_{f(\alpha_1,\alpha_2,\dots,\alpha_n)=1} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}.$$

Здесь каждое слагаемое $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ принимает значение 1 ровно на одном наборе значений переменных $(\alpha_1, \alpha_2, \dots, \alpha_n)$. Разные слагаемые принимают значение 1 на разных наборах. Значит, при любых значениях переменных в этой дизъюнкции либо имеется ровно одно слагаемое, равное 1, либо все слагаемые равны 0. Поэтому равенство сохранится, если логическую сумму заменить суммой по модулю 2:

$$f(x_1, x_2, \dots, x_n) = \sum_{f(\alpha_1,\alpha_2,\dots,\alpha_n)=1} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}.$$

Теперь остается заменить каждое $\bar{x_i}$ на $x_i \oplus 1$, раскрыть скобки и привести подобные.

Имеется способ построения полинома непосредственно по таблице функции – метод неопределенных коэффициентов. Общая формула полинома от переменных $x_1, x_2, \dots, x_n$ может быть записана следующим образом:

$$\sum_{\{i_1,i_2,\dots,i_k\}\subseteq\{1,2,\dots,n\}} a_{i_1,i_2,\dots,i_k} x_{i_1} x_{i_2} \dots x_{i_k}.$$

Здесь $a_{i_1,i_2,\dots,i_k}$ – коэффициент при конъюнкции $x_{i_1} x_{i_2} \dots x_{i_k}$, он равен 1 или 0 в зависимости от того, входит или не входит эта конъюнкция в полином в качестве слагаемого. В частности, при $n = 3$ эта формула имеет вид:

$$a_\emptyset \oplus a_1 x_1 \oplus a_2 x_2 \oplus a_3 x_3 \oplus a_{1,2} x_1 x_2 \oplus a_{1,3} x_1 x_3 \oplus a_{2,3} x_2 x_3 \oplus a_{1,2,3} x_1 x_2 x_3.$$

Чтобы определить конкретный полином, надо найти все коэффициенты. Это можно сделать, используя таблицу функции, следующим образом. Берем какой-нибудь набор значений переменных и подставляем эти значения в общую формулу полинома. Получается сумма некоторых коэффициентов. Приравниваем эту сумму к значению функции на данном наборе. Получается уравнение для коэффициентов. Всего таким образом получается система из 2^n уравнений, при этом имеется как раз 2^n неизвестных коэффициентов. Можно доказать, что эта система всегда имеет решение. На самом деле она решается очень просто: если рассматривать

наборы в лексикографическом порядке, и использовать вычисленные ранее коэффициенты, то каждое следующее уравнение будет уравнением простейшего вида и позволяет найти еще один коэффициент.

Построим для примера полином для функции $x_1 \to x_2$. Напомним таблицу этой функции:

x_1	x_2	$x_1 \to x_2$
0	0	1
0	1	1
1	0	0
1	1	1

Общий вид полинома для двух переменных:

$$a_\emptyset \oplus a_1 x_1 \oplus a_2 x_2 \oplus a_{1,2} x_1 x_2 .$$

Подставляя сюда $x_1 = 0$ и $x_2 = 0$ из первой строки таблицы и приравнивая полученное значение полинома к значению функции, т. е. к 1, получаем

$$a_0 = 1.$$

Из второй строки таблицы получаем $a_0 \oplus a_2 = 1$. Подставляя сюда уже известное значение a_0, получаем $1 \oplus a_2 = 1$, отсюда

$$a_2 = 0.$$

Аналогично, используя третью строку, находим

$$a_1 = 1.$$

Наконец, из последней строки, используя все найденные значения, находим

$$a_{1,2} = 1.$$

Подставляя найденные коэффициенты в общую формулу, получаем полином для импликации:

$$x_1 \to x_2 = 1 \oplus x_1 \oplus x_1 x_2 .$$

Глава 7. Логические функции. Полные системы

7.1. Суперпозиция, замкнутость и полнота

Итак, мы имеем два типа формул, с помощью которых можно описывать любые логические функции – булевы формулы и полиномы Жегалкина. Это означает, что каждый из наборов элементарных функций $\{0,1,\neg,\&,\vee\}$ и $\{0,1,\&,\oplus\}$ достаточен для того, чтобы выразить любую функцию. Такие наборы называют полными системами функций (точное определение будет дано позднее). Возможно, имеются другие полные системы, причем не обязательно состоящие из элементарных функций (содержащие, быть может, и функции более чем от двух переменных). В связи с этим возникает вопрос: как по множеству функций узнать, является ли оно полной системой? Этот вопрос и его разрешение – главная тема настоящей главы. Введем сначала некоторые необходимые понятия.

В предыдущей главе было введено понятие эквивалентных функций. Это такие функции, которые совпадают «с точностью до фиктивных переменных». Для большей ясности дадим еще одно определение этого понятия. Пусть f и g – две функции, а $x_1, x_2, \dots, x_n$ – список всех переменных, от которых зависит хотя бы одна из них (например, для функций $(1 \vee x_1)x_2$ и $x_2 \vee x_3\overline{x_3}$ это будут переменные x_1, x_2, x_3). Функции f и g эквивалентны, если их значения совпадают при любых значениях переменных $x_1, x_2, \dots, x_n$. Далее будем рассматривать множества функций, обладающих свойством: если некоторая функция f принадлежит множеству, то ему принадлежит и любая функция, эквивалентная f. Такое множество будем называть *классом* функций. Иначе говоря, класс функций – это множество логических функций, замкнутое относительно операций удаления и введения фиктивных переменных. Все множества функций, рассматриваемые в этой лекции – это на самом деле классы. Например, мы записываем множество $\{0,1\}$ и говорим, что это класс констант. На самом деле подразумевается бесконечное множество функций, содержащее две функции от n переменных при каждом n (и все эти переменные фиктивные).

Класс всех логических функций обозначается через P_2.

Рассмотрим две операции над функциями.

1) Операция *переименования переменных* состоит в том, что переменным даются новые имена. Функция $f(x_1, x_2, \dots, x_n)$ в результате переименования переменной x_1 в y_1, x_2 в y_2, …, x_n в y_n превращается в функцию $f(y_1, y_2, \dots, y_n)$. Некоторые переменные могут сохранить старые имена, например, может быть $y_1 = x_1$. Новое имя может быть одним из старых имен, отличным от имени переименовываемой переменной,

например, $y_1 = x_2$. Две разных переменных могут получить одно новое имя, например, $y_1 = x_1$ и $y_2 = x_1$ (это называется *отождествлением* переменных). Например, из функции $x_1 \to x_2$ путем переименования переменных можно получить функции $x_3 \to x_4$, $x_2 \to x_1$, $x \to x$ (заметим, что из функции $x \to x$ получить функцию $x_1 \to x_2$ переименованием переменных нельзя).

2) Операция *подстановки* функции в функцию вместо некоторой переменной. Результатом подстановки в функцию $f(x_1, x_2, \dots, x_n)$ вместо переменной x_k функции $g(y_1, y_2, \dots, y_\mathrm{m})$ является функция

$$f(x_1, \dots, x_{\mathrm{k}-1}, g(y_1, y_2, \dots, y_m), x_{\mathrm{k}+1} \dots, x_n).$$

При этом некоторые из имен переменных $y_1, y_2, \dots, y_m$ могут совпадать с некоторыми из имен $x_1, x_2, \dots, x_n$. Например, подставляя в функцию $x_1 \vee x_2$ вместо переменной x_2 функцию $x_2 \vee x_3$, получим функцию $x_1 \vee x_2 \vee x_3$.

Пусть A – некоторое множество функций. Говорят, что функция f является *суперпозицией* функций из множества A, если она может быть получена из них с помощью операций переименования переменных и подстановки. Например, функция $x_1 \vee x_2 \vee x_3$ является суперпозицией функций из множества $\{x_1 \vee x_2\}$.

Множество всех суперпозиций функций из множества A называется *замыканием* множества A и обозначается через $[A]$. Класс функций называется *замкнутым*, если он совпадает со своим замыканием, т. е. $A = [A]$.

Рассмотрим несколько примеров.

1. Пусть $A = \{0,1\}$. Так как переменные здесь только фиктивные, то никакие переименования или подстановки не дадут ничего нового. Следовательно, $[\{0,1\}] = \{0,1\}$, т. е. класс констант – замкнутый.

2. $A = \{0, \bar{x}\}$. Если в функцию $\bar{x}$ вместо переменной x подставить функцию 0, то получится функция $\bar{0} = 1$. Если в $\bar{x}$ вместо переменной x подставить ту же функцию $\bar{x}$, то получится функция $\bar{\bar{x}} = x$. Таким образом, замыкание множества $\{0, \bar{x}\}$ содержит все функции одной переменной. Никакими подстановками или переименованиями из функции одной переменной нельзя получить функцию, существенно зависящую от большего числа переменных. Следовательно, $[\{0, \bar{x}\}] = \{0,1, x, \bar{x}\}$.

3. $A = \{0, 1, \bar{x}, x_1x_2, x_1 \vee x_2\}$. Всякую логическую функцию можно представить булевой формулой. Всякую булеву формулу можно рассматривать как описание суперпозиции функций из этого множества A. Например, формула $x_1 \vee x_2\overline{x_3}$ описывает функцию, которую можно получить так: в функцию x_1x_2 подставляем вместо переменной x_2 функцию $\overline{x_3}$, у

полученной функции переменную x_1 переименовываем в x_2 и результат подставляем в функцию $x_1 \vee x_2$ вместо переменной x_2. Значит, всякая логическая функция является суперпозицией функций из A, т. е. $[\{0,1,\bar{x}, x_1x_2, x_1 \vee x_2\}] = P_2$.

4. $A = \{x_1 \vee x_2\}$. Мы уже видели, что дизъюнкцию трех переменных можно получить как суперпозицию из дизъюнкции двух переменных. Ясно, что таким образом можно получить дизъюнкцию любого числа переменных. Ясно также, что при подстановке в дизъюнкцию нескольких переменных вместо одной из переменных другой дизъюнкции нескольких переменных снова получится дизъюнкция нескольких переменных. Таким образом, замыкание этого множества состоит из всевозможных дизъюнкций. Этому замыканию принадлежит и тождественная функция – она получается из дизъюнкции отождествлением переменных. Следовательно, $[\{x_1 \vee x_2\}] = \{x, x_1 \vee x_2,\ x_1 \vee x_2 \vee x_3, \dots\}$.

Отметим некоторые свойства замыкания.

1. $A \subseteq [A]$.

2. $\big[[A]\big] = [A]$.

3. Если $A \subseteq B$, то $[A] \subseteq [B]$.

Справедливость этих утверждений непосредственно следует из определений суперпозиции и замыкания.

Множество функций называется *полной системой функций*, если любая логическая функция является суперпозицией функций их этого множества. Иначе говоря, полная система – это такое множество A, что $[A] = P_2$.

Примеры.

1. Мы уже видели, что $[\{0, 1, \bar{x},\ x_1x_2,\ x_1 \vee x_2\}] = P_2$. Значит,

$\{0, 1,\ \bar{x},\ x_1x_2,\ x_1 \vee x_2\}$ – полная система.

2. Обе константы можно выразить через отрицание, конъюнкцию и дизъюнкцию: $0 = x \cdot \bar{x}$, $1 = x \vee \bar{x}$. Поэтому из предыдущего примера следует, что $\{\bar{x},\ x_1x_2,\ x_1 \vee x_2\}$ – полная система.

3. Дизъюнкцию можно выразить через конъюнкцию и отрицание: $x_1 \vee x_2 = \overline{\overline{x_1} \cdot \overline{x_2}}$. Поэтому из предыдущего примера следует, что $\{\bar{x}, x_1x_2\}$ – полная система. Этот пример особенно важен – полнота этой системы будет использоваться при доказательстве общего критерия полноты.

4. Конъюнкцию тоже можно выразить через дизъюнкцию и отрицание: $x_1x_2 = \overline{\overline{x_1} \vee \overline{x_2}}$. Значит, система $\{\bar{x}, x_1 \vee x_2\}$ тоже полна.

5. Всякую логическую функцию можно представить полиномом Жегалкина. Полином можно рассматривать как формулу, выражающую некоторую функцию как суперпозицию функций 0, 1, x_1x_2, $x_1 \oplus x_2$. Значит, эти функции образуют полную систему. Так как $0 = 1 \oplus 1$, то множество $\{1, x_1x_2, x_1 \oplus x_2\}$ является полной системой.

Следующая теорема является главным инструментом в решении проблемы полноты.

Теорема 7.1 (теорема сведения). *Пусть A и B – множества функций, причем A – полная система и каждая функция из A является суперпозицией функций из B. Тогда B – тоже полная система.*

Доказательство. Так как каждая функция из A является суперпозицией функций из B, то $A \subseteq [B]$. По свойствам замыкания имеем $[A] \subseteq [[B]] = [B]$. Так как A – полная система, то $[A] = P_2$. Следовательно, $P_2 \subseteq [B]$. Но множество P_2 состоит из всех логических функций, значит, может быть только равенство $P_2 = [B]$. А это означает, что B – полная система. ■

Для примера возьмем $A = \{\bar{x}, x_1x_2\}$, $B = \{x_1|x_2\}$. Мы знаем, что множество A – полная система. Обе функции из A выражаются через функцию из B: $\bar{x} = x|x$, $x_1x_2 = \overline{x_1|x_2} = (x_1|x_2)|(x_1|x_2)$. Значит, B – полная система. Таким образом, каждая логическая функция может быть представлена формулой, в которой используется только операция штрих Шеффера. Например, $x_1 \vee x_2 = (x_1|x_1)|(x_2|x_2)$. Нетрудно показать, что тем же свойством обладает стрелка Пирса.

7.2. Важнейшие замкнутые классы

Рассмотрим замкнутые классы, играющие особую роль в решении вопроса о полноте.

7.2.1. Функции, сохраняющие константы

Говорят, что функция $f(x_1, x_2, \dots, x_n)$ *сохраняет константу* 0, если $f(0,0,\dots,0) = 0$. Класс всех функций, сохраняющих 0, обозначается через T_0.

Из элементарных функций классу T_0 принадлежат сама константа 0, тождественная функция, конъюнкция, дизъюнкция, сумма по модулю 2. Константа 1, отрицание, импликация, эквиваленция, штрих Шеффера и стрелка Пирса этому классу не принадлежат.

Теорема 7.2. *Класс T_0 замкнут.*

Доказательство. Достаточно доказать, что при применении операций переименования переменных и подстановки к функциям из класса T_0 получается функция из этого же класса. Для переименования это очевидно. Рассмотрим операцию подстановки. Пусть $f(x_1, x_2, \dots, x_n) \in T_0$ и $g(y_1, y_2, \dots, y_\mathrm{m}) \in T_0$, а функция h получается в результате подстановки функции g в функцию f вместо переменной x_k. Доказательство одинаковое при любом k, так что мы не потеряем общности, если положим $k = n$. Итак, $h(x_1, \dots, x_{n-1}, y_1, \dots, y_m) = f(x_1, \dots, x_{n-1}, g(y_1, \dots, y_m))$. Отметим, что некоторые из переменных $y_1, y_2, \dots, y_\mathrm{m}$ могут совпадать с некоторыми из переменных $x_1, x_2, \dots, x_n$, такие повторения пропускаются в списке переменных функции h, т. е. фактически она может зависеть не от $m + n - 1$, а от меньшего числа переменных. Подставляя нулевые значения, получаем

$$h(0, \dots, 0, 0, \dots, 0) = f\big(0, \dots, 0, g(0, \dots, 0)\big) = f(0, \dots, 0, 0) = 0,$$

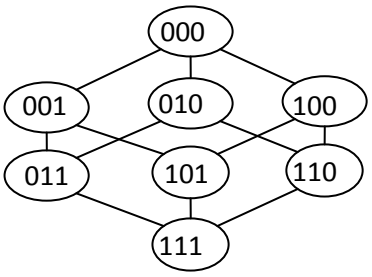
следовательно, $h \in T_0$. ■

Нетрудно подсчитать число функций от n переменных в классе T_0. Значение такой функции на нулевом наборе значений переменных зафиксировано, а на остальных $2^n - 1$ наборах функция может принимать любые значения из множества $\{0,1\}$. Имеется $2^{2^n-1} = \frac{1}{2}2^{2^n}$ таких функций, т. е. ровно половина всех логических функций (другая половина – это функции с $f(0,0, \dots, 0) = 1$).

Функция $f(x_1, x_2, \dots, x_n)$ *сохраняет константу* 1, если $f(1,1, \dots, 1) = 1$. Класс всех функций, сохраняющих 1, обозначается через T_1. Этот класс содержит константу 1, тождественную функцию, конъюнкцию, дизъюнкцию, импликацию, эквиваленцию и не содержит константу 0, отрицание, сумму по модулю 2, штрих Шеффера и стрелку Пирса. Замкнутость класса T_1 доказывается точно так же, как замкнутость класса T_0.

7.2.2. Монотонные функции

Определим на множестве всех двоичных наборов длины n отношение $\preccurlyeq$ следующим образом: для наборов $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ и $\tilde{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$ имеет место $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ тогда и только тогда, когда $\alpha_i \le \beta_i$ для всех $i = 1,2, \dots, n$. Прочитать запись $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ можно так: «$\tilde{\alpha}$ покомпонентно меньше или равен $\tilde{\beta}$». Легко проверить, что это отношение порядка, т. е. оно рефлексивно, антисимметрично и транзитивно. Этот порядок частичный, так как не любые два набора сравнимы. Например, при $n = 2$ наборы (0,1) и (1,0) несравнимы. На рисунке показана диаграмма Хассе этого отношения при $n = 3$.



Функция $f(x_1, x_2, \dots, x_n)$ называется *монотонной*, если всякий раз, когда $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, имеет место $f(\tilde{\alpha}) \leq f(\tilde{\beta})$. Класс всех монотонных функций обозначается через M.

Из элементарных функций монотонными являются обе константы, тождественная функция, конъюнкция и дизъюнкция, остальные немонотонны.

Теорема 7.3. *Класс M замкнут.*

Доказательство. Как и в доказательстве предыдущей теоремы, достаточно рассмотреть операцию подстановки. Пусть $f(x_1, x_2, \dots, x_n) \in M$ и $g(y_1, y_2, \dots, y_{\mathrm{m}}) \in M$, а функция h получается в результате подстановки функции g в функцию f вместо переменной x_n:

$$h(x_1, \dots, x_{n-1}, y_1, \dots, y_m) = f(x_1, \dots, x_{n-1}, g(y_1, \dots, y_m)).$$

Пусть $(\alpha_1', \dots, \alpha_{n-1}', \beta_1', \dots, \beta_m')$ и $(\alpha_1'', \dots, \alpha_{n-1}'', \beta_1'', \dots, \beta_m'')$ – два набора значений переменных функции h, причем

$$(\alpha_1', \dots, \alpha_{n-1}', \beta_1', \dots, \beta_m') \preccurlyeq (\alpha_1'', \dots, \alpha_{n-1}'', \beta_1'', \dots, \beta_m'').$$

Тогда

$$(\beta_1', \dots, \beta_m') \preccurlyeq (\beta_1'', \dots, \beta_m'').$$

Так как функция g монотонна, то

$$g(\beta_1', \dots, \beta_m') \leq g(\beta_1'', \dots, \beta_m'').$$

Поэтому

$$(\alpha_1', \dots, \alpha_{n-1}', g(\beta_1', \dots, \beta_m')) \preccurlyeq (\alpha_1'', \dots, \alpha_{n-1}'', g(\beta_1'', \dots, \beta_m'')),$$

а так как функция f тоже монотонна, то

$$f(\alpha_1', \dots, \alpha_{n-1}', g(\beta_1', \dots, \beta_m')) \leq f(\alpha_1'', \dots, \alpha_{n-1}'', g(\beta_1'', \dots, \beta_m'')),$$

т. е.

$$h(\alpha_1', \dots, \alpha_{n-1}', \beta_1', \dots, \beta_m') \leq h(\alpha_1'', \dots, \alpha_{n-1}'', \beta_1'', \dots, \beta_m'').$$

Значит, функция h монотонна. ■

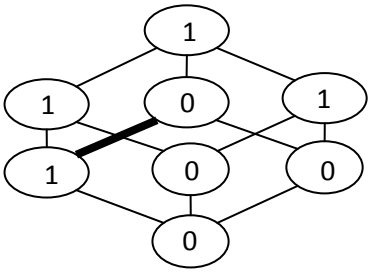
Для проверки монотонности заданной функции, если следовать определению, нужно перебрать всевозможные пары сравнимых наборов и сравнить значения функции на каждой такой паре. Следующая лемма показывает, что можно обойтись значительно меньшим числом проверок. Два набора называются *соседними*, если они различаются точно в одной компоненте.

Лемма 7.1. *Если функция немонотонна, то существуют соседние наборы, на которых нарушается монотонность.*

Доказательство. Два соседних набора всегда сравнимы и более того, они связаны отношением непосредственного предшествования относительно порядка $\preccurlyeq$. Верно и обратное – если один набор непосредственно предшествует другому по отношению $\preccurlyeq$, то эти наборы соседние.

Пусть f – немонотонная функция. Существуют такие наборы значений переменных $\tilde{\alpha}$ и $\tilde{\beta}$, что $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ и $f(\tilde{\alpha}) > f(\tilde{\beta})$. Значит, $f(\tilde{\alpha}) = 1, f(\tilde{\beta}) = 0$. По теореме о конечных упорядоченных множествах из лекции 2 существует такая последовательность наборов $\widetilde{\gamma_1}, \widetilde{\gamma_2}, \dots, \widetilde{\gamma_k}$, что $\widetilde{\gamma_1} = \tilde{\alpha}, \widetilde{\gamma_k} = \tilde{\beta}$ и набор $\widetilde{\gamma_i}$ непосредственно предшествует набору $\tilde{\gamma}_{i+1}$ для $i = 1,2, \dots, k-1$. Так как $f(\tilde{\gamma}_1) = 1$, а $f(\tilde{\gamma}_k) = 0$, то найдется такое i, что $f(\tilde{\gamma}_i) = 1, f(\tilde{\gamma}_{i+1}) = 0$. Но тогда $\tilde{\gamma}_i$ и $\tilde{\gamma}_{i+1}$ – соседние наборы, на которых нарушается монотонность. ■

При небольшом числе переменных проверку монотонности можно производить графически, с помощью диаграммы Хассе отношения $\preccurlyeq$ (которая изображает как раз отношение непосредственного предшествования). Нужно расставить значения функции в вершинах диаграммы. Функция монотонна, если эти значения не убывают при движении по этой диаграмме снизу вверх. На рисунке показан пример диаграммы для функции с набором значений $\tilde{f} = 01010011$. Выделено ребро, на котором нарушается монотонность.



Обе константы – монотонные функции. Так как класс монотонных функций замкнут, то подставляя в монотонную функцию вместо некоторых переменных константы, всегда получим монотонную функцию от меньшего числа переменных. Если же подставлять константы вместо переменных в

немонотонную функцию, то может получиться как немонотонная, так и монотонная функция. Например, функция $x \to 0 = \bar{x}$ немонотонна, а функция $x \to 1 = 1$ монотонна. Оказывается, в любую немонотонную функцию можно так подставить константы, что получится простейшая немонотонная функция – отрицание (единственная немонотонная функция одной переменной). Этот факт важен для дальнейшего, докажем его.

Лемма 7.2 (о немонотонной функции). *Если функция f немонотонна, то функция $\bar{x}$ является суперпозицией функций 0, 1 и f.*

Доказательство. Пусть $f(x_1, x_2, \dots, x_n)$ – немонотонная функция. По лемме 12.1 существуют соседние наборы, на которых нарушается монотонность. Пусть $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ и $\tilde{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$ – такие наборы, т.е. $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, $f(\tilde{\alpha}) > f(\tilde{\beta})$. Тогда $f(\tilde{\alpha}) = 1$, $f(\tilde{\beta}) = 0$. Так как $\tilde{\alpha}$ и $\tilde{\beta}$ соседние, то существует такое k, что $\alpha_k = 0$, $\beta_k = 1$, $\alpha_i = \beta_i$ при $i \neq k$. Подставим в функцию f вместо каждой переменной x_i, $i \neq k$, соответствующую константу α_i, а переменную x_k переименуем в x. Получим функцию одной переменной

$$h(x) = f(\alpha_1, \dots, \alpha_{k-1}, x, \alpha_{k+1}, \dots, \alpha_n).$$

Имеем

$$h(0) = f(\alpha_1, \dots, \alpha_{k-1}, 0, \alpha_{k+1}, \dots, \alpha_n) = f(\tilde{\alpha}) = 1,$$

$$h(1) = f(\alpha_1, \dots, \alpha_{k-1}, 1, \alpha_{k+1}, \dots, \alpha_n) = f(\tilde{\beta}) = 0.$$

Следовательно, $h(x) = \bar{x}$. ■

Для примера рассмотрим функцию с вектором значений $\tilde{f} = 01010011$. Она немонотонна, так как $f(0,0,1) = 1$, $f(1,0,1) = 0$. Делая подстановку и переименование, как указано в доказательстве, получаем $f(x, 0, 1) = \bar{x}$.

7.2.3. Самодвойственные функции

Двойственной функцией к функции $f(x_1, x_2, \dots, x_n)$ называется функция $\overline{f(\overline{x_1}, \overline{x_2}, \dots, \overline{x_n})}$. Она обозначается $f^*(x_1, x_2, \dots, x_n)$. Очевидно, $(f^*)^* = f$.

Примеры: двойственной функцией к константе 0 является константа 1, к тождественной функции – сама эта функция, к конъюнкции – дизъюнкция, к сумме по модулю 2 – зквиваленция, к штриху Шеффера – стрелка Пирса.

Функция называется *самодвойственной*, если двойственная функция совпадает с ней самой, т. е. выполняется тождество

$$f(x_1, x_2, \dots, x_n) = \overline{f(\overline{x_1}, \overline{x_2}, \dots, \overline{x_n})}.$$

Класс всех самодвойственных функций обозначается через S.

Среди элементарных функций самодвойственными являются только тождественная функция и отрицание. Самодвойственных функций, существенно зависящих от двух переменных, вообще не существует. В качестве примера самодвойственной функции трех переменных можно привести т. н. *медиану* – функцию $m(x,y,z) = xy \vee xz \vee yz$. Действительно,

$$m^*(x,y,z) = \overline{\bar{x}\bar{y} \vee \bar{x}\bar{z} \vee \bar{y}\bar{z}} = (x \vee y)(x \vee z)(y \vee z) = xy \vee xz \vee yz = \\ = m(x,y,z).$$

Теорема 7.4. *Класс S замкнут.*

Доказательство. Как и раньше, достаточно рассмотреть операцию подстановки. Пусть $f(x_1, x_2, \dots, x_n) \in S,\ g(y_1, y_2, \dots, y_\mathrm{m}) \in S,$

$$h(x_1, \dots, x_{n-1}, y_1, \dots, y_m) = f(x_1, \dots, x_{n-1}, g(y_1, \dots, y_m)).$$

Тогда

$$h^*(x_1, \dots, x_{n-1}, y_1, \dots, y_m) = \overline{f\left(\overline{x_1}, \dots, \overline{x_{n-1}}, g(\overline{y_1}, \dots, \overline{y_m})\right)} = \\ = \overline{f(\overline{x_1}, \dots, \overline{x_{n-1}}, \overline{g^*(y_1, \dots, y_m)})} = f^*\left(x_1, \dots, x_{n-1}, g^*(y_1, \dots, y_m)\right) = \\ = f\left(x_1, \dots, x_{n-1}, g(y_1, \dots, y_m)\right) = h(x_1, \dots, x_{n-1}, y_1, \dots, y_m).$$

Следовательно, h – самодвойственная функция. ■

Наборы $(\alpha_1, \alpha_2, \dots, \alpha_n)$ и $(\overline{\alpha_1}, \overline{\alpha_2}, \dots, \overline{\alpha_n})$ называют *противоположными*. Определение самодвойственности можно сформулировать и так: самодвойственная функция – это такая, которая на любой паре противоположных наборов принимает противоположные значения. Из этого определения видно, как по таблице функции узнать, самодвойственна ли она. Если наборы значений переменных расставляются в таблице в лексикографическом порядке, то противоположные наборы располагаются симметрично относительно середины таблицы.

Нетрудно подсчитать число самодвойственных функций от n переменных. Разобьем множество всех двоичных наборов длины n на два подмножества: одно составляют наборы с первой компонентой 0, другое – с первой компонентой 1 (при лексикографическом порядке расположения наборов в таблице функции одно подмножество заполняет верхнюю половину таблицы, второе – нижнюю). Для каждого набора из одного подмножества противоположный набор принадлежит другому подмножеству. Значит, если функция самодвойственная, то она полностью определяется своими значениями на одном из этих подмножеств. Так как каждое подмножество состоит из 2^{n-1} наборов, то получается, что число самодвойственных функций равно $2^{2^{n-1}}$.

Простейшие несамодвойственные функции – константы 0 и 1. Аналогично лемме о немонотонной функции справедлива

Лемма 7.3 (о несамодвойственной функции). *Если функция f несамодвойственна, то одна из констант является суперпозицией функций $\bar{x}$ и f.*

Доказательство. Пусть $f(x_1, x_2, \dots, x_n)$ – несамодвойственная функция. Тогда существует такой набор $(\alpha_1, \alpha_2, \dots, \alpha_n)$, что $f(\alpha_1, \alpha_2, \dots, \alpha_n) = f(\overline{\alpha_1}, \overline{\alpha_2}, \dots, \overline{\alpha_n}) = \mathrm{c}$. Для каждого $i = 1,2, \dots, n$ подставим в функцию f вместо переменной x_i функцию $x \oplus \alpha_i$. Если $\alpha_i = 0$, то подставляется x, что можно трактовать как переименование переменной x_i в x, а если $\alpha_i = 1$, то подставляется $x \oplus 1 = \bar{x}$. Получается функция одной переменной $h(x) = f(x \oplus \alpha_1, x \oplus \alpha_2, \dots, x \oplus \alpha_n)$. Имеем

$h(0) = f(\alpha_1, \alpha_2, \dots, \alpha_n) = c,$

$h(1) = f(1 \oplus \alpha_1, 1 \oplus \alpha_2, \dots, 1 \oplus \alpha_n) = f(\overline{\alpha_1}, \overline{\alpha_2}, \dots, \overline{\alpha_n}) = c.$

Следовательно, $h(x) = c$. ■

Для примера рассмотрим функцию с вектором значений $\tilde{f} = 01010011$. Она несамодвойственная, так как $f(0,0,1) = f(1,1,0)$. Делая подстановку и переименование, как указано в доказательстве, получаем $f(\bar{x}, \bar{x}, x) = 1$.

7.2.4. Линейные функции

Функция называется *линейной*, если она может быть представлена формулой вида

$$a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \dots \oplus a_n x_n.$$

Здесь $a_0, a_1, a_2, \dots, a_n$ – коэффициенты, каждый из которых равен 0 или 1. Класс всех линейных функций обозначается через L.

При $n = 1$ общий вид линейной функции $a_0 \oplus a_1 x_1$. Перебирая всевозможные комбинации двух коэффициентов, получаем 4 функции: 0 , 1 , x_1 , $1 \oplus x_1 = \overline{x_1}$. Таким образом, все функции одной переменной линейные.

При $n = 2$ для того, чтобы линейная функция существенно зависела от обеих переменных x_1 и x_2 , нужно, чтобы оба коэффициента a_1 и a_2 были равны 1. Варьировать можно только коэффициент a_0 и соответственно получается ровно две таких функции: $x_1 \oplus x_2$ и $1 \oplus x_1 \oplus x_2 = x_1 \sim x_2$. Все остальные элементарные функции, следовательно, нелинейные.

Теорема 7.5. Класс L замкнут.

Доказательство. Опять проверяем замкнутость относительно подстановки. Пусть $f(x_1, x_2, \dots, x_n) \in L,\ g(y_1, y_2, \dots, y_\text{m}) \in L,$

$h(x_1, \dots, x_{n-1}, y_1, \dots, y_m) = f(x_1, \dots, x_{n-1}, g(y_1, \dots, y_m)).$

Тогда

$f(x_1, x_2, \dots, x_n) = a_0 \oplus a_1 x_1 \oplus \dots \oplus a_n x_n$,
$g(y_1, y_2, \dots, y_\text{m}) = b_0 \oplus b_1 y_1 \oplus \dots \oplus b_m y_m,$

$h(x_1, \dots, x_{n-1}, y_1, \dots, y_m) =$

$= a_0 \oplus a_1 x_1 \oplus \dots \oplus a_{n-1} x_{n-1} \oplus a_n (b_0 \oplus b_1 y_1 \oplus \dots \oplus b_m y_m) =$

$= a_0 \oplus a_n b_0 \oplus a_1 x_1 \oplus \dots \oplus a_{n-1} x_{n-1} \oplus a_n b_1 y_1 \oplus \dots \oplus a_n b_m y_m.$

Видно, что h – линейная функция. ■

Как узнать, является ли заданная функция линейной? Можно заметить, что общий вид линейной функции – это частный случай полинома Жегалкина. Это такой полином, в котором нет произведений переменных. Полином Жегалкина для каждой функции единственный и нам известны способы его построения. Таким образом, чтобы выяснить, линейна ли данная функция, достаточно построить для нее полином Жегалкина и посмотреть, есть ли в нем произведения переменных.

В общей формуле линейной функции от n переменных имеется $n + 1$ коэффициентов, каждый из которых может принимать два значения. Всего получается 2^{n+1} функций. Но линейных функций, существенно зависящих от всех n переменных, значительно меньше. В самом деле, если $a_i = 0, i > 0,$ то переменная x_i в формулу не входит. Значит, чтобы все n переменных были существенными, нужно, чтобы $a_1 = a_2 = \dots = a_n = 1$. Остается только коэффициент a_0. Значит, при любом n имеется ровно две линейных функции, существенно зависящих от переменных $x_1, x_2, \dots, x_n$ – это $x_1 \oplus x_2 \oplus \dots \oplus x_n$ и $1 \oplus x_1 \oplus x_2 \oplus \dots \oplus x_n$. Это дает еще один способ проверки линейности – нужно выявить и удалить все фиктивные переменные, полученную функцию сравнить с одной из этих двух.

Лемма 7.4. (о нелинейной функции). *Если функция f нелинейна, то конъюнкция является суперпозицией функций 0, 1, $\bar{x}$ и f.*

Доказательство. Пусть $f(x_1, x_2, \dots, x_n)$ – нелинейная функция. Тогда в полином Жегалкина этой функции входит произведение каких-нибудь переменных. Пусть для определенности это произведение переменных x_1 и x_2. Это произведение может входить в несколько слагаемых, сгруппируем эти слагаемые и вынесем из их суммы произведение $x_1 x_2$ за скобки. В скобках останется полином от остальных переменных, обозначим его $p_0(x_3, \dots, x_n)$. Сгруппируем также все слагаемые, содержащие x_1, но не

содержащие x_2, и вынесем за скобки x_1, в скобках останется полином $p_1(x_3, \dots, x_n)$. Так же поступим со слагаемыми, содержащими x_2, но не содержащими x_1, в скобках останется полином $p_2(x_3, \dots, x_n)$. Наконец, все слагаемые, не содержащие ни x_1, ни x_2, образуют полином $p_3(x_3, \dots, x_n)$. Таким образом, имеем

$$f(x_1, x_2, \dots, x_n) = x_1 x_2 p_0(x_3, \dots, x_n) \oplus x_1\, p_1(x_3, \dots, x_n) \oplus x_2 p_2(x_3, \dots, x_n) \oplus \\ \oplus p_3(x_3, \dots, x_n).$$

Заметим, что функция $p_0(x_3, \dots, x_n)$ не равна тождественно 0, иначе оказалось бы, что функция f эквивалентна полиному

$$x_1\, p_1(x_3, \dots, x_n) \oplus x_2 p_2(x_3, \dots, x_n) \oplus p_3(x_3, \dots, x_n),$$

не содержащему произведения $x_1 x_2$, что противоречит единственности полинома и предположению о том, что данное произведение входит в этот единственный полином. Значит, существуют такие значения $\alpha_3, \dots, \alpha_n$, что $p_0(\alpha_3, \dots, \alpha_n) = 1$. Пусть $p_1(\alpha_3, \dots, \alpha_n) = a_1$, $p_2(\alpha_3, \dots, \alpha_n) = a_2$, $p_3(\alpha_3, \dots, \alpha_n) = a_3$. Подставляя константы $\alpha_3, \dots, \alpha_n$ вместо переменных $x_3, \dots, x_n$ в функцию f и в полином, получаем

$$f(x_1, x_2, \alpha_3, \dots, \alpha_n) = x_1 x_2 \oplus a_1 x_1 \oplus a_2 x_2 \oplus a_3.$$

Теперь подставим вместо x_1 функцию $x_1 \oplus a_2$, а вместо x_2 функцию $x_2 \oplus a_1$. Получим

$$f(x_1 \oplus a_2, x_2 \oplus a_1, \alpha_3, \dots, \alpha_n) = \\ = (x_1 \oplus a_2)(x_2 \oplus a_1) \oplus a_1 (x_1 \oplus a_2) \oplus a_2 (x_2 \oplus a_1) \oplus a_3 = x_1 x_2 \oplus a_1 a_2 \oplus a_3.$$

Отсюда

$$x_1 x_2 = f(x_1 \oplus a_2, x_2 \oplus a_1, \alpha_3, \dots, \alpha_n) \oplus b,$$

где $b = a_1 a_2 \oplus a_3$. Следовательно, конъюнкция $x_1 x_2$ является суперпозицией функции f, констант и отрицания ($x \oplus a = x$ при $a = 0$ и $x \oplus a = \bar{x}$ при $a = 1$). ■

Для иллюстрации рассмотрим функцию, представленную полиномом

$$f(x_1, x_2, x_3, x_4) = \\ = 1 \oplus x_1 \oplus x_4 \oplus x_1 x_4 \oplus x_2 x_4 \oplus x_3 x_4 \oplus x_1 x_2 x_3 \oplus x_1 x_2 x_4 \oplus x_2 x_3 x_4 \oplus x_1 x_2 x_3 x_4.$$

Группируем слагаемые:

$$f(x_1, x_2, x_3, x_4) = \\ = x_1 x_2 (x_3 \oplus x_4 \oplus x_3 x_4) \oplus x_1 (1 \oplus x_4) \oplus x_2 (x_4 \oplus x_3 x_4) \oplus 1 \oplus x_4 \oplus x_3 x_4.$$

Подставляем $x_3 = 1$, $x_4 = 0$:

$$f(x_1, x_2, 1,0) = x_1x_2 \oplus x_1 \oplus 1.$$

Подставляем $x_2 \oplus 1 = \overline{x_2}$ вместо x_2:

$f(x_1, \overline{x_2}, 1,0) = x_1x_2 \oplus 1$. Отсюда

$x_1x_2 = f(x_1, \overline{x_2}, 1,0) \oplus 1 = \overline{f(x_1, \overline{x_2}, 1,0)}$.

7.3. Критерий полноты

Теорема 7.6 (теорема Поста, критерий полноты). *Множество функций является полной системой тогда и только тогда, когда оно не включено ни в один из классов* T_0, T_1, M, S, L.

Доказательство. Пусть A – множество функций. Допустим, что $A \subseteq X$, где X – один из пяти классов. По свойствам замыкания $[A] \subseteq [X]$. Так как X – замкнутый класс, то $[X] = X$ и, следовательно, $[A] \subseteq X$. Ни один из пяти классов не содержит всех логических функций, значит, $X \neq P_2$. Следовательно, $[A] \neq P_2$ и множество A не является полной системой.

Докажем обратное утверждение. Допустим, множество A не является подмножеством ни одного из пяти классов. Тогда в нем имеются такие функции f_1, f_2, f_3, f_4, f_5, что $f_1 \notin T_0$, $f_2 \notin T_1$, $f_3 \notin M$, $f_4 \notin S$, $f_5 \notin L$ (это не обязательно пять разных функций, среди них могут быть и одинаковые).

Рассмотрим функцию f_1. Она не принадлежит классу T_0, поэтому $f_1(0,0,\dots,0) = 1$. Далее рассматриваем два варианта в зависимости от значения $f_1(1,1,\dots,1)$.

а) $f_1(1,1,\dots,1) = 0$. Тогда $f_1(x, x, \dots, x) = \bar{x}$, то есть отрицание получается отождествлением всех переменных у функции f_1. По лемме о несамодвойственной функции одна из констант является суперпозицией функций $\bar{x}$ и f_4. Пусть это константа c. Подставляя ее в отрицание, получим вторую константу, $\bar{c}$. Имея две константы, отрицание и нелинейную функцию f_5, по лемме о нелинейной функции можем получить конъюнкцию. Таким образом, отрицание и конъюнкция являются суперпозициями функций из множества A. Но, как известно, множество $\{\bar{x}, x_1x_2\}$ является полной системой. По теореме сведения множество A – тоже полная система.

б) $f_1(1,1,\dots,1) = 1$. Тогда при отождествлении переменных у функции f_1 получается константа 1: $f_1(x, x, \dots, x) = 1$. Так как функция f_2 не принадлежит классу T_1, то $f_2(1,1,\dots,1) = 0$. Значит, подставляя 1 вместо каждой переменной в функцию f_2, получим константу 0. Имея две константы и немонотонную функцию f_3, по лемме о немонотонной функции можем получить отрицание. Далее, как и в случае а) применяем лемму о нелинейной функции, получаем конъюнкцию и на основании теоремы сведения опять приходим к выводу, что множество A является полной системой. ■

Заметим, что имеются несложные способы проверки принадлежности функции к каждому из пяти классов. Поэтому теорема Поста – это эффективно проверяемый критерий полноты. Результаты проверки на полноту обычно оформляют в виде таблицы, строки которой соответствуют исследуемым функциям, а столбцы – пяти классам из теоремы Поста. В соответствующей клетке таблицы ставится плюс или минус в зависимости от того, принадлежит ли данная функция данному классу. Система функций полная, если в каждом столбце таблицы есть хотя бы один минус.

Примеры.

1. Рассмотрим множество из двух функций $\{x_1 \to x_2, x_1 \oplus x_2\}$. Результаты исследования на принадлежность пяти классам выглядят так:

	T_0	T_1	M	S	L
$x_1 \to x_2$	−	+	−	−	−
$x_1 \oplus x_2$	+	−	−	−	+

Отсюда видно, что это полная система.

2. Рассмотрим функцию $\overline{x_1 x_2 \dots x_n}$. Нетрудно проверить, что при любом $n \geq 2$ эта функция не принадлежит ни одному из классов T_0, T_1, M, S, L. Следовательно, при любом $n \geq 2$ множество, состоящее из одной функции $\overline{x_1 x_2 \dots x_n}$ является полной системой.

7.4. Предполные классы и базисы

Пять классов, фигурирующих в теореме Поста, очень не похожи друг на друга (если не считать «близнецов» T_0 и T_1). Тем не менее, есть одно свойство, объединяющее их, из-за которого они и появились в этой теореме.

Множество функций называется *предполным классом*, если оно не полное, но при добавлении любой новой функции становится полным. Иначе говоря, предполный класс – это максимальное по отношению включения неполное множество.

Теорема 7.7. *Существует ровно 5 предполных классов:* T_0, T_1, M, S, L.

Доказательство. Нужно доказать, что каждый из этих классов – предполный и что других предполных классов нет. Первое утверждение мы докажем для одного из пяти классов, для остальных рассуждения аналогичные.

Возьмем класс M. Он не является полным, так как $[M] = M \neq P_2$. Возьмем какую-нибудь функцию $f \notin M$ и рассмотрим множество $M \cup \{f\}$. Если оно не полное, то по теореме Поста оно должно быть подмножеством одного из классов T_0, T_1, S, L. Но тогда и M будет подмножеством этого класса. А это не так: $M \nsubseteq T_0$, так как $1 \in M - T_0$; $M \nsubseteq T_1$, так как $0 \in M - T_1$; $M \nsubseteq S$, так как $0 \in M - S$; $M \nsubseteq L$, так как $x_1 x_2 \in M - L$. Значит, множество $M \cup \{f\}$ не является полным для любой функции $f \notin M$, следовательно, M – предполный класс.

Докажем, что других предполных классов нет. Допустим, X – предполный класс, отличный от T_0, T_1, M, S, L. По определению предполного класса множество X не является полным, значит, по теореме Поста оно включено в один из пяти классов. Допустим для определенности, что $X \subseteq M$ (в остальных случаях доказательство точно такое же). Так как $X \neq M$, то существует функция f, принадлежащая M, но не принадлежащая X. Но тогда $X \cup \{f\} \subseteq M$ и по теореме Поста множество X не является полным, а это противоречит тому, что X – предполный класс. ■

Базисом называется минимальная по включению полная система функций. Иначе говоря, это такая полная система, которая перестает быть полной после удаления из нее любой из функций.

Рассмотрим примеры базисов.

1. Любая полная система, состоящая из одной функции, конечно является базисом. Мы видели, что существует бесконечно много таких базисов: $\{\overline{x_1 x_2 \dots x_n}\}$ при любом $n \geq 2$.

2. Нам известен также пример базиса, состоящего из двух функций: $\{\bar{x}, x_1 x_2\}$. Это действительно базис, так как ни отрицание, ни конъюнкция в одиночку не образуют полных систем (отрицание – линейная функция, конъюнкция – монотонная). Другой пример базиса из двух функций – множество $\{\bar{x}, x_1 \vee x_2\}$.

3. Пример базиса, состоящего из трех функций – $\{1, x_1 x_2, x_1 \oplus x_2\}$. Мы видели, что это полная система. Если удалить из нее 1, то обе оставшиеся функции принадлежат классу T_0, значит, не образуют полной системы. Если удалить конъюнкцию, то останутся две линейные функции, а если удалить сумму по модулю 2, то останутся две монотонные функции.

Итак, существуют базисы из разного число функций. Возникает вопрос – как велика может быть мощность базиса?

Теорема 7.8. *Каждый базис содержит не более четырех функций.*

Доказательство. Покажем, что в каждой полной системе содержится полная подсистема не более чем из четырех функций. По теореме Поста в любой полной системе имеются функции $f_1 \notin T_0$, $f_2 \notin T_1$, $f_3 \notin M$, $f_4 \notin S$,

$f_5 \notin L$. Множество $\{f_1, f_2, f_3, f_4, f_5\}$ само является полной системой. Рассмотрим функцию f_1 . Она не принадлежит классу T_0 , поэтому $f_1(0,0,\dots,0) = 1$. Если $f_1(1,1,\dots,1) = 1$, то f_1 – не самодвойственная функция (на противоположных наборах – одинаковые значения). Тогда множество $\{f_1, f_2, f_3, f_5\}$ является полной системой. Если же $f_1(1,1,\dots,1) = 0$, то f_1 – не монотонная, тогда $\{f_1, f_2, f_4, f_5\}$ – полная система. ■

Примером базиса из четырех функций может служить множество $(0,1, x_1x_2,\ x_1 \oplus x_2 \oplus x_3\}$. Сведения о принадлежности этих функций предполным классам показаны в таблице:

	T_0	T_1	M	S	L
0	+	−	+	−	+
1	−	+	+	−	+
x_1x_2	+	+	+	−	−
$x_1 \oplus x_2 \oplus x_3$	+	+	−	+	+

Видно, что это полная система. Легко проверить, что при удалении любой функции оставшиеся три все содержатся в одном из предполных классов. Значит, это множество – базис.

7.5. Дополнительные сведения

Пост доказал, что существует счетное множество замкнутых классов логических функций и дал их полное описание. Он также доказал, что любой замкнутый класс имеет конечный базис, т.е. минимальное множество функций, замыкание которого совпадает с этим классом (например, множество $\{0,1, xy, x \vee y\}$ является базисом класса монотонных функций, а множество $\{0, x \sim y\}$ – базисом класса линейных функций). Подробное изложение теории логических функций содержится в книге

Яблонский С.В., Гаврилов Г.П., Кудрявцев В.Б. Функции алгебры логики и классы Поста.– М.: Наука, 1966.

Обобщением двузначных функций являются k–значные функции или функции k-значной логики. Это такие функции, у которых каждая переменная и сама функция принимает значения в множестве $\{0,1,\dots,k-1\}$.

Некоторые свойства множества P_k всех функций k-значной логики при $k \geq 3$ существенно отличаются от свойств множества P_2. Так, при $k \geq 3$ в P_k

- существует континуальное множество замкнутых классов;
- существуют замкнутые классы, имеющие только бесконечные базисы;
- существуют замкнутые классы, не имеющие базиса.

Подробное изложение этих и других фактов из k–значной логики, а также и теории двузначных функций можно найти в учебнике по дискретной математике

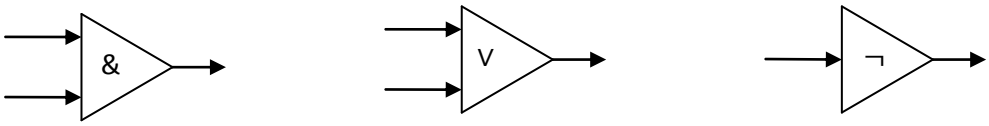
Яблонский С.В. Введение в дискретную математику.– М.: Высшая школа, 2008.

Глава 8. Схемы

8.1. Схемы из функциональных элементов

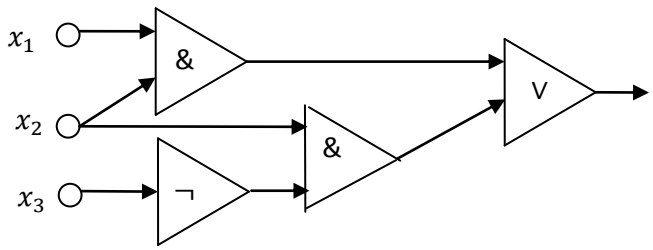
Схемы – еще один способ представления логических функций наряду с таблицами и формулами. Схемы можно рассматривать и как способ описания алгоритмов вычисления функций и как математические модели физических устройств. Известно несколько типов абстрактных схем, мы рассмотрим один из наиболее распространенных – схемы из функциональных элементов.

Схема, как и формула, описывает порядок выполнения операций для вычисления значений некоторой функции. Только формула – это строка символов, а схема – графический объект. Схемы составляются из *функциональных элементов*, вычисляющих некоторые «простейшие» функции и соединяемых по определенным правилам. Мы будем рассматривать схемы, состоящие из функциональных элементов трех типов: конъюнкторов, дизъюнкторов и инверторов. Они вычисляют соответственно конъюнкцию, дизъюнкцию и отрицание. Эти элементы будем изображать на схеме в виде треугольников с вписанными в них обозначениями функций, а также входящими и выходящими стрелками:



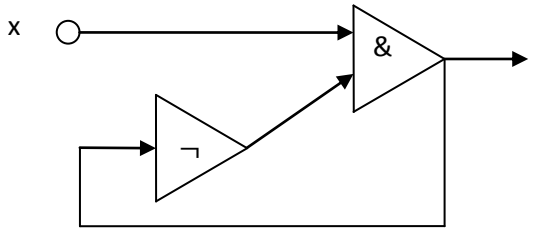
Входящие стрелки изображают *входы* элемента, на входы подаются значения аргументов – нули и единицы. Выходящая стрелка – *выход* элемента, на выходе появляется значение вычисляемой элементом функции. Например, если на один вход дизъюнктора подать значение 0, а на другой – значение 1, то на выходе будет значение 1, а если на оба входа подать значения 0, то и на выходе будет 0. В схеме у элемента может быть несколько выходящих стрелок, так как результат вычисления может использоваться несколько раз в дальнейших вычислениях.

Кроме элементов, на схеме присутствуют также *входы схемы*, представляющие аргументы вычисляемой данной схемой функции. Их будем изображать кружками, каждому из которых приписан символ переменной (разным входам приписываются разные переменные). На рисунке показан пример схемы:



Эта схема вычисляет функцию от переменных x_1, x_2, x_3 : сначала вычисляются x_1x_2 и $\overline{x_3}$, затем $x_2\overline{x_3}$ и, наконец, $x_1x_2 \vee x_2\overline{x_3}$. Это и есть функция, вычисляемая данной схемой.

Если мы хотим доказывать какие-то утверждения о схемах, то нужно определить понятие схемы как математического объекта. В основу такого определения естественно положить понятие ориентированного графа (сокращенно орграфа): вершины графа – это входы схемы и функциональные элементы, а связи между ними – ребра графа. Но если мы хотим, чтобы каждая схема представляла некоторую функцию, некоторые виды графов следует запретить. Рассмотрим, например, такую схему:



Какую функцию вычисляет эта схема? Если $x = 1$, то какое значение будет на выходе схемы? Если 1, то пройдя через инвертор, она превратится в 0, этот 0 будет подан на второй вход конъюнктора и тогда на выходе должен быть 0. Если же на выходе 0, то после инвертора он превратится в 1 и на выходе должна быть1. Таким образом, если на входе этой схемы 1, то значение на выходе не определено. Невозможно указать функцию, которую вычисляет эта схема. Причина, очевидно, в том, что информация с выхода конъюнктора некоторым образом (через инвертор) поступает на его вход. Это явление называется обратной связью и оно очень распространено и полезно во многих технических и других системах. Но в схемах рассматриваемого типа его необходимо запретить. На языке теории графов мы должны запретить ориентированные циклы.

Ориентированный цикл в орграфе – это такая последовательность вершин $x_1, x_2, \dots, x_k$, что каждая пара (x_i, x_{i+1}), $i = 1,2, \dots, k-1$, и пара (x_k, x_1) является ориентированным ребром. Орграф, не имеющий ориентированных циклов, называется *ациклическим.* Отметим одно свойство ациклических орграфов, которое будет полезно в дальнейшем. Нумерацию вершин орграфа назовем *монотонной*, если номер начальной вершины каждого ребра меньше номера его концевой вершины.

Лемма 8.1. *Для любого ациклического орграфа существует монотонная нумерация его вершин.*

Доказательство. Будем присваивать номера вершинам в порядке возрастания, начиная с 1. Допустим, номера 1,…, $i-1$ уже присвоены некоторым вершинами так, что эта частичная нумерация монотонна. Выберем какой-либо ориентированный путь P наибольшей длины, проходящий только через вершины, еще не имеющие номеров. Пусть a – начальная вершина этого пути, а b – вершина, из которой выходит ребро, ведущее в вершину a. Вершина b не может принадлежать пути P, так как в противном случае образовался бы цикл. Если b еще не имеет номера, то ее можно добавить ее в начале пути P и получится путь большей длины из непронумерованных вершин. Значит, все вершины, из которых выходят ребра в вершину a, уже имеют номера. Поэтому, если вершине a присвоить очередной номер i, то полученная частичная нумерация тоже будет монотонной. Продолжая действовать таким образом, в конце концов получим монотонную нумерацию всех вершин. ■

Теперь можно дать формальное определение схемы из функциональных элементов.

Схема из функциональных элементов – это ациклический орграф, содержащий вершины двух типов:

- Входные вершины. У входной вершины нет входящих ребер. Каждой входной вершине приписан символ переменной, причем разным входным вершинам – разные символы.
- Функциональные вершины. Каждой функциональной вершине приписан один из символов $\&$, $\vee$, $\neg$. У вершины, которой приписан символ $\&$, или $\vee$, имеется точно два входящих ребра, у вершины, которой приписан символ $\neg$ – точно одно.

Некоторые вершины схемы помечаются как выходные (их может быть несколько, т.е. схема может вычислять одновременно несколько функций).

Для каждой схемы из функциональных элементов можно определить функции, вычисляемые этой схемой. Так как граф схемы ациклический, то существует монотонная нумерация его вершин. Теперь для каждой вершины схемы можно определить функцию, вычисляемую в этой вершине. Это делается в порядке возрастания номеров вершин. Если очередная вершина является входной и ей приписана переменная x, то в ней вычисляется тождественная функция x. Если очередная вершина – функциональная и ей приписан символ $\&$, а в двух вершинах, из которых ведут ребра в данную вершину, вычисляются функции f и g, то в этой вершине вычисляется

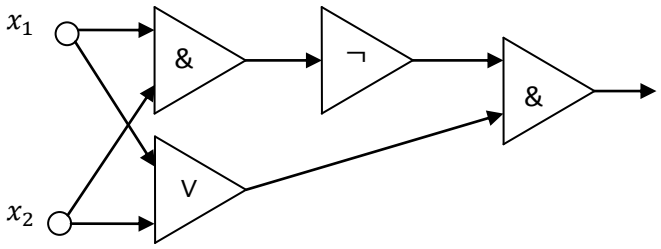
функция $f\&g$. Аналогично для дизъюнкции и отрицания. Функции, вычисляемые схемой – это функции, вычисляемые в ее выходных вершинах.

По существу, схема дает описание функции как суперпозиции трех функций: конъюнкции, дизъюнкции и отрицания. Так как эти функции образуют полную систему, то для любой булевой функции можно построить вычисляющую ее схему. С таким же успехом можно было за основу схем взять любой другой набор элементов, вычисляющих функции какой-нибудь полной системы. В частности, для любой функции можно построить вычисляющую ее схему, составленную только из элементов, вычисляющих штрих Шеффера.

Важной характеристикой схемы является число функциональных элементов в ней. Это число называется *сложностью* схемы. Наименьшая сложность схемы, вычисляющей функцию f , обозначается через $L(f)$ и называется *схемной сложностью* функции f . Рассмотрим, например, функцию $x_1 \oplus x_2$. Можно построить для нее схему, используя равенство $x_1 \oplus x_2 = \overline{x_1}x_2 \vee x_1\overline{x_2}$. Схема будет содержать 5 функциональных элементов. Но можно использовать другую булеву формулу для этой функции:

$$x_1 \oplus x_2 = (\overline{x_1} \vee \overline{x_2})(x_1 \vee x_2) = (\overline{x_1 x_2})(x_1 \vee x_2).$$

В схеме будет уже только 4 элемента:

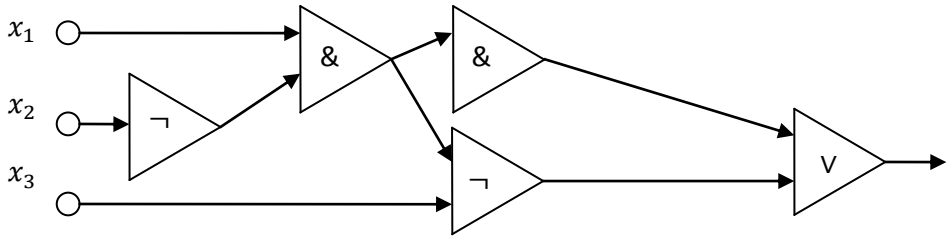


Можно доказать, что меньшего числа элементов для этой функции недостаточно (например, перебрать все схемы, содержащие не более трех функциональных элементов, и убедиться, что ни одна из них не вычисляет данную функцию). Таким образом, $L(x_1 \oplus x_2) = 4$.

Наибольшая схемная сложность функции от n переменных обозначается через $L(n)$. Определим, например, $L(1)$. Имеется четыре функции одной переменной: 0, 1, x, $\bar{x}$. Схема для функции x не содержит ни одного функционального элемента. Схема для отрицания содержит единственный функциональный элемент, а схемы для констант можно построить на основании формул $0 = x\bar{x}$, $1 = x \vee \bar{x}$, каждая из них будет содержать два элемента. Таким образом, $L(1) = 2$.

8.2. Построение схем

Мы уже видели, что можно строить схемы, используя в качестве исходного материала формулы. Следует, однако, отметить одно существенное различие между схемами и формулами. В формуле одно и то же выражение может встречаться несколько раз как часть этой формулы («подформула»). В схеме же, вычислив некоторое выражение, можно потом использовать сколько угодно раз, не вычисляя повторно. Рассмотрим, например, формулу $x_1\overline{x_2} \vee \overline{x_1\overline{x_2}}\, x_3$. В ней 7 операций. Но строя схему, можно вычислить выражение $x_1\overline{x_2}$ только один раз и использовать дважды:



В схеме только 5 операций. Для функций от многих переменных выигрыш схем перед формулами в числе операций может быть значительным.

Рассмотрим два простейших метода построения схем.

1. Использование нормальных форм.

Если функция задана таблицей, можно построить по ней СДНФ, а по ней – схему. Оценим число элементов в такой схеме для функции от n переменных. Допустим, что СДНФ состоит из s слагаемых (т.е. функция принимает значение 1 на s наборах).

Отрицание каждой переменной достаточно вычислить один раз. Это потребует n инверторов. Для вычисления одной простой конъюнкции необходимо $n-1$ конъюнкторов. Всего, следовательно, потребуется $s(n-1)$ конъюнкторов. Для того чтобы сложить все конъюнкции, нужно $s-1$ дизъюнкторов. Всего получается $n(s+1)-1$ элементов. Обозначим через $L_1(n)$ наибольшее число элементов в схеме, построенной этим способом. Для функции от n переменных $s \le 2^n$. Равенство $s = 2^n$ достигается только, когда функция – константа 1. Но для константы 1, как мы видели, можно построить схему из двух элементов. Поэтому ее можно исключить из рассмотрения. Для всех остальных функций выполняется неравенство $s \le 2^n - 1$ и мы имеем

$$L_1(n) \le n2^n - 1.$$

Аналогично можно строить схему по СКНФ. Число элементов в ней тем больше, чем больше нулей среди значений функции. Поэтому имеет смысл скомбинировать эти два способа – если в таблице функции единиц

меньше, чем нулей, то использовать СДНФ, в противном случае – СКНФ. Это позволяет уменьшить сложность схемы в наихудшем случае вдвое по сравнению с приведенной оценкой.

2. Применение теоремы о разложении.

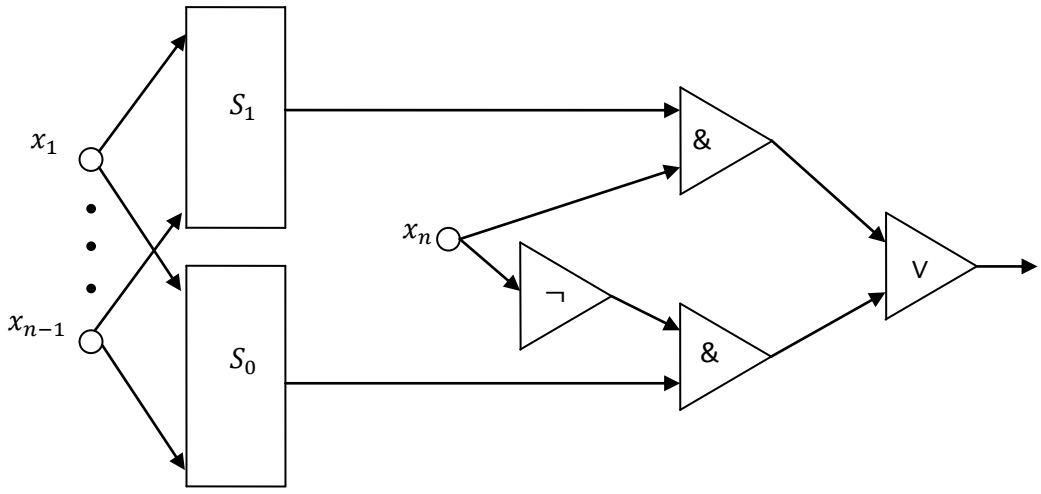
Теорема 8.1 (теорема о разложении функции по переменной). *Для любой логической функции* $f(x_1, \dots, x_n)$ *справедливо тождество*

$$f(x_1, \dots x_{n-1}, x_n) = x_n f(x_1, \dots, x_{n-1}, 1) \vee \overline{x_n} f(x_1, \dots, x_{n-1}, 0).$$

Утверждение теоремы легко проверяется подстановкой вместо переменной x_n значений 0 и 1.

Формула в правой части равенства называется разложением функции f по переменной x_n (конечно, функцию можно разложить по любой переменной). Это разложение выражает функцию от n переменных через две функции от $n-1$ переменной и элементарные функции. Его можно использовать для построения схем следующим образом.

Допустим, нужно построить схему для функции $f(x_1, \dots, x_n)$. Разложим ее по переменной x_n. Каждую из двух получившихся функций от $n-1$ переменной разложим по переменной x_{n-1}. Получим четыре функции от $n-2$ переменных, каждую из которых разложим по x_{n-2} и т.д., пока не дойдем до функций одной переменной. Для всех функций одной переменной мы умеем строить схемы. Реализуем эти функции схемами, из них соберем схемы для функций от двух переменных в соответствии с теоремой о разложении, затем для функций от трех переменных и т.д., пока не дойдем до исходной функции. На последнем шаге происходит следующее. Мы уже построили схемы для функций $f(x_1, \dots, x_{n-1}, 1)$ и $f(x_1, \dots, x_{n-1}, 0)$. Объединяем эти схемы в одну (у них общие входы), добавляем вход x_n, и добавляем элементы, реализующие формулу в правой части теоремы о разложении. Это выглядит так:



Здесь S_1 и S_0 – схемы, вычисляющие соответственно функции $f(x_1, \dots, x_{n-1}, 1)$ и $f(x_1, \dots, x_{n-1}, 0)$.

Обозначим через $L_2(n)$ наибольшее число элементов в схеме, построенной этим способом для функции от n переменных. Очевидно, «худшая» функция (с наибольшим числом элементов в схеме) от n переменных – это та, у которой $f(x_1, \dots, x_{n-1}, 1)$ и $f(x_1, \dots, x_{n-1}, 0)$ – «худшие» функции от $n-1$ переменной. Схема для функции от n переменных состоит из двух схем для функций от $n-1$ переменной и еще четырех элементов. Получаем рекуррентное уравнение

$$L_2(n) = 2L_2(n-1) + 4.$$

У нас есть также начальное значение $L_2(1) = 2$. Решая, находим

$$L_2(n) = 3 \cdot 2^n - 4.$$

8.3. Сумматор

Построим схему, вычисляющую сумму двух целых неотрицательных чисел, представленных n–разрядными двоичными записями.

Пусть $x_n x_{n-1} \dots x_1$ и $y_n y_{n-1} \dots y_1$ – двоичные записи двух слагаемых (старшие разряды, как обычно, слева). Сумма может иметь на один разряд больше, пусть $z_{n+1} z_n \dots z_1$ – двоичная запись суммы.

Мы рассматриваем обычный алгоритм сложения «столбиком». При этом могут возникать переносы из младших разрядов в старшие. Введем переменные u_i, $i = 1, \dots, n$, описывающие переносы: $u_i = 1$, если при сложении i–тых разрядов слагаемых образуется перенос в следующий разряд, в противном случае $u_i = 0$. Заметим, что u_n – это старший разряд суммы: $u_n = z_{n+1}$. Очередной разряд суммы зависит от соответствующих разрядов слагаемых и от того, был ли перенос из предыдущего разряда. От этих же аргументов зависит появление переноса в следующий разряд. Таблица этих двух функций и есть главная часть описания алгоритма сложения. Вот эта таблица:

x_i	y_i	u_{i-1}	z_i	u_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Теперь представим обе функции формулами. Можно, например, так:

$z_i = x_i \oplus y_i \oplus u_{i-1}$,

$u_i = x_i y_i \vee x_i u_{i-1} \vee y_i u_{i-1}$.

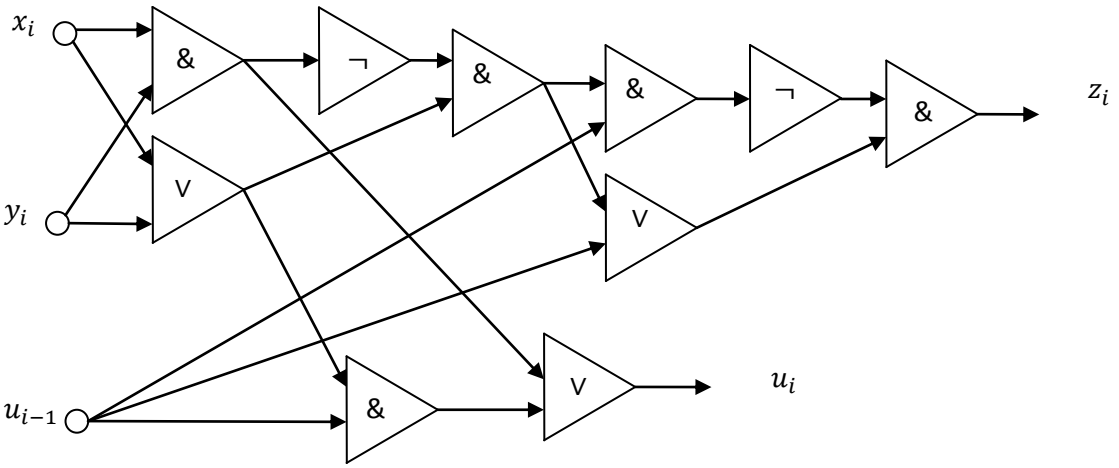
Для вычисления суммы по модулю 2 используем приведенную выше схему, основанную на тождестве

$x \oplus y = (x \vee y)\overline{xy}$.

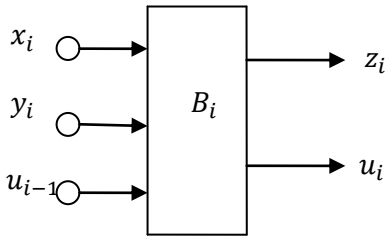
Вторую формулу слегка преобразуем с целью сэкономить одну операцию:

$u_i = x_i y_i \vee (x_i \vee y_i) u_{i-1}$.

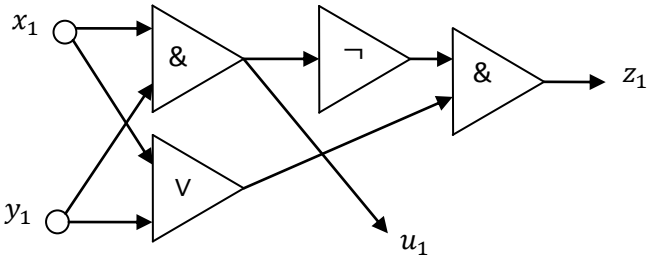
(теперь функцию $x_i \vee y_i$ можно вычислить один раз, а результат использовать дважды). Теперь можно построить схему, вычисляющую обе функции:



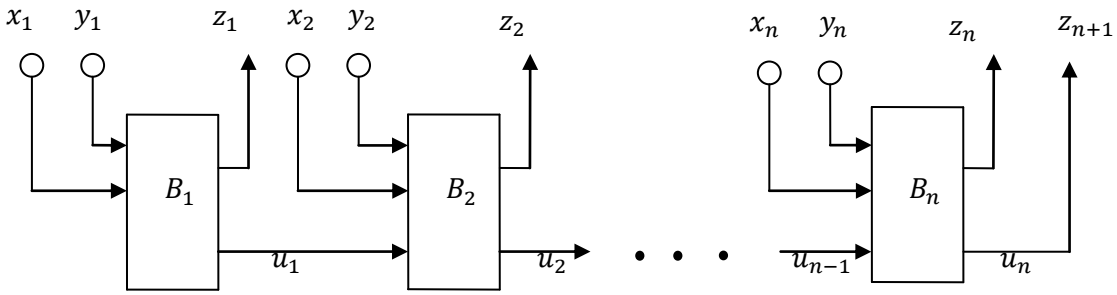
Эта схема – блок, вычисляющий один разряд суммы и разряд переноса. Обозначим его B_i, а на схеме будем изображать в виде прямоугольника:



Блок B_1 устроен несколько проще, так как отсутствует перенос из предыдущего разряда:



Из этих блоков собирается вся схема сумматора:



8.4. Дополнительные сведения

Основы теории логических схем были заложены в работах одного из основоположников кибернетики К. Шеннона. В частности, он доказал нижнюю границу схемной сложности самых сложных функций, т.е. функции $L(n)$ (ее называют функцией Шеннона). Метод построения схем, асимптотически достигающий этой границы предложил О.Б. Лупанов (асимптотически – это значит, что отношение сложности схемы, построенной методом Лупанова, к нижней границе стремится к 1 при $n \to \infty$). Вместе эти результаты дают асимптотическое значение функции Шеннона:

$L(n) = \frac{2^n}{n}(1 + \varepsilon_n)$, где $\varepsilon_n \to 0$ при $n \to \infty$.

Подробности можно найти в учебнике

Яблонский С.В. Введение в дискретную математику.– М.: Высшая школа, 2008.

Глава 9. Кодирование

Кодирование – преобразование информации, выполняемое с разнообразными целями: экономное представление (сжатие данных), защита от помех (помехоустойчивое кодирование), защита от чтения посторонними (криптография). Для решения многих задач кодирования используются математические методы. Здесь будет рассмотрена одна классическая задача кодирования, связанная со сжатием данных. Метод Хаффмана, дающий решение этой задачи, был разработан в середине XX в. и до сих пор применяется во многих системах сжатия данных (в сочетании с другими средствами).

9.1. Задача оптимального кодирования

Начнем с примера. Допустим, имеется текст, составленный из букв алфавита $A = \{a, b, c, d\}$, и его необходимо перекодировать в алфавит $B = \{0,1\}$. Очевидный способ это сделать – заменить каждую букву алфавита A двухбуквенным словом в алфавите B, например, по схеме:

$$(1)\ \begin{cases} a \to 00 \\ b \to 01 \\ c \to 10 \\ d \to 11 \end{cases}$$

При таком кодировании слово cba, например, преобразуется в слово 100100. Ясно, что в результате такого кодирования длина текста удваивается. Теперь допустим, что буквы алфавита A в кодируемом тексте встречаются неравномерно – половину общего количества букв в нем составляет буква a, четвертую часть – буква b и по одной восьмой – каждая из букв c и d. Нельзя ли это использовать для более экономного кодирования, заменяя часто встречающиеся буквы более короткими словами, а реже встречающиеся – более длинными. Можно попробовать, например такую систему кодирования:

$$(2)\ \begin{cases} a \to 0 \\ b \to 10 \\ c \to 110 \\ d \to 111 \end{cases}$$

Если кодируется текст в 10000 символов, среди которых 5000 букв a, 2500 букв b и по 1250 букв c и d, то при использовании системы (1) длина результирующего текста будет 20000 символов, а при использовании системы (2) – 17500 символов. Выигрыш не такой уж большой, но цель этого

примера – показать, что выигрыш вообще возможен. Величина его зависит от конкретных обстоятельств и иногда может быть значительной.

Перейдем к общему случаю. Пусть $A = \{a_1, a_2, \dots, a_k\}$ и $B = \{b_1, b_2, \dots, b_q\}$ – два алфавита. *Побуквенное кодирование* состоит в том, что в кодируемом тексте (слове в алфавите A) каждая буква алфавита A заменяется словом в алфавите B. Эти замены производятся в соответствии со *схемой кодирования*, указывающим для каждой буквы слово, которым она заменяется:

$$\begin{cases} a_1 \to u_1 \\ a_2 \to u_2 \\ \quad \vdots \\ a_k \to u_k \end{cases}$$

Здесь $u_1, u_2, \dots, u_n$ – слова в алфавите B. Таким образом, кодирование есть функция, преобразующая слово $\alpha = a_{i_1} a_{i_2} \dots a_{i_n}$ в слово $f(\alpha) = u_{i_1} u_{i_2} \dots u_{i_n}$. Кодируемое слово α будем называть *сообщением*, а результат кодирования $f(\alpha)$ – *кодограммой*.

Считая алфавит A линейно упорядоченным множеством (буква a_1 предшествует букве a_2 и т.д.), мы можем задать схему кодирования просто перечислением слов $u_1, u_2, \dots, u_k$, располагая их в том же порядке, в котором располагаются в алфавите кодируемые ими буквы. Упорядоченный набор слов $U = (u_1, u_2, \dots, u_k)$ будем называть *кодом*, а его элементы – *кодовыми словами*.

Экономность кодирования характеризуют *коэффициентом сжатия*, который равен отношению длины кодограммы к длине сообщения. Допустим, известно, что буква a_1 входит в кодируемый текст n_1 раз, буква a_2 – n_2 раза, буква a_k – n_k раз, а общая длина текста равна $n = n_1 + n_2 + \dots + n_k$. Длину слова α будем обозначать через $|\alpha|$. Тогда длина кодограммы будет равна $n_1|u_1| + n_2|u_2| + \dots + n_k|u_k|$ и коэффициент сжатия равен

$$\frac{|f(\alpha)|}{|\alpha|} = \frac{n_1}{n}|u_1| + \frac{n_2}{n}|u_2| + \dots + \frac{n_k}{n}|u_k|.$$

Числа $p_1 = \frac{n_1}{n}$, $p_2 = \frac{n_2}{n}$, ..., $p_k = \frac{n_k}{n}$ называются *частотами* соответствующих букв. Будем предполагать, что они известны, т.е. задан *набор частот* $P = (p_1, p_2, \dots, p_k)$. Коэффициент сжатия кода U при данном наборе частот P обозначим через $C(P, U)$, он равен

$$C(P, U) = \sum_{i=1}^{k} p_i |u_i|.$$

В рассмотренном выше примере частоты букв равны $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}$, коэффициент сжатия кодирования по схеме (1) равен 2, а коэффициент сжатия кодирования по схеме (2) равен 1,75. Нельзя ли придумать для этого примера кодирование с меньшим коэффициентом сжатия? Почему бы не использовать, например, такую схему:

$$\begin{cases} a \to 0 \\ b \to 01 \\ c \to 10 \\ d \to 11. \end{cases}$$

При таком кодировании коэффициент сжатия был бы равен $\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 2 + \frac{1}{8} \cdot 2 = 1{,}5$.

Но при применении такого кодирования возникает проблема из-за того, что, например, сообщения ba и ac имеют одинаковые кодограммы: $f(ba) = f(ac) = 010$. Таким образом, кодограмму 010 невозможно однозначно декодировать, т.е. восстановить исходное сообщение. Понятно, что коды, допускающие такую неоднозначность, надо исключить из рассмотрения.

В общем случае для того, чтобы любую кодограмму можно было однозначно декодировать, функция f должна быть инъективной, т.е. для любых двух различных слов α и β в алфавите A слова $f(\alpha)$ и $f(\beta)$ тоже должны быть различными. Код, для которого это выполняется, называется *обратимым*.

Допустим, функция f не инъективна. Тогда найдутся такие два слова $\alpha = a_{i_1} a_{i_2} \dots a_{i_n}$ и $\beta = a_{j_1} a_{j_2} \dots a_{j_m}$, что $\alpha \neq \beta$ (т.е. наборы $(i_1, i_2, \dots, i_n)$ и $(j_1, j_2, \dots, j_m)$ различны), а $f(\alpha) = f(\beta)$, т.е. $u_{i_1} u_{i_2} \dots u_{i_n} = u_{j_1} u_{j_2} \dots u_{j_m}$. Поэтому определение обратимости равносильно следующему: код $U = (u_1, u_2, \dots, u_k)$ обратим, если равенство $u_{i_1} u_{i_2} \dots u_{i_n} = u_{j_1} u_{j_2} \dots u_{j_m}$ имеет место тогда и только тогда, когда $(i_1, i_2, \dots, i_n) = (j_1, j_2, \dots, j_m)$ Иначе это требование можно сформулировать так: каждое слово в алфавите B не более чем одним способом может быть представлено как соединение кодовых слов.

Теперь можно сформулировать задачу построения оптимального кода.

Задача оптимального кодирования. Даны алфавиты $A = \{a_1, a_2, \dots, a_k\}$ и $B = \{b_1, b_2, \dots, b_q\}$ и набор частот $P = (p_1, p_2, \dots, p_k)$ букв алфавита A. Требуется найти обратимый код $U = (u_1, u_2, \dots, u_k)$ с наименьшим коэффициентом сжатия $C(P, U)$.

Решение этой задачи затрудняется тем, что проверка обратимости кода является не таким простым делом. Далее будет показано, что задачу можно свести к поиску оптимального кода специального вида, для которого свойство обратимости заведомо выполняется.

9.2. Префиксные коды

Код, в котором все слова имеют одинаковую длину, очевидно, обратим. Но и код, состоящий из слов разной длины, может быть обратимым. Таков, например, рассмотренный выше код (2). Пусть $f(\alpha)$ – кодограмма сообщения α, закодированного этим кодом. Если $f(\alpha)$ начинается с буквы 0, то первой буквой сообщения α может быть только буква a. Если $f(\alpha)$ начинается с 1, а за ней следует 0, то первой буквой в α является буква b. Если же $f(\alpha)$ начинается с двух 1, то в зависимости от третьего символа первая буква в α однозначно определяется как c или d. Таким образом, в любой кодограмме, полученной с помощью этого кода, всегда можно однозначно отделить первое кодовое слово. Поэтому и всю кодограмму можно однозначно разделить на кодовые слова. Однозначность декодирования (как и его простота) является следствием одного свойства этого кода – в нем ни одно из слов не является префиксом (начальным отрезком) никакого другого слова из этого кода.

Код, в котором ни одно из кодовых слов не является префиксом другого кодового слова .называется *префиксным кодом*.

Теорема 9.1. *Всякий префиксный код является обратимым.*

Доказательство. Если код $U = (u_1, u_2, \dots, u_k)$ не обратим, то существуют такие наборы $(i_1, i_2, \dots, i_n)$ и $(j_1, j_2, \dots, j_m)$, что $(i_1, i_2, \dots, i_n) \neq (j_1, j_2, \dots, j_m)$ и $u_{i_1} u_{i_2} \dots u_{i_n} = u_{j_1} u_{j_2} \dots u_{j_m}$. Найдем наименьшее s, при котором $i_s \neq j_s$. Тогда одно из слов u_{i_s} и u_{j_s} является началом другого, следовательно, код U не префиксный. ■

Не всякий обратимый код является префиксным. Например, код $U = (0, 01, 11)$ – не префиксный, но обратимый. Действительно, если кодограмма заканчивается буквой 0, то последняя буква сообщения, очевидно, a_1. Если же последней буквой кодограммы является 1, то последняя буква сообщения – a_2 или a_3 в зависимости от предпоследней буквы кодограммы.

Последний пример показывает одно преимущество префиксных кодов перед произвольными обратимыми кодами. Кодограмма, закодированная префиксным кодом, может начинаться только одним из кодовых слов. Легко найти это слово, отделить его и затем аналогично декодировать оставшуюся часть кодограммы. Для произвольных обратимых кодов декодирование может быть не таким простым. Например, допустим, используется код $U = (0, 01, 11)$, а кодограмма имеет вид 01… 1, где за первым нулем следуют n единиц. В этом случае первой буквой сообщения является a_1, если n четное, и a_2, если n нечетное. Следовательно, для восстановления первой буквы сообщения при таком способе кодирования может возникнуть

необходимость прочитать всю кодограмму. Кодограмму же, закодированную префиксным кодом, можно начинать декодировать уже в тот момент, когда прочитано первое составляющее ее кодовое слово. Уже это преимущество служит достаточным основанием для того, чтобы рассматривать только префиксные коды. Но оказывается, что и с точки зрения экономности кодирования, характеризуемой коэффициентом сжатия, префиксные коды не уступают произвольным обратимым кодам.

Следующие две теоремы позволяют свести задачу построения оптимального обратимого кода к задаче построения оптимального префиксного кода.

Теорема 9.2. (неравенство Макмиллана). *Если* $U = (u_1, u_2, \dots, u_k)$ – *обратимый код в алфавите из* q *букв, то выполняется неравенство*

$$\sum_{i=1}^{k} q^{-|u_i|} \leq 1.$$

Доказательство. Пусть функция f есть побуквенное кодирование, определяемое кодом U. Рассмотрим величину

$$S_n = \sum_{\alpha \in A^n} q^{-|f(\alpha)|}.$$

Очевидно,

$$S_1 = \sum_{i=1}^{k} q^{-|u_i|}$$

и мы должны доказать, что для обратимого кода выполняется неравенство $S_1 \leq 1$. При произвольном n имеем

$$S_n = \sum_{a_{i_1} a_{i_2} \dots a_{i_n}} q^{-|u_{i_1} u_{i_2} \dots u_{i_n}|} = \sum_{i_1=1}^{k} \sum_{i_2=1}^{k} \dots \sum_{i_n=1}^{k} q^{-|u_{i_1}|} q^{-|u_{i_2}|} \dots q^{-|u_{i_n}|} =$$

$$= \left(\sum_{i_1=1}^{k} q^{-|u_{i_1}|} \right) \left(\sum_{i_2=1}^{k} q^{-|u_{i_2}|} \right) \dots \left(\sum_{i_n=1}^{k} q^{-|u_{i_n}|} \right) = S_1^n.$$

Обозначим через $m(n, l)$ число сообщений длины n, имеющих кодограмму длины l. Группируя в сумме S_n слагаемые с одинаковым показателем степени, получаем

$$S_n = \sum_{l=1}^{nL} m(n, l)\, q^{-l},$$

где L – максимальная из длин слов кода U. Так как код U обратимый, то кодограммы различных сообщений различны, поэтому $m(n,l)$ не превосходит числа всех слов длины l в алфавите из q букв, т.е. $m(n,l) \le q^l$. Следовательно, каждое слагаемое в этой сумме не превосходит 1 и $S_n \le nL$. Учитывая, что $S_n = S_1^n$, получаем, что при любом n выполняется неравенство

$$S_1 \le (nL)^{\frac{1}{n}}.$$

Оно будет верно и в пределе при $n \to \infty$, а предел правой части равен 1. ■

Отметим, что неравенство Макмиллана является только необходимым, но не достаточным условием того, что код обратимый. Оно может выполняться и для кодов, не являющихся обратимыми. Например, оно выполняется для кода (0, 01, 10), который не является обратимым.

Теорема 9.3 (о существовании префиксного кода). *Пусть* $l_1, l_2, \dots, l_k$ *– натуральные числа, удовлетворяющие неравенству*

$$\sum_{i=1}^{k} q^{-l_i} \le 1.$$

Тогда существует префиксный код $U = (u_1, u_2, \dots, u_k)$ *в алфавите из* q *букв, в котором* $|u_i| = l_i,\ i = 1, \dots, k$.

Доказательство. Не теряя общности, можно предположить, что $l_1 \le l_2 \le \cdots \le l_k$. Будем строить код U, последовательно добавляя слова с длинами $l_1, l_2, \dots, l_k$ и следя за тем, чтобы после добавления очередного слова код оставался префиксным.

В качестве u_1 берем любое слово длины l_1 в алфавите $B = \{b_1, b_2, \dots, b_q\}$ и образуем код $U_1 = (u_1)$. Он, очевидно, префиксный. Допустим, уже построен префиксный код $U_{i-1} = (u_1, \dots, u_{i-1})$ с длинами слов $l_1, l_2, \dots, l_{i-1}$. Составим список всех слов длины l_i в алфавите B, этот список состоит из q^{l_i} слов. Вычеркнем из списка все слова, у которых префикс принадлежит коду U_{i-1}. Так как U_{i-1} – префиксный код, то каждое слово будет вычеркнуто не более одного раза. Число слов с префиксом u_j в этом списке равно $q^{l_i - l_j}$, значит, всего будет вычеркнуто $q^{l_i - l_1} + \cdots + q^{l_i - l_{i-1}}$ слов. Покажем, что после этого список будет непустым, т.е. $q^{l_i} - q^{l_i - l_1} - \cdots - q^{l_i - l_{i-1}} \ge 1$. Действительно, это неравенство равносильно $q^{-l_1} + \cdots + q^{-l_{i-1}} + q^{-l_i} \le 1$, а последнее следует из условия теоремы. Таким образом, список непустой и мы можем взять из него любое слово и добавить его к коду U_{i-1}. Новый код U_i тоже будет префиксным. ■

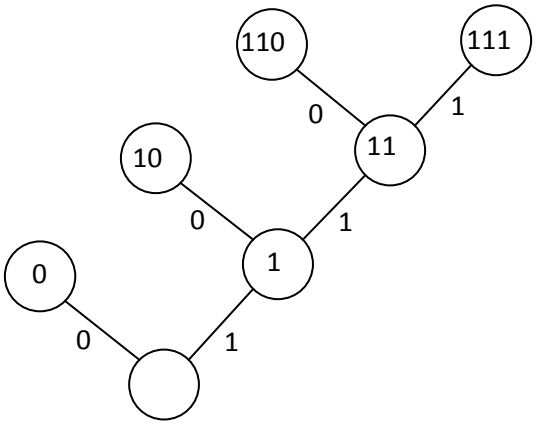
Следствие. Для любого набора частот $P = (p_1, p_2, \dots, p_k)$ и любого обратимого кода $U = (u_1, u_2, \dots, u_k)$ существует такой префиксный код $V = (v_1, v_2, \dots, v_k)$ в том же алфавите, что $C(P,V) = C(P,U)$.

Доказательство. Так как код U обратимый, то для него выполняется неравенство Макмиллана $\sum_{i=1}^{k} q^{-|u_i|} \leq 1$. По теореме 9.3 существует префиксный код $V = (v_1, v_2, \dots, v_k)$ такой, что $|v_i| = |u_i|$ для $i = 1, \dots, k$. Но тогда $\sum_{i=1}^{k} p_i|v_i| = \sum_{i=1}^{k} p_i|u_i|$, т.е. $C(P, V) = C(P, U)$. ■

Таким образом, при любом наборе частот P среди обратимых кодов с наименьшим коэффициентом сжатия имеется хотя бы один префиксный код. Поэтому при решении задачи построения оптимального кода можно ограничиться рассмотрением префиксных кодов.

Существует простое графическое представление префиксных кодов. Пусть алфавит B состоит из двух букв 0 и 1. Рассмотрим бесконечное корневое бинарное дерево. В этом дереве у каждой вершины имеется ровно два сына, один из которых считается левым, другой – правым, так же классифицируются ребра, ведущие от отцовской вершины к сыновним. Каждому левому ребру приписывается буква 0, каждому правому – буква 1. Двигаясь вдоль какого-нибудь пути в этом дереве, можно на ребрах прочитать некоторое слово в алфавите $\{0,1\}$. Каждой вершине дерева поставим в соответствие слово, читающееся вдоль пути из корня в эту вершину (самому корню соответствует пустое слово). Получается взаимно однозначное соответствие между вершинами дерева и словами в алфавите $\{0,1\}$. Заметим, что если слово α является префиксом слова β, то в дереве вершина, соответствующая α , лежит на пути из корня в вершину, соответствующую β.

Пусть $U = (u_1, \dots, u_k)$ – префиксный код. Рассмотрим фрагмент бесконечного бинарного дерева, состоящий из вершин, соответствующих словам кода и всех вершин, лежащих на путях, соединяющих их с корнем. Это конечное дерево является графическим представлением кода U. Обратно, каждое конечное бинарное дерево представляет некоторый префиксный код. Например, рассмотренный выше код (0, 10, 110, 111) представляется таким деревом:



Слова кода представляются листьями дерева, а длина слова есть длина пути из соответствующей вершины к корню. Задача построения оптимального

префиксного кода теперь может быть сформулирована как задача построения для данного набора чисел $P = (p_1, p_2, \dots, p_k)$ бинарного дерева с k листьями $a_1, a_2 \dots, a_k$ и минимальным значением величины

$$\sum_{i=1}^{k} p_i l_i,$$

где l_i– расстояние от вершины a_i до корня.

9.3. Оптимальный двоичный префиксный код

Рассмотрим случай, когда алфавит B состоит из двух букв: $B = \{0,1\}$. Предполагается, что буквы алфавита $A = \{a_1, a_2, \dots, a_k\}$ упорядочены по убыванию частот: $p_1 \geq p_2 \geq \cdots \geq p_k > 0$.

Лемма 1. *Существует оптимальный префиксный код* $U = (u_1, u_2, \dots, u_k)$ *такой, что* $|u_1| \leq |u_2| \leq \cdots \leq |u_k|$.

Доказательство. Пусть $U = (u_1, u_2, \dots, u_k)$ – оптимальный префиксный код и допустим, что при некоторых i и j, $i < j$, имеет место $|u_i| > |u_j|$. Рассмотрим код U', получающийся из кода U перестановкой слов u_i и u_j (напомним, что код – это упорядоченный набор слов). Тогда

$$C(P,U) - C(P,U') = p_i|u_i| + p_j|u_j| - p_i|u_j| - p_j|u_i| =$$

$$= (p_i - p_j)(|u_i| - |u_j|) \geq 0,$$

т.е. $C(P,U') \leq C(P,U)$. Так как код U оптимальный, а код U' – тоже префиксный, то может быть только равенство: $C(P,U') = C(P,U)$. Значит, U'– тоже оптимальный код. Повторяя такие перестановки слов, в конце концов получим оптимальный код, в котором слова располагаются в порядке возрастания длин. ■

Лемма 2. *Существует оптимальный префиксный код* $U = (u_1, \dots, u_k)$, *в котором слова* u_{k-1} *и* u_k *имеют наибольшую длину среди слов кода и различаются только в последней букве.*

Доказательство. Предыдущая лемма дает право считать, что в некотором оптимальном префиксном коде $U = (u_1, \dots, u_k)$ слова расположены по возрастанию длин: $|u_1| \leq |u_2| \leq \cdots \leq |u_k|$. Слово u_k имеет наибольшую длину среди кодовых слов. Допустим для определенности, что оно оканчивается буквой 1: $u_k = \alpha 1$. Рассмотрим код U', который получается отбрасыванием этой буквы: $U' = (u_1, \dots, u_{k-1}, \alpha)$. Тогда, очевидно, $C(P,U') < C(P,U)$. Но код U имеет наименьший коэффициент сжатия среди

префиксных кодов, значит, код U' – не префиксный. Префиксность могла нарушиться только из-за того, что слово α является префиксом другого кодового слова, скажем, u_i. Так как $u_k = \alpha 1$ – слово наибольшей длины в коде U, то u_i имеет такую же длину и отличается от u_k только в последней букве, т.е. $u_i = \alpha 0$. Все слова от u_i до u_k имеют одинаковую длину. Если $i \neq k-1$, то обменяем местами слова u_i и u_{k-1} и получим код с тем же коэффициентом сжатия. В этом коде два последних слова будут иметь одинаковую длину и отличаться только последней буквой. ■

Излагаемый ниже метод Хаффмана построения оптимального префиксного кода основан на сведении задачи для данного алфавита A к той же задаче для алфавита с меньшим числом букв. Это сведение описывает и обосновывает следующая теорема.

Пусть задан алфавит $A = \{a_1, a_2, \dots, a_k\}$ и набор частот $P = (p_1, p_2, \dots, p_k)$ его букв. Определим *редуцированный алфавит* $A' = \{a_1, a_2, \dots, a_{k-2}, b\}$ и редуцированный набор частот $P' = (p_1, p_2, \dots, p_{k-2}, p_{k-1} + p_k)$. Это можно трактовать как замену каждой из букв a_{k-1} и a_k новой буквой b.

Теорема 9.4 (теорема редукции). *Если* $U' = (u_1, \dots, u_{k-2}, w)$ – *оптимальный префиксный код для редуцированного алфавита* A' *при наборе частот* P', *то* $U = (u_1, , \dots, u_{k-2}, w0, w1)$– *оптимальный префиксный код для алфавита* A *при наборе частот* P.

Доказательство. Допустим, что это не так и оптимальным префиксным кодом при наборе частот P является код $V = (v_1, \dots, v_k)$, тогда $C(P,V) < C(P,U)$. Ввиду леммы 2 можно считать, что $v_{k-1} = \alpha 0$, $v_k = \alpha 1$ для некоторого слова α. Рассмотрим код $V' = \{v_1, \dots, v_{k-2}, \alpha\}$ и вычислим разность

$$C(P,V) - C(P',V') = p_{k-1}(|\alpha| + 1) + p_k(|\alpha| + 1) - (p_{k-1} + p_k)|\alpha| = \\ = p_{k-1} + p_k.$$

Такова же величина разности $C(P,U) - C(P',U')$. Следовательно,

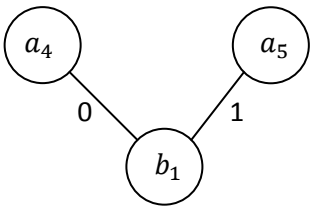
$$C(P,V) - C(P,U) = C(P',V') - C(P',U')$$

и если код U не является оптимальным кодом для распределения P, то и код U' не является оптимальным кодом для распределения P'. ■

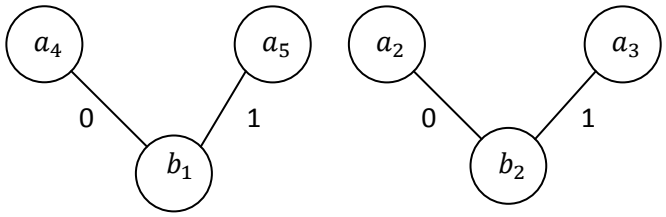
Алгоритм построения оптимального кода, основанный на этой теореме, удобно описывать как алгоритм построения бинарного дерева, представляющего код. Оно называется *деревом Хаффмана.*

Рассмотрим пример. Найдем оптимальный префиксный код для алфавита $A = \{a_1, a_2, a_3, a_4, a_5\}$ с набором частот $P = (0{,}5,\ 0{,}2,\ 0{,}1,\ 0{,}1,$

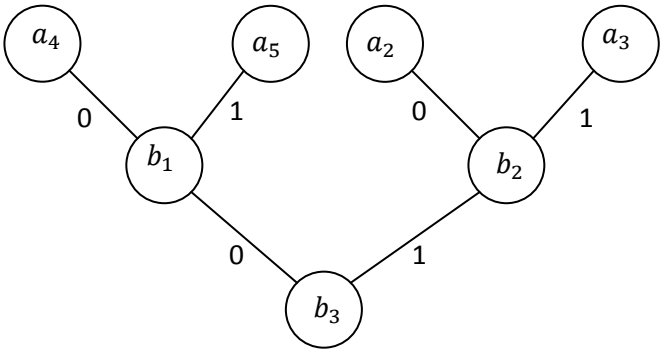
$0{,}1)$. Первый шаг редукции приводит к редуцированному алфавиту $A_1 = \{a_1, a_2, a_3, b_1\}$ и набору частот $P_1 = (0{,}5,\ 0{,}2,\ 0{,}1,\ 0{,}2)$. Соответствующий фрагмент дерева Хаффмена выглядит следующим образом:



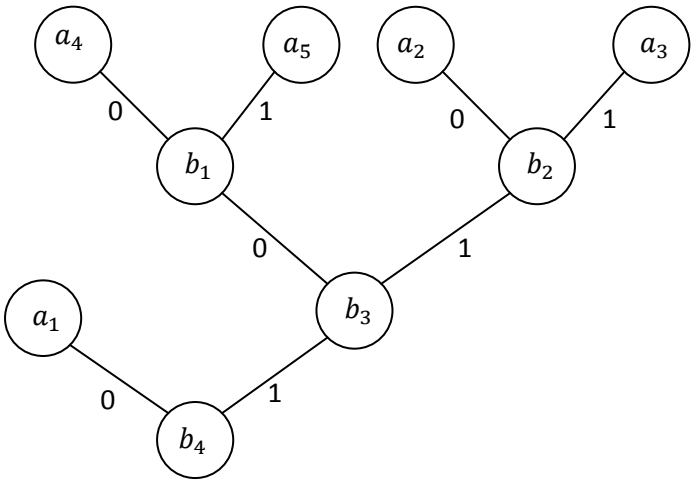
Для следующего шага редукции выбираем две буквы с наименьшими частотами в редуцированном алфавите. В данном случае это a_3 и одна из букв a_2, b_1. Возьмем a_2. Тогда следующий редуцированный алфавит будет $A_2 = \{a_1, b_1, b_2\}$ и набор частот $P_2 = (0{,}5, 0{,}2, 0{,}3)$. Графическое представление двух шагов редукции:



Теперь наименьшие частоты имеют буквы b_1 и b_2, следующий шаг редукции приводит к алфавиту $A_3 = \{a_1, b_3\}$ с набором частот $P_3 = (0{,}5, 0{,}5)$. Графическое оформление этого шага:



Оптимальным кодом для алфавита из двух букв при любом распределении частот будет, очевидно, код (0, 1). Построение этого кода можно рассматривать как еще один шаг редукции – к алфавиту из одной буквы. Итоговое дерево выглядит так:



Это дерево представляет все кодовые слова. Чтобы узнать, например, слово, кодирующее букву a_2, нужно проследить в этом дереве путь от корня (вершина b_4) к вершине, помеченной буквой a_2, и собрать все двоичные символы, приписанные ребрам этого пути. Получается слово 110. Таким образом, можно по этому дереву построить схему кодирования:

$$\begin{cases} a_1 \to 0 \\ a_2 \to 110 \\ a_3 \to 111 \\ a_4 \to 100 \\ a_5 \to 101 \end{cases}$$

В общем случае алгоритм построения оптимального префиксного кода, основанный на теореме редукции, можно описать следующим образом.

Алгоритм построения дерева Хаффмана

Дано: положительные числа $p_1, \dots, p_k$.

Результат: бинарное дерево с k листьями $a_1, \dots, a_k$; каждой вершине x этого дерева приписано число $p(x)$, причем $p(a_i) = p_i$, $i = 1, \dots, k$.

1. Создать вершины $a_1, \dots, a_k$;
   положить $p(a_i) = p_i$, $i = 1, \dots, k$;
   положить $A = \{a_1, \dots, a_k\}$.
2. Повторить $k - 1$ раз
   2.1. Создать новую вершину z.
   2.2. Выбрать в множестве A вершину x с наименьшим значением $p(x)$;
   сделать вершину x левым сыном вершины z;
   удалить x из A.
   2.3. Выбрать в множестве A вершину y с наименьшим значением $p(y)$;
   сделать вершину y правым сыном вершины z;

удалить y из A.

2.4. Положить $p(z) = p(x) + p(y)$;
добавить z к A.

8.4. Недвоичный оптимальный код

Алгоритм построения оптимального префиксного кода при $q > 2$ (напомним, q – это мощность алфавита B) немногим отличается от двоичного случая, опишем его без обоснования. В общем случае слова в алфавите из q букв тоже можно представлять путями в бесконечном корневом дереве, только в этом дереве у каждой вершине имеется q сыновей, а соответствующим ребрам приписаны буквы алфавита B. Как и в двоичном случае, префиксный код можно представить конечным фрагментом этого дерева.

Как и в двоичном случае, задача построения оптимального кода для данного алфавита A с данным набором частот P сводится к той же задаче для редуцированного алфавита с редуцированным набором частот. Только теперь на каждом шаге редукции находятся q букв с наименьшими частотами, они заменяются одной буквой, частота которой равна сумме частот заменяемых букв.

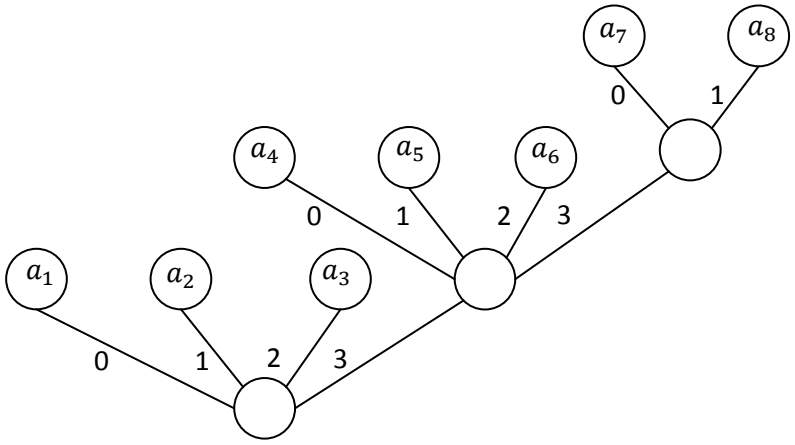
Осложнение по сравнению с двоичным случаем заключается в том, что теперь после нескольких шагов редукции в алфавите может остаться меньше q букв, но больше одной буквы. Может быть, следует на последнем шаге выполнить «неполную» редукцию – просто объединить все оставшиеся буквы, т.е. сделать соответствующие вершины сыновьями корня дерева? Оказывается, для построения оптимального кода «неполную» редукцию необходимо делать на первом шаге. Для того, чтобы она тоже стала полной, можно добавить к алфавиту A несколько фиктивных букв с нулевыми частотами.

Нетрудно рассчитать, сколько именно фиктивных букв нужно добавить. В результате каждого шага редукции q букв заменяются одной, т.е. число букв в алфавите уменьшается на $q - 1$. Если вначале в алфавите A было k букв, а к ним были добавлены еще s фиктивных букв, то после m шагов редукции получится алфавит из $k + s - m(q - 1)$ букв. Если после m шагов остается одна буква, то выполняется равенство $k + s - m(q - 1) = 1$, следовательно,

$$m = \frac{k + s - 1}{q - 1}.$$

Отсюда видно, что в качестве s нужно взять наименьшее целое неотрицательное число, при котором $k + s - 1$ делится без остатка на $q - 1$.

Для иллюстрации рассмотрим алфавит A из восьми букв с набором частот (0,2, 0,2, 0,2, 0,1, 0,1, 0,1, 0,05, 0,05) и алфавит B из четырех букв: $B = \{0, 1, 2, 3\}$. Наименьшее s, при котором $7 + s$ делится на 3, равно 2. Добавляем фиктивные буквы a_9 и a_{10} с нулевыми частотами и производим три шага полной редукции. В результате получается дерево (листья, соответствующие фиктивным буквам, не нарисованы):



Это дерево задает схему кодирования:

$$\begin{cases} a_1 \to 0 \\ a_2 \to 1 \\ a_3 \to 2 \\ a_4 \to 30 \\ a_5 \to 31 \\ a_6 \to 32 \\ a_7 \to 330 \\ a_8 \to 331 \end{cases}$$