Шамин Роман Вячеславович

Лекции по дискретной математике

Москва — 2016

УДК 517.98
ББК 22.16
Ш19

Шамин Р.В.
Лекции по дискретной математике. Москва, 2016.

Книга представляет собой учебник по дискретной математике. Этот учебник годится для первоначального изучения дискретной математики, но будет полезен и для углубленного изучения, поскольку содержит материал, который обычно не включают в учебники по дискретной математики.

Книга будет полезна студентам и аспирантам, а также всем желающим познакомится с современной абстрактной математикой.

Светлой памяти моего отца

Оглавление

Введение

Настоящее издание представляет собой лекции, читаемые автором а Российском университете дружбы народов для магистрантов, обучающихся по специальности «Фундаментальная информатика и информационные технологии».

В кратком пособии изложены основные темы, которые затрагиваются на лекциях. Так, в курс лекций входит теория множеств, комбинаторика, математическая логика, основы общей алгебры, теория графов, конечные автоматы, теория алгоритмов и элементы теории игр.

Изложение материала имеет ознакомительный характер, поэтому в данном курсе практически нет доказательств теорем и утверждений. Лекции ориентированы на повышение математической культуры и эрудиции будущих программистов, а также на прикладное использование материала в практике программирования и математического моделирования. В последующих изданиях планируется, что лекции будут расширены путем добавления в них примеров программ на Python для реализации основных конструкций.

Глава I

Теория множеств

1. Определение множества

Понятие множества относится к фундаментальным понятиям не только математики, но и любого абстрактного мышления. Строгих определений понятий «множество», «элемент множества», «принадлежность к множеству» и т.д. не существует. Однако они и не нужны, поскольку каждый это понимает априорно. В тоже время интуитивное понимание множества и способов его задания приводит к известным парадоксам и логическим проблемам типа «множество всех множеств». Для разрешения этих проблем используют конструкции, основанные на введении аксиомы выбора или эквивалентных ей утверждений. В курсе дискретной математики мы, как правило, будем иметь дело с конечными или счетными множествами, для которых этих проблем нет.

Мы будем говорить о множестве, как об определенной совокупности объектов, которые будем называть элементами множества. Сразу же нужно ввести понятие пустого множества. Множество, которое не содержит ни одного элемента называется пустым множеством, которое обозначается через $\varnothing$. Традиционно множества будем обозначать заглавными буквами: A, B, C, $\ldots$, а элементы маленькими буквами: a, b, c, $\ldots$. Если элемент множества a принадлежит множеству A, то будем писать

$$a \in A.$$

В противном случае пишем

$$a \notin A.$$

Мы будем говорить, что два множества A и B равны и писать

$$A = B,$$

если оба множества состоят из одинаковых элементов. В этом определении есть один (и не один) нюанс. Равны ли множества

$$A = \{1, 2, 3, 4, 5\}$$

и

$$B = \{5, 4, 3, 2, 1, \}?$$

Да, равны, поскольку множества предполагаются неупорядоченными. А равны ли множества

$$F_1 = \{Apple, Pear, Orange\}$$

и

$$F_1 = \{Apple, Apple, Pear, Orange\}?$$

Если в качестве элементов мы рассматриваем *одни и те же* яблоки, то да — эти множества равны[1].

Мы будем говорить, что множество A есть подмножество множества B, если каждый элемент множества A является элементом множества B. Будем использовать для этого стандартное обозначение

$$A \subset B.$$

Отношение подмножества является нестрогим, поэтому если $A = B$, то $A \subset B$ (и, соответственно, $B \subset A$). Если нужно рассматривать именно строгое отношение подмножества, которое называется собственным подмножеством, то это будем каждый раз оговаривать специальною.

Будем рассматривать следующие основные операции над множествами. Во-первых, множества можно объединять. Пусть A и B два множества, тогда их объединением, обозначаемым

[1] Выходит два яблока равны одному яблоку.

$A \cup B$ называется множество, состоящее из всех элементов множества A и множества B. Заметим, что, например,

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

Иногда вместо объединения множеств говорят о сумме множеств. Во-вторых, операция пересечения множеств. Пусть A и B два множества, тогда их пересечением, обозначаемым $A \cap B$ называется множество, состоящее из элементов, являющихся одновременно элементами множества A и множества B. Эта операция легко может приводить к пустым множествам. Будем говорить, что множества A и B не пересекаются, если

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

Очевидно, что обе эти операции являются симметричными

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

В-третьих, рассматривается операция (уже не симметричная) разности двух множеств, обозначаемой $A \setminus B$. При этом разность множества A и B определяется, как множество, содержащее те и только те элементы множества A, которые не содержатся в B. Разумеется,

$$A \setminus A = \varnothing.$$

Симметричную разность двух множеств A и B определим следующей формулой

$$A \triangle B = (A \setminus B) \cup (B \setminus A).$$

Симметричная разность множеств состоит из тех элементов, которые не являются одновременно элементами обоих множеств.

Чтобы определить операцию дополнения множеств, нужно ввести понятие универсального множества. Пусть в *фиксированной ситуации* мы рассматриваем только множества, являющиеся подмножеством одного множества U, которое называется универсальным множеством. Тогда для множества A, $(A \subset U)$ дополнением является множество $\overline{A}$, которое определяется по формуле

$$\overline{A} = U \setminus A.$$

Иногда используют обозначение $\mathbf{C}A$, для дополнения, но мы так делать не будем. Еще раз подчеркнем, что дополнение имеет смысл только при фиксации универсального множества.

2. Мощность множеств

Давайте зафиксируем во всей книге обозначения для некоторых множеств, с которыми мы будем постоянно иметь дело.

Во-первых, это множество натуральных чисел

$$\mathbb{N} = \{0, 1, 2, \dots\}.$$

Мы будем считать, что ноль является натуральным числом. В этом вопросе много споров и каждый по своему прав. Я выбрал этот вариант только для того, что если потребуется ряд из положительных целых чисел, то эстетичнее исключить ноль, чем его добавлять в противном случае.

Во-вторых, множество целых чисел

$$\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}.$$

В-третьих, множество рациональных чисел, обозначаемое $\mathbb{Q}$, заметим, что для определения этого множества мы воспользовались схемой свертки. В-четвертых, самое главное множество — это множество действительных чисел $\mathbb{R}$. И, наконец, множество комплексных чисел — $\mathbb{C}$.

В следующей главе мы будем встречаться с большим количеством пространств, которые являются множествами, и имеют устоявшиеся обозначения, но о них мы пока говорить не будем.

Рассмотренные выше множества являются множествами чисел. А что такое число? По идее число выражает некоторую количественную характеристику, необходимую для счета. В этом смысле только комплексные числа выбиваются из ряда. Их появление обусловлено операцией извлечения квадратного корня. Есть и дальнейшие обобщения чисел — кватернионы и др. Однако для нас важным будет определение чисел как алгебраическое поле. В общей алгебре полем называется множество M, которая является коммутативной группой относительно операции сложения $+$ с нулевым элементом 0, а также коммутативную группу с операцией умножения $*$ над ненулевыми элементами множества M. При этом эти операции должны удовлетворять свойствам дистрибутивности умножения относительно сложения.

Множества $\mathbb{Q}$, $\mathbb{R}$ и $\mathbb{C}$ являются полями с обычными операциями сложения и умножения. Часто употребляются устойчивые выражения «поле комплексных числе», «над полем действительных чисел».

Теперь мы рассмотрим принципиальное понятие в теории множеств — понятие мощности множества. В быту, встречаясь с множествами — множество гостей, множество подчиненных, множество комнат в квартире, множество конфет в вазе и т.д., нас прежде всего интересует количество элементов в данном множестве. Если для одних множеств мы можем мыслить (помнить и различать) все элементы как, например, множество комнат в своей квартире или множество своих детей, то для других множеств, наоборот, важно лишь количество, например, количество баранов в стаде, множество патронов в магазине и т.д. Для конечных множеств понятие мощности совпадает с количеством элементов.

С конечными множествами более менее все понятно, обратимся к бесконечным множествам.

Можно несколькими способами дать определение бесконечного множества. Определение «множество, содержащее бесконечное множество элементов» не годится, потому что тогда нужно давать определение «бесконечности», что, как раз, определяется через бесконечные множества. Определение — бесконечное множество это множество не являющееся конечным, корректно, но неконструктивно и, не раскрывает сути.

Мы сделаем следующим образом. По определению множество натуральных чисел $\mathbb{N}$ является бесконечным. Тогда множество A называется бесконечным, если существует такое подмножество $B \subset A$, что между элементами множества $\mathbb{N}$ и B существует взаимно однозначное соответствие. Можно дать еще одно эквивалентное определение. Множество называется бесконечным, если у него существует собственное подмножество (то есть не совпадающее с самим множеством) для элементов которых можно установить взаимно однозначное соответствие. Проверим, что множество $\mathbb{N}$ в этом определении является бесконечным. Действительно, рассмотрим собственное подмножество $\mathbb{N}_1 = \mathbb{N} \setminus \{0\}$, тогда для любому элементу $a \in \mathbb{N}_1$ соответствует единственный элемент $b = a - 1$, который принадлежит множеству $\mathbb{N}$.

И так, мощность конечного множества есть количество элементов

этого множества. Множества, которые допускают взаимно однозначное соответствие имеют одинаковую мощность. Мощность множества $\mathbb{N}$ обозначается $\aleph_0$[2]. Множества имеющие мощность $\aleph_0$ называются счетными. Используют различные обозначения для мощности множества: $|A|$, $\#A$ и др.

Мощность множества не является числом, однако для этих величин можно ввести порядок, то есть для любых двух мощностей a и b верно одно из трех $a < b$, $a = b$, $a > b$. Не являясь числами, мощности множеств являются трансфинитами или кардинальными числами.

Мощность $\aleph_0$ является наименьшей мощностью бесконечных множеств. Хорошо известно, что $|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}|$. Бесконечным множеством, имеющим большую мощность, является множество $\mathbb{R}$. Мощность этого множества называется мощностью континуума и обозначается $\mathfrak{c}$. Хорошо известно, что $\aleph_0 < \mathfrak{c}$. Возникает вопрос о существовании множества, имеющего мощность a такую, что

$$\aleph_0 < a < \mathfrak{c}.$$

Континуум-гипотеза, сформулированная Г. Кантором в 1877 году, утверждает, что таких множеств не существует. В 1940 году К. Гедель доказал, что *отрицание* континуум-гипотезы является недоказуемым при аксиоме выбора.

3. Отображения множеств

Еще одно понятие, которое понимается интуитивно, относится к отображением множеств. Пусть задано два множества X и Y. Мы будем говорить, что A есть отображение из X в Y, если задано *правило*, которое каждому элементу $x \in X$ ставит в *однозначное* соответствие элемент $y \in Y$. При этом пишут

$$A : X \to Y.$$

Мы будем говорить всегда об однозначных отображениях. В случаях, когда необходимо рассматривать многозначные отображения, мы будем говорить об однозначных отображениях в подмножества.

[2]Произносится «алеф-ноль».

Вместо термина «отображение» будем также использовать и другие синонимы, например, функция. Если множества X и Y представляют собой пространства, то часто будем говорить об операторе, если множество Y представляет собой множество чисел (действительных или комплексных), то будем говорить и функционале.

Иногда отображение будет определено не на всем X, а лишь на некотором подмножестве $X_1 \subset X$, это подмножество называется областью определения отображения A и обозначается $D(A)$. С другой стороны подмножество $Y_1 \subset Y$ состоящее из $y \in Y$, таких, что существует $x \in D(A)$, $y = A[x]$, называется областью значения отображения Y.

Для любого $x \in D(A)$ элемент $y = A[x]$ называется образом x. А для фиксированного $y \in Y$ множество таких $x \in D(A)$, что $A[x] = y$ называется прообразом y. Понятия образа и прообраза легко обобщаются на понятия образа и прообраза для множеств.

Если область значения отображения есть все Y, то такое отображение называется сюръективным. Если для двух различных $x_1, x_2 \in X$ их образы различны, то такое отображение называется инъективным. Отображение одновременно сюръективное и инъективное называется биективным. Биективное отображение, или биекция — это взаимнооднозначное отображение.

Введем понятие прямого произведения двух множеств X и B. Прямым произведением, обозначаемым $X \times Y$ называется множество, состоящее из всех *упорядоченных* пар (x, y), где $x \in X$, а $y \in Y$. Например, прямое произведение $\mathbb{R} \times \mathbb{R}$ есть множество точек, обозначаемое $\mathbb{R}^2$. Аналогично, можно ввести понятие и прямого произведения n множеств.

Рассмотрим некоторое непустое множество X. Для *некоторых* упорядоченных пар из этого множества мы введем понятия отношения. Будем говорит, что x_1 находится в отношении φ с x_2 и писать

$$x_1 \varphi x_2.$$

Отношения могут иметь различные свойства:

1. Отношение называется рефлексивным, если $x \varphi x$ для всех $x \in X$.

2. Отношение называется симметричным, если из $x_1 \varphi x_2$ следует, что $x_2 \varphi x_1$.

3. Транзитивным отношение называется, если из $x_1 \varphi x_2$ и $x_2 \varphi x_3$ следует, что $x_1 \varphi x_3$.

Отношение, обладающее всеми тремя свойствами — рефлексивностью, симметричностью и транзитивностью, называется отношением эквивалентности. Если на множестве X задано отношение эквивалентности, то это множество можно разбить на классы эквивалентности. Будем говорить, что подмножество $K \subset X$ есть класс эквивалентности для отношения эквивалентности φ если для всех $x_1, x_2 \in K$ имеет место $x_1 \varphi x_2$ и в $X \setminus K$ не существует элементов, состоящих в отношении с элементами из K. Для любого элемента $x \in X$ класс эквивалентности определяется следующим образом

$$K_x = \{x' \in X : x_1 \varphi x'\}.$$

Для любых x и y, которые не состоят в отношении эквивалентности, соответствующие классы эквивалентности не пересекаются. Поэтому все множество X можно разбить на классы эквивалентности

$$X = \bigcup_{\alpha} K_\alpha.$$

При этом можно каждый класс эквивалентности отождествить с некоторым новым элементом и рассматривать исходное множество, как множество классов эквивалентности.

Глава II

Комбинаторика и вероятность

1. Основные комбинаторные понятия

Комбинаторика или комбинаторный анализ посвящен изучению подмножеств конечных множеств. Комбинаторыне задачи играют выдающуюся роль в прикладной математики. Дело в том, что комбинаторные методы часто позволяют находить решения для множеств, содержащих огромное число элементов.

Рассмотрим конечное[1] множество A, содержащее N элементов, т.е. $|A| = N$. Через 2^A мы обозначим множество всех подмножеств множества A. Какова мощность этого множества? Само обозначение уже подсказывает, что мощность этого множества равна $2^{|A|} = 2^N$. Покажем, что это, действительно, так. Заметим, что в множество 2^A включается и пустое множество, которое тоже считается. Итак, занумеруем произвольным образом все элементы множества A

$$A = \{a_1, a_2, \ldots, a_N\}.$$

Далее для любого множества $\theta \in 2^A$ можно построить двоичную строку из N нулей и единиц

$$\theta_N = (\alpha_1, \alpha_2, \ldots, \alpha_N),$$

где

$$\alpha_n = \begin{cases} 1, & a_n \in \theta, \\ 0, & a_n \notin \theta. \end{cases}$$

[1] В этой главе все множества будут предполагаться конечными.

Ясно, что количество этих двоичных строк равно 2^N.

Вот пример множества всех подмножеств. Пусть

$$A = \{a, b, c\}.$$

Тогда множество всех подмножеств будет состоять из 8 элементов

$$2^A = \{\varnothing, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$$

Кроме вычисления (перечисления) всех подмножеств множества A в дискретной математике рассматривают такие важные понятия, как размещение, перестановки и сочетания. Размещением элементов из A по k называется упорядоченное подмножество множество из k элементов множества A. Размещения по 2 для множества $A = \{a, b, c\}$ следующие

$$(a, b), (b, a), (a, c), (c, a), (b, c), (c, b).$$

Число размещений множества из N элементов по k обозначается $(N)_k$. Не сложно вычислить это число, которое завит от двух параметров — N и k. По определению считаем

$$(N)_k = 0, \quad k > N,$$

$$(0)_0 = (N)_0 = 1.$$

Далее для случая, когда $0 < k < N$ имеем, что при построении размещения на первое место можно выбрать один из N вариантов, на второе — из $(N - 1)$, последний — из $(N - k + 1)$. В итоге мы получаем

$$(N)_k = N(N - 1) \dots (N - k + 1).$$

Эту формулу можно записать компактнее с помощью факториала

$$(N)_k = \frac{N!}{(N - k)!}.$$

Напомним, что факториал определяется для натуральных чисел следующим образом

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

для $n > 0$, по определению $0! = 1$.

Мы рассмотрели размещения без повторений, когда не допускается повторяющихся элементов. Если разрешить повторы, то мы получим размещения с повторами. Размещения с повторами имеют смысл слов определенной длины, составленных из фиксированного алфавита — множества A. Например, пусть $A = \{0, 1\}$ есть двоичный алфавит, тогда множество, размещений из A по 8 с повторениями будет множество двоичных чисел, состоящих из 8 бит, что соответствует множеству байтов. Количество байтов равно $256 = 2^8$. Несложно понять, что для множества A, состоящего из N элементов количество размещений по k с повторами равно N^k. Действительно, на первое место можно выбрать одним из N вариантов, на второе — тоже N и т.д. k раз.

В частном случае размещения без повторов, когда $N = k$, такое размещение называется перестановкой. Для нашего множества $A = \{a, b, c\}$ перестановки суть следующие множества

$$(a, b, c), (a, c, b), (b, a, c), (b, c, a), (c, a, b), (c, b, a).$$

Количество перестановок равно

$$(N)_N = N!.$$

Еще очень важным комбинаторным понятием является понятие сочетания элементов из множества A по k. Сочетанием называется неупорядоченное подмножество из k элементов, принадлежащих множеству A. Например, для $A = \{a, b, c\}$ сочетанием по 2 будут подмножества

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

Количество сочетаний множества из N элементов по k обозначается C_N^k. Сочетания отличаются от размещений тем, что они неупорядоченные. Соответственно, каждому сочетанию соответствуют $k!$ размещений, поэтому для числа сочетаний имеем следующая формула

$$C_N^k = \frac{(N)_k}{k!} = \frac{N!}{k!(N-k)!},$$

в которой предполагается, что $0 \le k \le N$. По определению полагаем

$$C_N^k = 0$$

при $k > N$.

Сочетания возникают, например, в биноме Ньютона

$$(a+b)^n = C_N^0 a^n + C_n^1 a^{n-1} b + \cdots + C_n^{n-1} a b^{n-1} + C_n^n b^n =$$

$$= \sum_{k=0}^{n} C_n^k a^{n-k} b^k.$$

Рассмотрим пример применения сочетаний для оценки шансов в игре «Спортлото». В этой игре нужно угадать 5 чисел из 36, либо 6 из 49. Какая из этих игр лучше (для игрока)? Число комбинаций «5 из 36» равняется сочетанию

$$C_{36}^5 = \frac{36!}{5!31!} = 376992.$$

А для игры «6 из 49» число комбинаций

$$C_{49}^6 = \frac{49!}{6!43!} = 13983816.$$

Таким образом, выиграть в лотерею «5 из 36» примерно в 37 раз легче.

Еще простой пример применения сочетаний. Сколько возможно раскладов 6 карт из колоды в 36 карт? Ответ

$$C_{36}^6 = \frac{36!}{6!30!} = 1947792.$$

Поэтому можно утверждать, что с огромной вероятностью Вам ни разу не выпадал одинаковый набор в карты при игре в «Дурака» (при условии, если колода была хорошо перетасованна).

2. Принцип включения-исключения

Как мы видим, комбинаторика позволяет с единых позиций находить точные ответы на нетривиальные вопросы. Очень эффективным средством является принцип включения-исключения.

Пусть для конечного множества A задана некоторая система подмножеств $A_1, A_2, \ldots, A_n$. Не предполагается, что эти подмножества

исчерпывают исходное множество или являются непересекающимися — это могут быть любые подмножества. Если мы знаем (можем легко сосчитать) мощности любых пересечений этих множеств, то можно найти мощность исходного множества A, если объединение всех A_i совпадает с Aб либо наоборот найти мощность подмножества множества A, которое состоит из элементов, не входящих ни в одно из A_i.

Действительно имеет место следующая формула, называемая принципом включения-исключения

$$|A \setminus (A_1 \cup \cdots \cup A_n)| = |A| - \sum_{i=1}^{n} |A_i| +$$

$$+ \sum_{1 \le i_1 < i_2 \le n} |A_{i_1} \cap A_{i_2}| - \cdots + (-1)^n \sum_{1 \le i_1 < \cdots < i_n \le n} |A_{i_1} \cap \cdots \cap A_{i_n}|.$$

Для доказательства этой формулы рассмотрим элемент $a \in A$ такой, что a входит ровно в k подмножеств. Следовательно,

$$a \notin A \setminus (A_1 \cup \cdots \cup A_n).$$

При этом в слагаемом $|A|$ этот элемент учитывается 1 раз, в слагаемом $\sum\limits_{i=1}^{n} |A_i|$ учитывается C_k^1 раз, в

$$\sum_{1 \le i_1 < i_2 \le n} |A_{i_1} \cap A_{i_2}|$$

учитывается C_k^2 раз и т.д. Таким образом, правая часть равна

$$C_k^0 - C_k^1 + \cdots + (-1)^k C_k^k = 0.$$

Последнее равенство следует из бинома Ньютона

$$(1 + x)^k = C_k^0 + C_k^1 x + \cdots + C_k^k x^k,$$

где $x = -1$.

Если элемент a не входит в $(A_1 \cup \cdots \cup A_n)$, то он учитывается в правой части один раз в слагаемом $|A|$.

Доказанный принцип включения-исключения имеет большое значение при вычислении вероятностей различных событий в дискретной теории вероятности.

3. Дискретная теория вероятностей

Теория вероятности представляет собой огромную математическую дисциплину со сложным и развитым математическим аппаратом. Однако наивная теория вероятности, которая имеет дело с конечными исходами может быть рассмотрена с помощью комбинаторных методов.

Теория вероятности занимается явлениями, которые, во-первых, повторяются, а, во-вторых, происходят случайным образом. Каждый единичный опыт, который рассматривается в теории вероятности может иметь некоторое количество элементарных исходов. Мы будем рассматривать такие модели, где множество элементарных исходов конечно. Обозначим множество исходов следующим образом

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_N\}.$$

Это означает, что в единичном опыте возникает (случайным образом) один и только один исход — $\omega_n \in \Omega$.

Например, пусть мы бросаем стандартный шестигранный кубик, то множество элементарных исходов будет следующее

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

Чтобы определить нашу вероятностную модель нужно каждому элементарному исходу приписать вероятность этого исхода. Исход ω_1 имеет вероятность p_1, ω_2 — p_2 и т.д. При этом должны выполняться следующие условия

$$0 \leq p_i \leq 1, \quad i = 1, 2, \ldots, N;$$

$$p_1 + p_2 + \cdots + p_N = 1.$$

Это значит, что с вероятностью единица (почти наверное) реализуется один исход.

В примере с кубиком, мы имеем равные вероятности

$$p_1 = p_2 = \cdots = p_6 = \frac{1}{6}.$$

Обычно нас интересуют не только и не сколько вероятности элементарных исходов, а события, связанные с этими исходами. Через $\mathcal{A}$

мы обозначим множество всех подмножеств множества Ω. Элементы множества $\mathcal{A}$ называются событиями. Для каждого события мы можем рассматривать вероятность этого события, которую обозначим $P(A)$, $A \in \mathcal{A}$. По определению считаем

$$P(\varnothing) = 0.$$

Далее

$$P(A) = \sum_{\omega_n \in \Omega} p_n.$$

То есть вероятность события есть сумма вероятностей элементарных исходов, которые формируют данное событие. Отсюда следует, что $P(\Omega) = 1$.

Заметим, что для любых $A, B \in \mathcal{A}$ имеет место

$$A \cup B \in \mathcal{A}, \quad A \cap B \in \mathcal{A}, \quad \Omega \setminus A \in \mathcal{A}.$$

Это значит, что совместное событие двух событий $(A \cap B)$ само является событием, а также противоположное событие $(\Omega \setminus A)$ тоже есть событие. Поэтому множество $\mathcal{A}$ является алгеброй событий.

В примере с кубиком можно рассмотреть событие, что выпадет нечетное значение. Это событие есть следующее множество

$$A_o = \{1, 3, 5\}.$$

Вероятность этого события есть

$$P(A_o) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}.$$

Введем еще событие

$$A_{123} = \{1, 2, 3\},$$

которое означает, что выпало не больше 3. Какова вероятность совместного события для событий A_o и A_{123}? Найдем это событие в виде множества

$$A_o \cap A_{123} = \{1, 3\}.$$

Легко видеть, что вероятность этого события равна $\frac{1}{3}$.

Таким образом, конечная вероятностная схема описывается следующей тройкой

$$\langle \Omega, \mathcal{A}, P \rangle,$$

где функция P есть функция, заданная на множестве Ω.

Сделаем два замечания. Во-первых, мы можем рассматривать не все множество подмножеств $\mathcal{A}$ множества Ω, а некоторую систему подмножеств множества Ω, но такую, что выполнены условия

1. $\varnothing \in \mathcal{A}$, $\Omega \in \mathcal{A}$,
2. $A \cup B \in \mathcal{A}$, $A, B \in \mathcal{A}$,
3. $\Omega \setminus A \in \mathcal{A}$, $A \in \mathcal{A}$.

Во-вторых, в ситуации, когда множество элементарных исходов является бесконечным, то вообще говоря, нельзя задать вероятность описанным выше способом, поскольку вероятность каждого элементарного исхода будет равна нулю. Более того, в качестве множества событий $\mathcal{A}$ уже не всегда можно будет рассматривать множество подмножеств множества Ω. Но в курсе дискретной математики мы не будем касаться этого.

4. Применение комбинаторных методов в задачах теории вероятности

Рассмотрим некоторые простые примеры из теории вероятности, которые решаются с использованием методов комбинаторики.

Пусть из колоды 36 карт случайным образом вынимается четыре карты. Найдем вероятность того, что мы получим в точности одну даму (любой масти). Все возможные варианты четырех карт содержат C_{36}^4 возможностей. Все эти возможности равновероятны. Одну даму мы можем выбрать C_4^1 вариантами, а три оставшиеся карты (не дамы) можно выбрать C_{32}^3 вариантами. Поэтому число всех возможных ситуаций, когда у нас будет одна дама, равно $C_4^1 C_{32}^3$. Таким образом, искомая вероятность вычисляется по формуле

$$p = \frac{C_4^1 \cdot C_{32}^3}{C_{36}^4} = \frac{\frac{4}{1} \cdot \frac{32 \cdot 31 \cdot 30}{1 \cdot 2 \cdot 3}}{\frac{36 \cdot 35 \cdot 34 \cdot 33}{1 \cdot 2 \cdot 3 \cdot 4}} = \frac{4 \cdot 32 \cdot 31 \cdot 30 \cdot 2 \cdot 3 \cdot 4}{2 \cdot 3 \cdot 36 \cdot 35 \cdot 34 \cdot 33} \approx 0.337$$

Теперь рассмотрим в этом же примере вероятность того, что в выбранных случайно четырех картах будет хотя бы одна дама. Обозначим это событие через A. А противоположное событие, которое

состоит в том, что у нас не будет ни одной дамы, обозначим через $\overline{A}$. Поскольку в любом случае должно произойти либо событие A, либо событие $\overline{A}$, то мы имеем

$$P(A) + P(\overline{A}) = 1.$$

Найдем сначала вероятность события $\overline{A}$. Выбрать четыре недамы можно C_{32}^4 способами. А всего возможностей выбрать четыре карты состоит из C_{36}^4 вариантов, поэтому имеем

$$P(\overline{A}) = \frac{C_{32}^4}{C_{36}^4} = \frac{32 \cdot 31 \cdot 30 \cdot 29}{36 \cdot 35 \cdot 34 \cdot 33} \approx 0.61.$$

Следовательно, искомая вероятность равна

$$P(A) \approx 1 - 0.61 \approx 0.39.$$

Рассмотрим схему Бернулли. В этой вероятностной модели рассматривается серия однотипных опытов, в каждом из которых возможен «успех» или «неудача». Причем успех возникает с вероятностью p, $0 < p < 1$, а неудача, соответственно, с вероятностью $q = 1 - p$. Будем считать, что проводится n отдельных опытов. В результате реализации схемы Бернулли мы получаем серию из n нулей и единиц.

Найдем вероятность того, что будет ровно m успехов при реализации схемы Бернулли. Будем рассуждать комбинаторно. Вероятность реализации любой конкретной последовательности, содержащей ровно m единиц (успехов) и $n - m$ нулей (неудач), например

$$11 \ldots 100 \ldots 0$$

равна

$$P_{mn} = pp \ldots pqq \ldots q = p^m q^{n-m}.$$

Но нас интересует вероятность любой последовательности, содержащей m единиц и $n - m$ нулей. Поскольку вероятности любых таких последовательностей равны P_{mn}, то для этого необходимо подсчитать количество всех возможных таких последовательностей и умножить на на вероятность P_{mn}. Количество таких последовательностей равняется числу сочетаний n элементов по m, поэтому число этих последовательностей в точности равно C_n^m. Таким образом, искомая вероятность может быть найдена по формуле

$$P_n(m) = C_n^m p^m q^{n-m}.$$

Кстати, можно заметить, что сумма получившихся вероятностей равна единице, что согласуется с вероятностными соображениями

$$\sum_{m=0}^{n} P_n(m) = \sum_{m=0}^{n} C_n^m p^m q^{n-m} = (p+q)^n = 1.$$

Глава III

Математическая логика

1. Логика высказываний

Математическая логика играет важнейшую роль в дискретной математики и ее приложениях. Как отдельная математическая дисциплина математическая логика включает в себя формальные системы, доказуемость математических утверждении, теорию алгоритмов, вычислимость и многие другие аспекты оснований математики. Мы рассмотрим вопросы исчисления высказываний и предикатов, а также приложение математической логики к формализации знаний в компьютерных программах.

Высказыванием называется утверждение, которое может быть либо истинным, либо ложным. Сразу же сделаем замечание, что мы будем рассматривать двоичную логику, где для высказываний возможны только «истина» или «ложь». Варианты типа «не знаю» или «не определено» не допустимы. Также мы не будем рассматривать варианты «истинно с вероятностью p» и т.д. Вообще существуют теории многозначной логики и нечеткой логики, но мы будем рассматривать только двоичную логику.

Итак, высказывания это утверждения типа «Волга впадает в Каспийское море», «$2 \times 2 = 5$», «Любое четное число больше 2 может быть разложено в сумму двух простых чисел». Здесь первое высказывание является истинным, второе — ложным, а третье — в настоящий момент нерешенная математическая проблема (проблема Гольдбаха). Сразу договоримся, что математическая логика не занимается вопросам какое высказывание ложно, а какое нет. Поскольку вы-

сказывания являются субъективными, например «Плутон — девятая планета Солнечной системы» раньше было истинным, а сейчас считается ложным, не говоря уже об оценочных высказываниях типа «черешня вкуснее вишни» или спорных высказываниях «Бога есть» / «Бога нет». В математической логике высказывания — это символы, часто обозначаемые буквами A, B, C и т.д., которые могут иметь одно из двух значений, которые мы будем обозначать 0 и 1, причем 0 означает ложное высказывание, а 1 — истинное, как это принято в языках программирования C/C++. Будем писать

$$A = 0, \quad B = 1.$$

Имея определенный запас высказываний, мы можем создавать новые высказывания с помощью логических операций. Для таких высказываний мы уже сможем методами математической логики выяснить ложность или истинность.

Первая логическая операция — это операция отрицания

$$\overline{A}$$

Высказывание $\overline{A}$ истинно тогда, когда ложно A и ложно, когда истинно A. Читается «не A».

Вторая операция называется конъюнкцией, которая применяется к паре высказываний

$$A\&B,$$

читается «A и B». Высказывание $A\&B$ истинно тогда и только тогда, когда $A = 1$ и $B = 1$.

Третьей операций является операция дизъюнкция также применяемая к паре высказываний

$$A \vee B,$$

читается «A или B». Соответственно, высказывание $A \vee B$ истинно тогда, когда истинно хотя бы одно из высказываний A или B.

Четвертая логическая операция — это операция импликация или логическое следование. Также применяется для пары высказываний

$$A \to B,$$

читается «из A следует B». Высказывание $A \to B$ ложно тогда и только тогда, когда A истинно, а B ложно. Во всех остальных случаях это высказывание истинно. Если операции логического отрицания, конъюнкции и дизъюнкции интуитивно поняты любому разумному человеку, то с операцией импликации иногда возникают сложности. Дело в том, что если $A = 0$, то результатом операции будет 1 вне зависимости от B. Как говорил, Д. Гильберт «Разрешите мне принять, что дважды два — пять, и я докажу, что из печной трубы вылетает ведьма!» – из ложного предположения следует что угодно.

Пятая операция — это операция эквивалентности двух высказываний

$$A \sim B,$$

читается «A эквивалентно B». Высказывание $A \sim B$ истинно тогда, когда либо $A = 1$ и $B = 1$, либо $A = 0$ и $B = 0$.

С помощью введенных операций и с использованием скобок, для обозначения приоритета можно конструировать произвольно сложные высказывания, например,

$$W = (A \vee B) \to \overline{((B\&C) \vee (D \sim E))}.$$

Знак равенства означает, что мы получаем новое высказывание W по приведенной формуле.

Заметим, что используемые нами операции не являются независимыми и могут быть выражены друг через друга. Вот примеры

$$A\&B = \overline{\overline{A} \vee \overline{B}},$$

$$A \sim B = (A \to B)\&(B \to A),$$

$$A \to B = (\overline{A} \vee B).$$

Вообще говоря, все логические операции можно выразить через две операции $\vee$ и $^-$ или операции $\&$ и $^-$.

Далее, существует большое количество формул, устанавливающих равносильность сложных высказываний. Приведем только некоторые из них

$$\overline{\overline{A}} = A,$$

$$A\&B = B\&A,$$

$$A \vee B = B \vee A,$$

$$(A \vee B) \vee C = A \vee (B \vee C),$$
$$(A\&B)\&C = A \vee (B\&C),$$
$$A\&(B \vee C) = (A\&B) \vee (A\&C),$$
$$A \vee (A\&B) = A,$$
$$A\&(A \vee B) = A.$$

Рассмотрим подробнее первую формулу, которая имеет название «отрицание отрицания». Согласно этой формуле высказывание будет истинным, если ложным будет отрицание этого высказывания. С точки зрения логики высказываний эта формула не вызывает сомнения. Но с точки зрения конструктивной логики здесь есть вопросы. Дело в том, что опровергнуть отрицание может быть значительно сложнее, чем доказать исходное высказывание. Например, вместо доказательства вины обвиняемого можно предложить самому обвиняемому опровергнуть обвинение. Ясно, что это неэквивалентные ситуации. Поэтому принцип презумпции невиновности в России гарантируется статьей 49 Конституции РФ и является базовым в большинстве стран. Этот принцип звучит так: «Человек не виновен, пока не доказано обратное».

Формула называется тождественно истинной или тавтологией, если она принимает истинное значение при всех значениях входящих в нее высказываний. Например такая формула

$$A \vee \overline{A} = 1,$$

которая называется исключение третьего «A или не A». Формула называется невыполнимой, если она принимает ложное значение при всех значениях переменных. Например

$$A\&\overline{A}.$$

Если формула не является невыполнимой, то она называется выполнимой.

Для того, чтобы выяснить будет ли истинной заданная формула при определенных значениях входящих в нее переменных необходимо построить таблицу истинности для этой формулы. В этой таблице перечисляются все возможные комбинации входящих переменных и значение формулы. Рассмотрим пример для формулы $A \vee (B\&C)$.

A	B	C	$A \vee (B\&C)$
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

Из комбинаторных формул мы знаем, что количество строк у таблицы истинности равно 2^N, где N равно количеству переменных. Ясно, что для формул имеющих большое количество переменных таблицы истинности будут очень громоздкими. Существуют различные методы эквивалентного преобразования логических формул, которые облегчают проверку истинности или ложности при заданных значениях переменных. В частности, можно доказать, что любую формулу можно привести к эквивалентной формуле, которая является дизъюнкцией элементарных конъюнкций. Такая форма формулы называется дизъюнктивной нормальной формой. Аналогично конъюнктивной нормальной формой называется вид формулы, когда она представляет собой конъюнкцию элементарных дизъюнкций. Например, формула

$$(A\&B\&\overline{C}) \vee A \vee (A\&\overline{B})$$

находится в дизъюнктивной нормальной форме, а формула

$$(A \vee \overline{B})\&(B \vee C)\&A$$

находится в конъюнктивной нормальной форме. Использование нормальных форм позволяет в ряде случаев сразу сделать вывод о значении формулы. Например, сразу видно, что последняя формула имеет ложное значение, если ложным является переменная A.

2. Правила вывода и рассуждения

Математическая логика дает формальные приемы для проведения правильных рассуждений, при заданных посылках. Рассуждением

называется процесс установления истинности высказывания, которое называется логическим выводом при конъюнкции набора высказываний, которые называются посылками. Дадим определение правильного рассуждения. Рассуждение называется правильным, если из конъюнкции посылок следует заключение, т.е. всякий раз, когда все посылки истинны, заключение тоже истинно.

Пусть $X_1, X_2, \ldots, X_n$ — набор высказываний, называемых посылками, а D — высказывание, являющееся заключением рассуждения. Тогда это рассуждение записывается следующим образом

$$\frac{X_1, X_2, \ldots, X_n}{D}.$$

Наиболее часто используемое рассуждение можно записать следующим образом

$$\frac{A \to B, A}{B}.$$

Это рассуждение по правилу modus ponens. Легко видеть, что это правильное рассуждение. В качестве примера такого рассуждения можно привести следующее рассуждение «Если космической аппарат стал искусственным спутником Земли, то он движется с первой космической скоростью. Наш космический аппарат вышел на орбиту спутника Земли. Следовательно, наш аппарат движется со второй космической скоростью».

Другой распространенный способ правильного рассуждения записывается в виде

$$\frac{A \to B, \overline{B}}{\overline{A}}.$$

Пример такого рассуждения. «Если продукты ГМО вредны, то они опасны для человека. Продукты ГМО не опасны для человека[1]. Следовательно, продукты ГМО не вредны.»

Рассмотрим теперь неправильные рассуждения. Например рассуждение по формуле

$$\frac{A \to B, B}{A}.$$

Примером такого рассуждения является рассуждение «Если функция $f(x)$ дифференцируема, то она непрерывна.

[1]European Commission Directorate-General for Research and Innovation; Directorate E — Biotechnologies, Agriculture, Food; Unit E2 — Biotechnologies (2010)

Функция $f(x)$ непрерывна. Следовательно функция $f(x)$ дифференцируема». Заметим, что неправильные рассуждения могут приводить к правильным выводам.

Правила вывода и автоматические рассуждения используются в экспертных системах, которые могут получать заключения по заданным посылкам. Экспертные системы последовательно задают пользователям вопросы, после чего система, используя правила вывода, приходит к заключению.

Рассмотрим простейшую схему для экспертной системы, основанной на логических правилах. Во-первых, рассматриваем набор высказываний

$$X_1, X_2, \ldots, X_n.$$

Эти высказывания называются фактами в экспертной системе. Экспертная система «узнает» факты с помощью опроса пользователя.

Во-вторых, в системе считается истинным набор высказываний вида

$$X_i \to Y_j,$$

$$Y_k \to Y_m.$$

Здесь Y_k — это выводы экспертной системы.

В-третьих, в системе используются истинные высказывания вида

$$(Y_{i_1} \& Y_{i_2} \& \ldots \& Y_{i_k}) \to D_i,$$

где $D_1, D_2, \ldots, D_m$ суть решения экспертной системы.

Схема работы экспертной системы состоит в трех этапах.

1. Получение фактов, т.е. присвоение переменным X_i логических значений.

2. Получение выводов Y_j с использованием логических правил вывода.

3. Вывод решений экспертной системы.

Второй этап в этой системе может быть опущен, если правила вывода сразу могут дать решение экспертной системы по полученным фактам.

Рассмотрим простой пример экспертной системы для классификации животных. Фактами в данной системе будет следующий набор высказываний:

X_1 = животное имеет крылья.

X_2 = животное умеет плавать.

X_3 = животное умеет летать.

X_4 = животное имеет бинокулярное зрение.

X_5 = животное имеет усы.

С помощью опроса пользователя система присваивает каждому высказыванию значение 0 или 1.

В качестве выводов в экспертной системе будем использовать следующие высказывания.

Y_1 = животное — птица.

Y_2 = животное — рыба.

Y_3 = животное — млекопитающее.

Наконец, в нашей экспертной системе мы используем следующие решения.

D_1 = животное — курица.

D_2 = животное — сокол.

D_3 = животное — сом.

D_4 = животное — щука.

D_5 = животное — тигр.

D_5 = животное — баран.

Для нахождения значений высказываний Y_i и D_j необходимо использовать правила вывода. Мы будем считать верными следующие формулы

$$X_1 \to Y_1,$$

$$X_2 \to Y_2,$$

$$(\overline{X}_1 \& \overline{X}_2) \to Y_3,$$

которые дают возможность прийти к некоторым промежуточным выводам.

Далее рассмотрим правила вывода для нахождения решений экспертной системы.

$$(Y_1 \& \overline{X}_3) \to D_1,$$

$$(Y_1 \& X_3) \to D_2,$$

$$(Y_2 \& X_5) \to D_3,$$

$$(Y_2 \& \overline{X}_5) \to D_4,$$
$$(Y_3 \& X_4) \to D_5,$$
$$(Y_3 \& \overline{X}_4) \to D_6.$$

Таким образом, мы получаем, что если в результате опроса пользователя будут установлены следующие факты

$$X_1 = 0, \quad X_2 = 0, \quad X_3 = 0, \quad X_4 = 1, \quad X_5 = 1,$$

то наша система придет к выводу, что «животное — млекопитающее» и заключению, что мы имеем дело с тигром.

3. Логика предикатов

Рассмотренные выше высказывания по сути были только логические переменные, значение которых было фиксированным. Для проведения настоящих рассуждений необходимо использовать логику предикатов. Предикаты — это высказывания с параметрами, которые могут быть истинными или ложными в зависимости от параметров. Дадим точное определение. Рассмотрим произвольное множество M, которое будем называть множеством предметов или объектов. На этом множестве определим однозначную функцию

$$P : M \to \{0, 1\},$$

которая каждому объекту (предмету) $x \in M$ ставит в соответствие ложь или истину (0 или 1). Следовательно, для каждого $x^* \in M$ конструкция $P(x^*)$ есть высказывание.

Приведем примеры простых предикатов. Пусть $M = \mathbb{N}$ — множество целых чисел. Построим предикат

$$P(n) = \{n \text{ является простым числом}\}.$$

Имеем различные высказывания $P(2) = 1$, $P(3) = 1$, но $P(4) = 0$ и т.д. Можно рассматривать предикаты, зависящие от нескольких переменных. В этом случае множество M можно считать прямым произведением, например, $M = Peoples \times \mathbb{N}$, где $Peoples$ — множество людей в фиксированный момент времени. Рассмотрим предикат

$$P(x, n) = \{\text{Человек } x \text{ имеет возраст } n \text{ лет}\}.$$

Математическая логика не рассматривает вопрос о том, как именно вычислять предикат, поскольку это вопросы предметной области. Будем считать, что предикат определен на всем множестве объектов. Вообще говоря, для каждого предиката нужно задавать свою область определения — множество предметов.

Из предикатов и высказываний можно строить новые предикаты с помощью логических операций. Например, пусть предметное множество состоит из множества автомобилей и суммы денег, введем предикаты

$$P_1(m, x) = \{\text{машина } m \text{ стоит } x \text{ рублей}\},$$

$$P_2(x) = \{\text{я имею } x \text{ рублей}\}.$$

Тогда можно определить новый предикат

$$P_3(m, x) = P_1(m, x)\&P_2(x).$$

Этот предикат может иметь смысл — «я могу купить машину m».

Однако с помощью предикатов можно создавать новые высказывания и с использованием кванторов. Пусть мы имеем множество предметов M и предикат $P(x)$, тогда построим высказывание

$$\forall x P(x),$$

которое является истинным, если $P(x)$ истинно для всех $x \in M$, и ложным, если существует хотя бы один $x' \in M$, для которого $P(x') = 0$. Этот квантор читается «для всех x $P(x)$ истинно». Квантор $\forall$ называется квантором всеобщности. Говорят, что квантор всеобщности связывает переменную. Если предикат зависит от нескольких переменных, а квантор всеобщности применяется по одной, то мы получаем предикат от одного переменного. Например, пусть у нас есть предикат, зависящий от двух натуральных чисел, и заданный по формуле

$$P(n, m) = \{n \leq m\}.$$

Образуем новый предикат

$$Q(n) = \forall m(P(n, m)).$$

Имеем

$$Q(0) = 1; \quad Q(n) = 0,\ n > 0.$$

Следующий квантор — это квантор существования. Который обозначается следующим образом

$$\exists x P(x).$$

Этот квантор из предиката делает высказывание, которое истинно, если существует хотя бы один элемент $x' \in M$ такой, что $P(x') = 1$, если такого элемента не существует, то это высказывание будет ложным. Читается «сущесвтует такой x, что $P(x)$ истинно».

Например, для предиката

$$P(x) = \{|\sin x| > 1\},$$

определенного на множестве действительных чисел, высказывание

$$\exists x P(x)$$

ложное. Впрочем, если мы рассмотрим этот предикат, определенный на множестве комплексных чисел, то полученное высказывание будет уже истинным.

Квантор существования также связывает переменную. Можно комбинировать кванторы. Например, для предиката

$$P(n, m) = \{n \leq m\}$$

построим высказывание

$$A = \exists n(\forall m(P(n, m))).$$

Это высказывание будет истинным. А высказывание

$$B = \forall n(\exists m(P(n, m))).$$

будет ложным, поскольку существует такое m, что не для любого n будет выполнено $n \leq m$.

Кванторы всеобщности и существования являются двойственными друг к другу в том смысле, что имеют место следующие соотношения

$$\overline{\forall x(P(X))} = \exists x(\overline{P}(x))$$

и

$$\overline{\exists x(P(X))} = \forall x(\overline{P}(x)).$$

С помощью предикатов можно определять множества, как подмножества множества предметов. Пример, множество предметов — действительные числа $\mathbb{R}$, предикат

$$P(x) = \{\sin x = 0\},$$

построим множество корней для функции $\sin x$ с помощью этого предиката

$$Q = \{x \in \mathbb{R} : P(x) = 1\}.$$

Таким образом, любой предикат задает два подмножества в множестве предметов — подмножество, для которого этот предикат истинный, и подмножество, для которого этот предикат ложный.

Рассмотрим на одном общем множестве предметов M два предиката $P(x)$ и $Q(x)$. Как мы видели выше эти предикаты определяют два подмножества множества

$$M_P = \{x \in M : P(x) = 1\}$$

и

$$M_Q = \{x \in M : Q(x) = 1\}.$$

Определим теперь новый предикат на этом же множестве предметов

$$R(x) = P(x) \vee Q(x)$$

и для этого предиката свое множество

$$M_R = \{x \in M : R(x) = 1\}.$$

Легко заметить, что имеет место следующее соотношение

$$M_R = M_P \cup M_Q.$$

Аналогично, если мы определим предикат

$$S(x) = P(x) \& Q(x)$$

и, соответственно, множество

$$M_S = \{x \in M : S(x) = 1\},$$

то получим соотношение

$$M_S = M_P \cap M_Q.$$

Рассмотрим еще предикат, построенный по формуле

$$T(x) = \overline{P}(x).$$

Тогда для множества

$$M_T = \{x \in M : T(x) = 1\},$$

имеем

$$M_T = \overline{M}_P,$$

то есть дополнение к множеству.

По сути предикат в данном случае используется в качестве индикаторной функции для задания множеств.

Глава IV

Алгебраические структуры

1. Алгебраические операции

В настоящей главе мы рассмотрим основы общей алгебры. Общая или абстрактная алгебра это важнейшая математическая дисциплина, которая изучает множества с введенными в них алгебраическими операциями. Методы общей алгебры находят свое применение почти во всех разделах математики. Большое количество прикладных математических задач требуют использования методов общей алгебры. Как замечательно сказал Жан Д'Аламбер: «Алгебра щедра: она нередко дает больше, чем от нее можно было бы требовать».

Будем рассматривать множество M, которое в дальнейшем без оговорок будем считать непустым. Бинарной алгебраической операцией в множестве M будем называть операцию, заданную на упорядоченной паре элементов a и b из M, результатом этой операции будет также элемент множества M. Будем это записывать следующим образом

$$c = a \circ b,$$

где $c \in M$. В этом случае говорят, что эта операция является замкнутой в множестве M. Пара

$$\langle M, \circ \rangle$$

называется группоидом. Отметим, что в данном случае мы не предполагаем ни коммутативность, ни ассоциативность этой операции. Группоиды являются простейшими алгебраическими структурами,

но их общность не позволяет развить более менее содержательную теорию.

Группоид, у которого групповая операция удовлетворяет условию ассоциативности

$$(a \circ b) \circ c = a \circ (b \circ c)$$

называется полугруппой. Для полугруппы можно просто писать

$$a_1 \circ a_2 \circ \cdots \circ a_n$$

для произвольных $a_1, a_2, \ldots, a_n \in M$. Если полугруппа имеет такой элемент $e \in M$, что для всех $a \in M$ выполнены равенства

$$e \circ a = a \circ e = a,$$

то такой элемент называется единицей, а сама полугруппа называется полугруппой с единицей или моноидом.

Очевидно, что единичный элемент единственный. Действительно, пусть в моноиде два единичных элемента e' и e'', тогда

$$e' \circ e'' = e,$$

где по определению $e = e'$ и $e = e''$.

Для элемента $a \in M$ элемент $a^{-1} \in M$ называется обратным элементом, если выполнены равенства

$$a \circ a^{-1} = a^{-1} \circ a = e,$$

где e есть единичный элемент. Группой называется моноид, для всех элементов которого существует обратный элемент.

Покажем, что в группе обратный элемент определяется единственным образом. Пусть для любого $a \in M$ существует два элемента $a' \in M$ и a'' такие, что

$$a \circ a' = e = a'' \circ a.$$

Используя ассоциативность операции, мы имеем

$$a'' \circ a \circ a' = a'' \circ (a \circ a') = a'' \circ e = a''$$

и

$$a'' \circ a \circ a' = (a'' \circ a) \circ a' = e \circ a' = a'.$$

Следовательно, $a'' = a' = a^{-1}$.

Покажем, что имеет место

$$(a_1 \circ a_2)^{-1} = a_2^{-1} \circ a_1^{-1}.$$

Действительно,

$$a_2^{-1} \circ a_1^{-1} \circ a_1 \circ a_2 = a_2^{-1} \circ a_2 = e$$

и

$$a_1 \circ a_2 \circ a_2^{-1} \circ a_1^{-1} = a_1 \circ a_1^{-1} = e.$$

Вообще говоря, групповая операция не предполагается коммутативной, и многие содержательные примеры групп этим свойством не обладают. Но если для группы (полугруппы) для любых элементов a и b имеет место

$$a \circ b = b \circ a,$$

тогда такая группа (полугруппа) называется коммутативной или абелевой.

Групповая операция $\circ$ часто называется «умножением» или (особенно для абелевых групп) «сложением». Для случая «сложения» единичный элемент называется нулем.

2. Примеры полугрупп, групп

Простейшими примерами полугрупп являются числа. Рассмотрим полугруппу на множестве натуральных чисел $\mathbb{N}$, где в качестве алгебраической операции рассмотрим сложение. Тогда получаем полугруппу

$$\langle \mathbb{N}, + \rangle.$$

Эта полугруппа является моноидом, где единичный элемент 0. Поскольку от перемены мест слагаемых сумма не меняется, то эта полугруппа будет абелевой. Однако эта полугруппа не будет группой, поскольку для положительных чисел нет натуральных обратных.

Если мы рассмотрим такую же полугруппу, но для множества $\mathbb{N}_1 = \mathbb{N} \setminus \{0\}$, то такая полугруппа не будет моноидом.

Полугруппой также будет

$$\langle \mathbb{N}, \times \rangle,$$

где групповая операция — это умножение. Единицей в этой полугруппе будет уже 1.

Для множества целых чисел $\mathbb{Z}$ с групповой операцией сложения мы имеем группу. А для действительных чисел $\mathbb{R}$ или рациональных чисел $\mathbb{Q}$ мы получаем абелевы группы как по сложению, так и по умножению.

Теперь рассмотрим общую схему для определения полугрупп и групп. Рассмотрим произвольное непустое множество A. В этом множестве можно рассмотреть все возможные отображение этого множества в себя

$$f : A \to A.$$

Множество всех таких отображений обозначается 2^A. Для двух любых отображений $f, g \in 2^A$ можно определить алгебраическую операцию через композицию отображений

$$(f \circ g)(x) = f(g(x)),$$

для всех $x \in A$.

По определению эта операция является ассоциативной, поэтому множество 2^A является полугруппой. Поскольку тождественное отображение e

$$e(x) = x$$

тоже входит в 2^A, то мы имеем моноид. Конечно, эта полугруппа не будет абелевой.

Рассмотрим подмножество в 2^A, состоящее из взаимно однозначных отображений множества A. Такие отображения называются автоморфизмами множества A. Обозначим это множество G_A. Очевидно, что композиция взаимно однозначных отображений — будет взаимно однозначным отображением, поэтому G_A является полугруппой с единицей, поскольку единичное отображение входит в G_A. Кроме того, взаимно однозначные отображения имеют обратные отображения, поэтому G_A является группой, порожденной автоморфизмами множества A.

Рассмотрим случай конечного множества A, элементы которого можно занумеровать от 1 до N, где N — мощность множества A. Поскольку в алгебре нет значения конкретной природе этого множества, то для сокращения письма будем считать, что это множество

$$A = \{1, 2, \ldots, N\}.$$

Автоморфизмы конечного множества называются перестановками (мы это помним из комбинаторики). Эта группа называется симметрической группой n-й степени. Количество этих перестановок равно $N!$.

Удобно записывать перестановки в виде

$$\begin{pmatrix} 1 & 2 & \dots & N \\ a_1 & a_2 & \dots & a_N \end{pmatrix},$$

где $a_1, a_2, \dots, a_N$ неповторяющиеся числа от 1 до N.

Эти перестановки можно перемножать, получая новые перестановки. Например, для $N = 5$, пусть

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

и

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

Тогда

$$g \circ f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 4 & 1 & 2 \end{pmatrix}$$

Благодаря ассоциативности групповой операции для каждого элемента $f \in G_A$ можно рассматривать степени

$$f^k = f \circ f \circ \dots \circ f \quad - \ k \text{ раз.}$$

Рассмотрим степени для $N = 4$ для перестановки

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}.$$

Имеем

$$f^2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix},$$

далее

$$f^3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 \end{pmatrix}$$

и еще раз

$$f^4 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

Мы получили, что четвертая степень нашей перестановки является единичной. Это не случайно, в любой конечной группе[1] для любого элемента $f \in G_A$ существует степень k такая, что

$$f^k = e.$$

Пусть G любая группа. Тогда подмножество $H \subset G$ называется подгруппой группы G, если для любых $a, b \in H$ верно, что

$$a \circ b \in H$$

и вместе с любым $a \in H$ обратный элемент a^{-1} принадлежит множеству H.

Строить подгруппы можно следующим образом. Выбираем любое $a \in G$ и включаем в множество H все степени этого элемента

$$a^k, \quad k = 0, \pm 1, \pm 2, \ldots,$$

где $a^0 = e$ и a^{-k} есть обратный элемент для a^k. Такая подгруппа называется циклической подгруппой, порожденной элементом a. Если начиная с некоторого $k \geq 0$ имеет место

$$a^k = e,$$

то элемент a называется элементов конечного порядка с порядком k[2]. В противном случае циклическая группа называется бесконечной.

Рассмотрим группу вращений на плоскости. Пусть на плоскости xOy определим преобразования V_α — вращения на угол $\alpha \geq 0$ против часовой стрелки вокруг центра O. Поскольку композиция таких вращений есть снова вращение на угол, то можно определить алгебраическую операцию

$$V_\alpha \circ V_\beta = V_{\alpha+\beta}.$$

Элемент V_0 есть единичный элемент, поэтому $\{V_\alpha\}$ есть моноид. С другой стороны поскольку

$$V_\alpha \circ V_{2\pi-\alpha} = V_{2\pi} = V_0,$$

то мы имеем группу, причем абелеву.

---

[1] Конечная группа — это группа, содержащая конечное количество элементов.

[2] Здесь предполагается, что k минимальное число для которого выполнено это равенство

3. Кольца, тела, поля

Кольцом называется непустое множество R, на котором заданы две бинарные алгебраические операции — сложение

$$a + b \in R$$

и умножение

$$ab \in R$$

для любых $a, b \in R$, по сложению R должно быть абелевой группой, а по умножению только группоид, но должны быть выполнены законы дистрибутивности

$$a(b + c) = ab + ac,$$

$$(b + c)a = ba + ca,$$

для всех $a, b, c \in R$.

Если операция умножения является ассоциативной, то мы имеем ассоциативное кольцо.

Рассмотрим равенство

$$c + (b - c) = b.$$

Умножим слева это равенство на a, получаем

$$ac + a(b - c) = ab,$$

далее

$$a(b - c) = ab - ac.$$

Это означает, что закон дистрибутивности выполняется и для разности.

Кстати, каждая абелева группа может рассматриваться как кольцо, если в этой группе ввести нулевое умножение

$$ab = 0.$$

Это кольцо называется нулевым кольцом.

В качестве нетривиального примера кольца можно рассматривать кольцо целых чисел с обычными операциями сложения и умножения. Это будет ассоциативное кольцо.

Еще одним примером кольца является кольцо квадратных матриц с действительными элементами. Кстати, в этом кольце есть делители нуля, т.е. ненулевые $a, b \in R$ такие, что

$$ab = 0.$$

Более того, если мы имеем какое-либо кольцо R, то можно рассматривать квадратные матрицы порядка n, определяя для них обычным образом сложение и умножение матриц. Таким образом, мы получим новое кольцо матриц с элементами исходного кольца.

Еще один содержательный пример кольца — это функции со значением в кольце. Пусть X есть произвольное множество, а R — произвольное кольцо. Через $F(X, R)$ обозначим множество функций

$$f : X \to R,$$

т.е. множество функций, заданных на X со значениями в R. Для элементов множества $F(X, R)$ введем две алгебраические операции

$$(f + g)(x) = f(x) + g(x),$$

$$(fg)(x) = f(x)g(x).$$

Легко видеть, что с этими операциями само множество $F(X, R)$ будет кольцом. Это кольцо называется полным кольцом функций на множестве X со значениями в R.

В задачах дискретной математики возникает еще кольцо многочленов. Рассмотрим множество всех возможных многочленов

$$a_0 + a_1x + a_2x^2 + \cdots + a_nx^n, \quad n \geq 0.$$

Это множество обозначается P. Если в множестве P определить обычные операции сложения и умножения, то это множество становится ассоциативным кольцом, которое называется кольцом многочленов.

Если кольцо R состоит не только из нуля, то такое кольцо не может быть группой по умножению, поскольку для любого $a \in R$ имеет место

$$a \cdot 0 = 0 \cdot a = 0.$$

Однако, если в R все отличные от нуля элементы составляют группу по умножению, то такое кольцо называется телом. Тело с коммутативной операцией умножения называется полем.

Примерами полей являются поля рациональных, действительных и комплексных чисел. Это все бесконечные поля, т.е. поля, содержащие бесконечное количество элементов. В дискретной математике большую роль играют конечные поля. Конечные поля еще называются полями Галуа. Самый простой (нетривиальный) пример конечного поля состоит из двух элементов

$$\mathbb{Z}_2 = \{0, 1\}.$$

В этом множестве введем операции сложения и умножения по следующим правилам

$+$	0	1
0	0	1
1	1	0

и

$\cdot$	0	1
0	0	0
1	0	1

Через $\mathbb{Z}_m$ обозначаются вычеты по модулю m, где $m \geq 2$. Это значит, что

$$\mathbb{Z}_m = \{0, 1, \ldots, m-1\},$$

сумма двух элементов $x, y \in \mathbb{Z}_m$ определяется, как

$$x + y = (x + y) \mod m,$$

где $z \mod m$ равно остатку от деления z на m. Например, для $x, y \in \mathbb{Z}_4$, где $x = 2$, $y = 3$ имеем

$$x + y = (2 + 3) \mod 4 = 1.$$

Аналогично для умножения элементов $x, y \in \mathbb{Z}_m$ имеем

$$xy = (xy) \mod m.$$

Для любого m множество является кольцом, но если m является простым числом, то $\mathbb{Z}_m$ будет полем, состоящим из конечного числа элементов.

4. Изоморфизмы алгебраических структур

Мощь алгебры состоит в том, что алгебраические структуры не зависят от природы множеств на которых они определены. Более того, различные алгебраические структуры (группоиды, полугруппы, группы, кольца и поля) могут быть неразличимы с точки зрения алгебраических свойств. Речь идет о том, что между объектами различной природы может быть установлено взаимно однозначное соответствие, которое сохраняет алгебраические операции.

Два группоида G и G' называются изоморфными, если существует такое взаимно однозначное отображение

$$\varphi : G \to G',$$

что для любых $a, b \in G$ имеет место равенство

$$\varphi(a \circ b) = \varphi(a) \circ \varphi(b).$$

Само это отображение φ называется изоморфизмом между G и G'. Разумеется в этом случае φ^{-1} есть изоморфизм между G' и G. Поэтому понятие изоморфизма группоидов — это симметричное понятие. Будем писать

$$G \simeq G'$$

для изоморфных группоидов. Легко видеть, что свойство изоморфности не только симметрично, но и рефлексивно и транзитивно, поэтому отношение изоморфности является отношением эквивалентности.

Покажем, что если групповая операция в группоиде G удовлетворяет условию ассоциативности, то для любого группоида G' изоморфного G групповая операция тоже ассоциативна. Для любых $a, b, c \in G$ имеем

$$a' = \varphi(a),\ b' = \varphi(b),\ c' = \varphi(c).$$

Далее,

$$(a' \circ b') \circ c' = (\varphi(a) \circ \varphi(b)) \circ \varphi(c) = \varphi(a \circ b) \circ \varphi(c) =$$

$$= \varphi((a \circ b) \circ c) = \varphi(a \circ (b \circ c)) = \varphi(a) \circ \varphi(b \circ c) =$$

$$= a' \circ (b' \circ c').$$

Аналогично доказывается, что свойства коммутативности, дистрибутивности и другие свойства сохраняются при изоморфизмах. Следовательно, изоморфным образом полугруппы или группы будет полугруппа или группа. Причем абелева полугруппа после изоморфизма останется абелевой полугруппой. При этом единичный элемент в группе G при изоморфизме переходит в единичный элемент в группе G'.

Рассмотрим пример изоморфизма для мультипликативной группы положительных чисел $G^{\times}$ и аддитивной группы всех действительных чисел G^{+}. Изоморфизм определим следующим образом

$$\ln : G^{\times} \to G^{+},$$

при этом

$$\ln(a \cdot b) = \ln a + \ln b.$$

Оказывается, что циклические группы одного порядка изоморфны между собой. Рассмотрим циклическую аддитивную группу классов вычетов $\mathbb{Z}_n$ порядка n. Покажем, что любая циклическая группа G_n порядка n изоморфна $\mathbb{Z}_n$. Единица $1 \in \mathbb{Z}_n$ перейдет в образующий элемент $a \in G_n$, далее

$$k \to a^k, \quad k = 0, 1, \ldots, n-1.$$

Бесконечные циклические группы тоже изоморфны между собой. Покажем, что любая бесконечная циклическая группа будет изоморфна аддитивной группе целых чисел $\mathbb{Z}$. Сама группа $\mathbb{Z}$ является бесконечной циклической с образующим элементом 1. Возьмем образующий элемент a в любой бесконечной циклической группе G_∞. И рассмотрим отображение

$$a^k \to k, \quad k = 0, \pm 1, \pm 2, \ldots.$$

Наконец, рассмотрим еще одно содержательно определение — изоморфное вложение. Группоид G изоморфно вкладывается в группоид H, если существует изоморфное отображение G на некоторый подгруппоид H' группоида H.

Глава V

Теория графов

1. Основные определения теории графов

Графы описывают структуры «кружочков со стрелочками», которые возникают в очень многих областях: в естественных и общественных науках и инженерных приложениях. Дадим формальное определение ориентированного графа. Пусть V — непустое множество, элементы которого называются вершинами графа. Обычно мы будем рассматривать конечное множество вершин, но в ряде случаев можно рассматривать и бесконечное множество вершин. Далее рассмотрим множество упорядоченных пар элементов множества V, которое будем обозначать E и называть множеством дуг графа. Таким образом,

$$V \subset E \times E.$$

Далее, пара $\langle V, E\rangle$ называется ориентированным графом порядка $n = |V|$. Будем обозначать также через V_G и E_G множество вершин и множество дуг графа G. В ориентированном графе дуга $e = (v_i, v_j)$ представляет собой стрелку, которая начинается в вершине v_i, которая называется началом дуги, и указывает на вершину v_j, которая называется концом дуги. Также будем говорит, что дуга исходит из вершины v_i и входит в вершину v_j. Ориентированные графы можно называть орграфами.

Если для каждой пары $(v_i, v_j) \in E$ существует пара $(v_j, v_i) \in E$, т.е. для каждой стрелочки существует стрелка в обратную сторону, тогда граф называется неориентированным, и пара дуг

$(v_i, v_j), (v_j, v_i)$ называется ребром графа и обозначается, как неупорядоченная пара $\{v_i, v_j\}$.

Вершины v_i и v_j называются смежными, если существует ребро $\{v_i, v_j\}$. Вершина $v \in V$ и ребро $\{v_i, v_j\}$ называются инцидентными, если $v = v_i$. Граф называется полным, если любые две вершины этого графа смежны.

В ориентированном графе $G = \langle V, E \rangle$ последовательность

$$v^1 e^1 v^2 e^2 \dots e^{k-1} v^k$$

такая, что каждая дуга e^n исходит из вершины v^n и входит в вершину v^{n+1}, называется путем в графе. Причем вершина v^1 называется началом пути, а вершина v^k концом пути. Для случая неориентированного графа путь называется маршрутом.

Путь (маршрут) называется замкнутым, если $v^1 = v^k$, т.е. начало пути совпадает с концом. В противном случае будем говорить о незамкнутом пути (маршруте). Незамкнутый путь (маршрут) называется цепью, если все дуги (ребра) в нем различны. Цепь, в которой и вершины различные, называется простой цепью. Замкнутый путь (маршрут), в котором все дуги (ребра) различные называется контуром (циклом), а контур (цикл), в котором различные вершины, называется простым контуром (циклом). Контур, состоящий из одной вершины $v^1 e^1 v^1$ называется петлей.

Вершина v называется достижимой из вершины u, если существует путь, начинающийся в вершине u и заканчивающийся в вершине v. Граф называется связным, если для любых двух вершин u и v вершина v достижима из u.

Еще рассмотрим понятие композиции путей (маршрутов) в графе. Пусть мы имеем два пути (маршрута) π_1 и π_2 такие, что конечная вершина в π_1 является начальной вершиной в π_2, тогда композицией путей (маршрутов) называется путь

$$v_1^1 e_1^1 v_1^2 e_1^2 \dots e_1^{k-1} v_1^k e_2^1 v_2^2 e_2^2 \dots e_2^{m-1} v_2^m,$$

где $\pi_1 = v_1^1 e_1^1 v_1^2 e_1^2 \dots e_1^{k-1} v_1^k$ и $\pi_2 = v_1^k e_2^1 v_2^2 e_2^2 \dots e_2^{m-1} v_2^m$.

Рассмотрим еще вопрос о способах задания графа. Самый интуитивно понятный способ состоит в рисовании кружочков (вершин) и соединения их стрелками (дугами). Для формального описания графа согласно определению необходимо задать множество вершин V и

подмножество $V \times V$ для определения множества дуг (ребер). Чтобы описать это подмножество часто используют матрицу смежности. Матрица смежности представляет собой квадратную матрицу $n \times n$, где n — порядок графа. Эта матрица задается следующим образом

$$a_{ij} = \begin{cases} 1 & \text{если } (v_i, v_j) \in E \\ 0 & \text{если } (v_i, v_j) \notin E \end{cases}$$

Для неориентированного графа матрица смежности является симметричной.

2. Операции над графами

Рассмотрим граф G. Граф H называется подграфом графа G, если выполнены следующие включения

$$V_H \subset V_G, \quad E_H \subset E_G.$$

Будем писать $H \subset G$. Если H подграф графа G и $V_H = V_G$ то подграф H называется остовным подграфом или фактором.

Пусть G некоторый граф, содержащий более одной вершины. Тогда для любой вершины $v \in V_G$ можно рассмотреть операцию удаления вершины из графа G. Будем говорить, что $H = G - v$ получается в результате удаления вершины v из графа G, если $V_H = V_G \setminus v$ и из множества дуг исходного графа удалены все дуги, которые начинались или заканчивались в вершине v. Можно рассмотреть и операции добавления вершины в граф, когда к множеству вершин добавляется новый элемент — новая вершина. Также можно определить операцию добавления новой дуги в существующем графе.

Рассмотрим два графа G и H. Граф F мы назовем объединением графов G и H, если

$$V_F = V_G \cup V_H, \quad E_F = E_G \cup E_H.$$

Будем использовать обозначение $F = G \cup H$.

Чуть более сложной операцией над графами является операция прямого произведения графов. Пусть G_1 и G_2 суть два графа. Прямым произведением этих графов называется граф

$$G = G_1 \times G_2.$$

Множество вершин графа G состоит из прямого (декартового) произведения множеств вершин графов G_1 и G_2

$$V_G = V_{G_1} \times V_{G_2},$$

а множество дуг E_G определяется следующим образом

$$((u_1, u_2), (v_1, v_2)) \in E_G$$

тогда и только тогда, когда или $u_1 = v_1$ и $(u_2, v_2) \in E_{G_2}$, или $u_2 = v_2$ и $(u_1, v_1) \in E_{G_1}$.

3. Нагруженные графы

Согласно определению граф — это пара множеств — вершин и дуг. Мы можем определить функции на этих множествах. Такие графы называются нагруженными графами. Пусть мы имеем граф $G = \langle V, E \rangle$. Этот граф называется нагруженным, если заданы следующие функции

$$L : V \to A_V,$$

$$M : E \to A_E,$$

где A_V и A_E суть произвольные множества. Наиболее часто рассматривается случай, когда $A_E = \mathbb{R}$, т.е. каждой дуге приписано числовое значение. Такая ситуация возникает, когда мы рассматриваем длину дуги в графе.

Множество A_V, как правило, описывает предметную область в математической модели. Это может быть множество городов, этапов выполнения работы, предприятий, людей и т.д.

Рассмотрим случай неориентированного нагруженного графа, который будем считать связным. Предположим, что

$$A_E = \{x \in \mathbb{R} : x > 0\},$$

т.е. каждому ребру приписано положительное число. В этом случае для любой пары вершин графа u и v можно определить расстояние или метрику $d(u, v)$ следующему правилу

$$d(u, v) = \min_{\pi(u,v)} \sum_{n=1}^{k} M(e^n),$$

где минимум берется по всем цепям, которые начинаются в вершине u, а заканчиваются в вершине v, сумма берется по всем ребрам, которые входят в эту цепь.

Покажем, что так определенное расстояние является метрикой, т.е. выполнены следующие условия.

1. $d(u, v) \geq 0$, причем $d(u, v) = 0$ тогда и только тогда, когда $u = v$;
2. $d(u, v) = d(v, u)$;
3. $d(u, v) \leq d(u, w) + d(w, v)$ для любой вершины $w \in V$.

Первое условие отражает факт, что расстояние неотрицательное и равно нулю только для совпадающих вершин. Это условие выполнено по определению функции $d(u, v)$. Второе условие, которое означает симметричность расстояния. В нашем случае это условие выполнено вследствие неориентированного графа. Наконец третье условие — условие треугольника. Это свойство следует из определения минимальной цепи.

Расстояние в нагруженном неориентированном графе является длиной кратчайшего пути между двумя вершинами. Вопрос о фактическом вычислении кратчайшего пути в графе имеет принципиальное значение во многих приложениях. Опишем алгоритм для нахождения кратчайшего пути.

Будем рассматривать, вообще говоря, ориентированный нагруженный граф, где

$$A_E = \{x \in \mathbb{R} : x \geq 0\}.$$

В данном случае мы допускаем нулевой вес для дуги. Через $w(u, v)$ мы обозначаем вес дуги $(u, v) \in E$.

Начальную и конечную вершину мы обозначим через s и t соответственно. Будем использовать обозначение (s, t)-путь. Рассмотрим алгоритм Дейкстры для поиска кратчайшего пути. Этот алгоритм является итерационным. На каждом шаге итерации каждая вершина u графа имеет метку $m(u)$, которая может быть постоянной или временной. В случае постоянной метки значение $m(u)$ равно кратчайшему (s, u)-пути. В случае же, когда метка $m(u)$ является временной, то ее значение — это вес кратчайшего пути, проходящего только через вершины с постоянными метками. Если на какой-либо итерации метка становится постоянной, то она остается постоянной

и в течение всего алгоритма. Поскольку нас будет интересовать не только значение минимального пути, но и сам кратчайший путь, то мы будем использовать еще одну метку $p(u)$ для вершин графа. На каждой итерации $p(u)$ является номером вершины, предшествующей u в (s,u)-пути, имеющем минимальный вес среди всех (s,u)-путей, проходящих через вершины, имеющие постоянные метки. В конце работы алгоритма с помощью этих меток можно получить кратчайший путь.

Алгоритм Дейкстры поиска кратчайшего пути.

1. Вершине s назначаем постоянную метку $m(s)=0$, метки всех остальных вершин являются временными и равны значению ∞. Положим $r=s$.

2. Для всех не помеченных вершин v, в которые есть дуги из вершины r выполняем следующее: если

$$m(v) > m(p) + w(r,v),$$

то положим

$$m(r) = m(r) + w(r,v)$$

и

$$p(v) = r.$$

3. Пусть V' есть множество вершин с временными метками. Находим вершину v^* такую, что

$$m(v^*) = \min_{v \in V'} m(v).$$

При этом метку $m(v^*)$ считаем постоянной.

4. Положим $r=v^*$. Если $r \neq t$, то перейти к шагу 2.

5. Конец работы алгоритма. Значение $m(t)$ есть вес минимального пути, а

$$\pi(s,t) = s, \ldots, p^3(t), p^2(t), p(t), t$$

есть минимальный путь, где

$$p^k(t) = p(p^{k-1}(t)),$$

т.е. $p^k(t)$ — это вершина из которой вершина t достигается через k дуг.

4. Деревья

Среди графов выдающееся значение имеют графы со специальной структурой, которые называются деревьями. Деревьями могут быть как ориентированные, так и неориентированные графы, но их определения несколько различны.

Неориентированный граф называется неориентированным деревом, если он связный и не имеет циклов.

Ориентированный граф $G = \langle V, E \rangle$ называется ориентированным деревом, если он связный, в нем существует ровно одна вершина $v^0 \in V$, в которую не входят никакие дуги, и в каждую вершину $v \in V \setminus v^0$ входит ровно одна дуга. При этом вершина v^0 называется корнем дерева.

Рассмотрим более подробно сначала неориентированные деревья. Эти графы обладают рядом характерных свойств, каждое из которых может быть положено в определение дерева. Будем рассматривать дерево $G = \langle V, E \rangle$.

1. В дереве для любых двух вершин существует единственный маршрут, соединяющий эти вершины.

2. Количество ребер $|E|$ ровно на 1 меньше количества вершин $|V|$.

3. Если у дерева удалить любое ребро, то этот граф станет несвязным.

4. При добавлении нового ребра в дерево, это дерево станет графом, имеющим ровно один цикл.

Вершина в неориентированном графе называется висячей, если у нее есть ровно одно ребро. Если из дерева удалить одну висячую вершину и ребро, которое имеет эта вершина, то мы снова получим неориентированное дерево.

Большие приложения имеют ориентированные деревья, которые имеют собственную терминологию. Вершина называется листом, если из нее не выходит ни одна дуга. Лист является корнем только в том случае, когда дерево состоит из одной вершины. Путь из корня до любого листа называется ветвью дерева. Максимальная длина ветви называется высотой дерева. Глубиной вершины называется длина

пути из корня до этой вершины. Если из вершины u идет дуга в v, вершина u называется родительской по отношению к v, а вершина v называется дочерней по отношению к u.

Ориентированное дерево задает частичный порядок на множестве вершин. Напомним, что порядок называется частичным, если для некоторых элементов множества задано отношение порядка, т.е. рефлексивное, антисимметричное и транзитивное. Элементы, которые связаны отношением порядка называются сравнимыми. Для ориентированного дерева две вершины u и v будут сравнимыми, если они принадлежат одной из ветви дерева. Для сравнимых вершин будем писать

$$u \geq v,$$

если существует путь из u в v и

$$u \leq v,$$

если существует путь из v в u.

Также как и для неориентированного дерева, если в неориентированном дереве удалить вершину, являющуюся листом[1], и дугу, входящую в эту вершину, мы получим снова ориентированное дерево.

[1]Предполагается, что эта вершина не является корнем.

Глава VI

Конечные автоматы

1. Абстрактные конечные автоматы

Абстрактные математические автоматы позволяют формализовать устройства или объекты, которые имеют вход, внутреннюю память и выход. При этом выход или реакция автомата зависит от входной информации и внутреннего состояния автомата, которое также меняется под воздействием входных сигналов.

Мы будем рассматривать дискретные конечные автоматы. Конечный автомат может оперировать только конечными множествами и имеет конечную внутреннюю память. Дискретность автомата означает, что он может обрабатывать входные сигналы, которые возникают в дискретные моменты времени. При этом предполагается, что действия автомата происходят мгновенно.

Итак, мы будем рассматривать дискретную временную ось

$$t_0, t_1, t_2, \ldots,$$

где t_n — это моменты времени такие, что

$$t_0 < t_1 < t_2 < \ldots.$$

Поскольку для дискретных автоматов совершенно не важно, что происходит в между этими выделенными моментами времени, то без ограничения общности будем считать, что

$$t_n = n, \quad n = 0, 1, 2, \ldots.$$

Входные сигналы конечного автомата состоят из символов конечного алфавита

$$A = \{a_1, a_2, \ldots, a_{N_A}\}.$$

Внутренняя память конечного автомата или множество внутренних состояний — это тоже конечное множество

$$Q = \{q_1, q_2, \ldots, q_{N_Q}\}.$$

Далее реакция автомата на сигналы входного алфавита описывается конечным множеством, называемым выходным алфавитом

$$B = \{b_1, b_2, \ldots, b_{N_B}\}.$$

Используем обозначение $q(n)$ для состояния конечного автомата в момент времени n. Будем считать, что в начальный момент автомат находится в некотором фиксированном внутреннем состоянии $q(0) \in Q$. В каждый момент времени на вход конечному автомату подается один из символов входного алфавита. Таким образом, мы имеем последовательность входных сигналов. Обозначим через $a(n)$ символ входного алфавита, который поступает в момент времени $n \geq 1$. Соответственно, через $b(n)$ будем обозначать выходной сигнал в момент времени $n \geq 1$ из выходного алфавита. Логика нашего конечного автомата будет описываться следующими уравнениями

$$b(n) = f(q(n-1), a(n)),$$

$$q(n) = g(q(n-1), a(n)).$$

В этих уравнениях мы используем функции перехода

$$f : Q \times A \to B$$

и

$$g : Q \times A \to Q.$$

Таким образом, конечным автоматом называется шестерка

$$K = \langle A, B, Q, q(0), f, g \rangle.$$

Будем использовать следующую запись для работы конечного автомата

$$b(n) = K(a(n)).$$

Примерами конечных автоматов могут быть почти все цифровые электронные устройства, например, компьютеры. Разумеется, в современном компьютере невозможно представить мощность множества, описывающего внутренние состояние, но формально — это конечный автомат. Примерами автоматов, которые не являются конечными автоматами — это машины с механическими регуляторами, устройства с истинными случайными явлениями[1], по-видимому, конечными автоматами не являются живые существа.

Приведем пример конечного автомата. Пусть множество входного алфавита, множество внутренних состояний и выходной алфавит имеют следующий вид

$$A = B = Q = \{1, 2, \dots, 100\}.$$

В начальный момент $q(0) = 1$. Зададим функции перехода следующим образом

$$f(q, a) = \begin{cases} q + a, & \text{если } q + a \leq 100, \\ 1, & \text{если } q + a > 100. \end{cases}$$

Функция g задается аналогично образом

$$g(q, a) = \begin{cases} q + a, & \text{если } q + a \leq 100, \\ 1, & \text{если } q + a > 100. \end{cases}$$

Если на вход автомату мы будем подавать последовательность

$$a = (1, 1, \dots),$$

то на выходе у нас будет последовательность

$$b = (2, 3, 4, \dots, 99, 100, 1, 2 \dots)$$

Теперь мы запустим наш автомат, когда выходной символ будет подаваться на вход на следующем такте. Это возможно, поскольку у нас входной и выходной алфавиты совпадают. Рассмотрим следующую схему работы конечного автомата

$$b(0) = 1,$$

[1]Генератор случайных чисел в компьютере является на самом деле генератором псевдослучайных чисел.

$$b(n) = K(b(n-1)), \quad n = 1, 2, \ldots, 9.$$

Тогда на выходе мы будем получать первые девять членов последовательности чисел Фибоначчи:

$$2, 3, 5, 8, 13, 21, 34, 55, 89.$$

2. Конечно автоматные языки

Рассмотренные конечные автоматы в предыдущем параграфе были автоматами-преобразователями, поскольку их работа состояла в том, чтобы по входным сигналам генерировать выходные сигналы. Есть и еще одна «работа» для конечных автоматов — это распознавания слов в формальном языке. Такие конечные автоматы называются автоматами-распознавателями.

Алфавитом в формальном языке называется конечное множество

$$\Lambda = \{a_1, a_2, \ldots, a_N\},$$

где $a_n \in \Lambda$ суть различимые символы, которые иногда называются также буквами алфавита Λ. Словом в этом алфавите называется конечная упорядоченная последовательность символов из алфавита. Все возможные слова в алфавите Λ обозначаются Λ^*.

Среди всех возможных слов в алфавите можно выделить те слова, которые имеют смысл. Таким образом языком называется множество слов

$$L \subset \Lambda^*.$$

Разумеется, множество L может быть бесконечным, но не более чем счетным. Знание языка означает возможность распознавать слова в этом языке. Естественные языки распознаются их носителями интуитивно, а какие языки можно использовать в кибернетических устройствах, какие языки можно распознавать с помощью компьютеров. Одним из важнейших классов формальных языков является класс конечно автоматных языков.

Для этого модифицируем определение конечного автомата. Конечным автоматом-распознавателем называется пятерка

$$K_L = \langle \Lambda, Q, q(0), f, \Phi \rangle,$$

где Λ — алфавит распознаваемого языка, Q — конечное множество внутренних состояний, $q(0) \in Q$ — начальное состояние автомата,

$$f : Q \times \Lambda \to Q$$

есть функция перехода, $\Phi \subset Q$ множество заключительных состояний.

Работа автомата-распознавателя состоит в том, что ему на вход подается последовательность букв из алфавита

$$A = (a(1), a(2), \ldots, a(M))$$

автомат меняет свои внутренние состояния по следующему правилу

$$q(n) = f(q(n-1), a(n)), \quad n = 1, 2, \ldots, M.$$

Если в момент времени $n = M$ автомат находится в состоянии из множества Φ, т.е.

$$q(M) \in \Phi,$$

то слово A принимается или распознается автоматом K_L. В противном случае слов не распознается автоматом.

Каждый конечный автомат порождает язык, который состоит из слов, распознаваемых этим конечным автоматом

$$L_K = \{A \in \Lambda^* : A \text{ распознается автоматом } K_L\}.$$

Можно поставить вопрос и обратно. Язык L называется конечно автоматным, если существует такой конечный автомат-распознаватель, что множество распознаваемых автоматом слов совпадает с языком L.

Приведем пример. Пусть алфавит состоит из букв

$$\Lambda = \{a, m, p\}.$$

Множество внутренних состояний будет состоять из шести элементов

$$Q = \{-1, 0, 1, 2, 3, 4\}.$$

В начальный момент времени автомат находиться в состоянии $q(0) = 0$. Состояние $q = -1$ будет идентифицировать ошибку в слове, а множество

$$\Phi = \{4\}$$

описывает заключительное состояние.

Зададим функцию перехода следующим образом

$$\begin{aligned}
&f(-1,a) = f(-1,m) = f(-1,p) = -1,\\
&f(0,a) = -1,\\
&f(0,m) = f(0,p) = 1,\\
&f(1,a) = 2,\\
&f(1,m) = f(1,p) = -1,\\
&f(2,a) = -1,\\
&f(2,m) = f(2,p) = 3,\\
&f(3,a) = 4,\\
&f(3,m) = f(3,p) = -1,\\
&f(4,a) = f(4,m) = f(4,p) = -1.
\end{aligned}$$

Легко убедиться, что язык нашего конечного автомата состоит из четырех слов

$$L_K = \{mama, papa, mapa, pama\}.$$

3. Клеточные автоматы и другие обобщения

Среди конечных автоматов большое значение имеют автономные автоматы, которые имеют свою собственную эволюцию в зависимости от начального состояния и собственных правил перехода. При этом такой автомат не зависит от входных сигналов, роль которых сводится только к отсчету тактов времени. Такие конечные автоматы называются автоматами Мура.

В определении конечного автомата мы не конкретизировали каких-либо геометрических соотношений между внутренними состояниями, но в некоторых ситуациях имеет смысл учитывать геометрические соотношения.

Рассмотрим квадратную сетку, состоящую из $N \times N$ клеток. Будем считать, что эта сетка периодическая по горизонтали и вертикали, т.е. по сути является тором. Введем координаты

$$(i,j), \quad i,j = 1,\ldots,N.$$

Каждая клетка может находиться в одном из конечного числа состояний. Пусть эти состояния занумерованы от 1 до K, тогда мы будем

обозначать значение состояния в ячейке (i, j) через

$$q(i, j) \in \{1, 2, \ldots, K\}.$$

Таким образом, внутреннее состояние конечного автомата описывается всеми значениями состояний в каждой ячейки. Множество всех возможных состояний обозначим через Q.

Зададим динамику клеточного автомата, как функцию

$$F : Q \to Q,$$

т.е. последующее состояние зависит только от текущего состояния. Рассмотрим более подробно какие переходные функции используются в клеточных автоматах.

Для каждой ячейки (i, j) мы введем понятие окрестности этой ячейки, как множество следующих ячеек

$$(i, j), (i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1),$$

$$(i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1).$$

Поскольку мы рассматриваем периодическую структуру, то мы предполагаем, что

$$(0, j) = (N, j), (N+1, j) = (1, j),$$
$$(i, 0) = (i, N), (i, N+1) = (i, 1).$$

Таким образом, окрестность любой ячейки состоит из самой ячейки и восьми соседних ячеек. Обозначим это множество через $\Omega(i, j)$. А множество

$$\omega(i, j) = \Omega(i, j) \setminus (i, j)$$

называется окружением клетки (i, j).

В клеточном автомате значение ячейки (i, j) зависит только от значений в ячейках их окрестности этой ячейки

$$F(i, j) = F(\Omega(i, j)).$$

В качестве самого известного клеточного автомата можно рассмотреть игру в «Жизнь», которая была придумана английским математиком Дж. Конвеем в 1970 году. Каждая ячейка может иметь лишь два состояние «живая» или «мертвая». Функция перехода задается следующим образом:

1. Если в окрестности пустой клетки есть ровно три живые клетки, то эта клетка становится живой.

2. Если живая клетка имеет в окружении меньше двух живых клеток или больше трех, то эта клетка становится мертвой.

3. Во всех остальных случаях значение клетки сохраняется.

В оригинальной игре «Жизнь» пространство клеток предполагается бесконечным во все стороны, но таком случае клеточный автомат перестает быть конечным.

Одним из очевидных обобщений понятия конечного автомата является отказ от конечности для множеств входного и выходного алфавита, а также для множества внутренних состояний. Такие автоматы называются просто дискретными автоматами. Во многих приложениях отказ от конечности позволяет более гибко описывать автоматы. Например, использование действительных чисел во входных сигналах и при описании внутренних состояний.

Другим обобщением конечного автомата является отказ от детерминированности конечных автоматов. Так, при в качестве переходных функций можно рассматривать элементы случайности, что тоже часто оказывается адекватным моделируемой задаче. Такой конечный автомат называется вероятностным автоматом.

Глава VII

Алгоритмы и машины

1. Понятие алгоритма

Понятие алгоритма в настоящее время прочно вошло в нашу жизнь и уже является не только специальным термином, но и бытовым словом. Под алгоритмом понимают заданную последовательность определенных действий, выполнение которых приведет к заданной цели. Примерами алгоритмов в быту являются различные кулинарные рецепты, а также описания процедуры установки программного обеспечения и описание подключения бытовых приборов.

Более точно алгоритм можно описать, как четко сформулированная последовательность действий, включающая в себя условные переходы, которая применяется к исходным данным, в результате чего на выходе (по завершению алгоритма) мы получаем выходные данные (результат). При этом принципиально, чтобы выполнение этого алгоритма было автоматическим, т.е. последовательность действий не должна требовать интуиции, каких-либо дополнительных данных и т.д. Кстати, тут уже видно, что кулинарный рецепт не полностью является алгоритмом, поскольку во много зависит от искусства хозяйки. Результат алгоритма не должен зависеть от его исполнителя. Алгоритм должен выполняться автоматически.

Можно привести пример — алгоритм нахождения вещественных корней квадратного уравнения, где на вход алгоритму подаются три

целых[1] числа a, b, c, которые являются коэффициентами уравнения

$$ax^2 + bx + c = 0,$$

и $a \neq 0$. Дальше нужно вычислить дискриминант $D = b^2 - 4ac$ и в зависимости от его знака получить один из трех вариантов ответа: два действительных решения, одно кратное действительное решение или нет действительных решений.

Любая компьютерная программа является записью алгоритма. Более того, любой алгоритм *желательно* описать с помощью компьютерной программы, поскольку именно такое представление алгоритма гарантирует однозначность и четкость.

Отметим, что в определении алгоритма мы не требуем, чтобы этот алгоритм выполнялся для всех входных данных. Например, алгоритм может «зацикливаться» для некоторых исходных данных.

Еще важное уточнение. Коль скоро мы требуем механического исполнения алгоритма, то мы должны беспокоиться о том, чтобы данные, которыми оперирует алгоритм допускали механическую обработку. Например, входные данные являлись конечной последовательностью символов некоторого конечного алфавита. Причем, мы считаем, что эти символы алфавита являются четко различимыми. В частности, входные данные могут быть заданы конечными последовательностями целых чисел или последовательностью битов $\{0, 1\}$.

Последнее замечание сразу же приводит к тому, что оказывается алгоритмы не могут работать с действительными (иррациональными) числами! Поскольку иррациональное число представляет собой непериодическую десятичную дробь, то его, вообще говоря, не всегда можно описать конечным набором символов (цифр). Об этой принципиальной проблеме мы еще поговорим в разделе, посвященном конструктивным действительным числам.

Определение алгоритма, которое мы используем является использует различные интуитивные понятия, поэтому это определение не может быть использовано в математических рассуждениях. В частности, это определение не позволяет доказывать утверждения о том, что какая-либо задача является алгоритмически неразрешимой. Речь идет не об отсутствии решений, а об отсутствии алгоритма, который может решить задачу. Например, пусть один алгоритм печатает по-

[1]Почему не действительных будет понятно в дальнейшем.

следовательность цифр после запятой. Тогда задача определения равенства нулю этого числа, как мы увидим, является алгоритмически неразрешимой. Поэтому в начале прошлого века возникла необходимость в строгом определении понятия алгоритма.

Существуют различные определения алгоритма, которые принято называть «уточнениями понятия алгоритма», но эти уточнения оказываются эквивалентными.

2. Уточнения понятия алгоритма

Говоря об алгоритме, мы всегда будем иметь в виду, что этот алгоритм представляет собой процедуру вычисления некоторой функции

$$f : \mathbb{N} \to \mathbb{N},$$

поскольку, как мы уже отмечали, входные и выходные данные алгоритма должны быть конечными словами конечного алфавита.

Поскольку в настоящее время основам программирования учат в школе, то мы будем предполагать, что читатель имеет представление о каком-нибудь языке программирования. Дадим следующее определение алгоритма. Алгоритмом называется корректная программа, написанная на каком-либо языке программирования (C++, Pascal, C#, Python,...), которая вычисляет целочисленную функцию.

Сделаем несколько замечаний.

Во-первых, мы будем предполагать, что данная программа может использовать потенциально неограниченную память при создании переменных, массивов и т.д.

Во-вторых, нужно помнить, что переменные любого типа в языках программирования, например, *double* все равно могут представлять лишь конечное число значений.

В-третьих, необходимо учитывать, что программа не должна обращаться ни к каким данным операционной системы или внешним файлам. В частности, нельзя обращаться к часам.

В-четвертых, нужно понимать, что встроенные функции тоже являются частью программы математические, поэтому их использование не влияет на определение алгоритма. Отсутствие часов делает невозможным использование случайных чисел.

Разумеется, для точного определения следует еще добавить полное описание языка программирования, но это лишь технический момент. Тем более, что можно описать и самый примитивный язык программирования и использовать его.

3. Машина Тьюринга

Сейчас мы опишем исторически одну из первых конструкций для уточнения понятия алгоритма — машину Тьюринга. Это не просто язык программирования, но и формализация процесса выполнения программы на мыслимой «мыслимой» машине. Важно, что эта конструкция возникла до первых ЭВМ и являлась прологом к созданию настоящих компьютеров.

Машина Тьюринга состоит из бесконечной в обе стороны ленты, которая разделена на ячейки с номерами

$$\ldots, -2, -1, 0, 1, 2, \ldots$$

и головку машины, которая может быть строго над одной ячейкой и последовательно перемещаться влево или вправо по ленте[2]. Ячейка, над которой находится головка, называется текущая ячейка. Каждая клетка на ленте может или пустой или содержать один из символов конечного алфавита A, а головка может находиться в одном из состояний из конечного множества Q.

В начальный момент на ленте лишь конечное число ячеек содержит какой-либо символ, а головка находиться над ячейкой с номером 0 и в состоянии $q_0 \in Q$. Также в множестве состояний выделено подмножество $F \subset Q$ финальных состояний. Работа машины Тьюринга состоит в следующем:

1. Если головка находится в состоянии $q \in F$, то конец работы машины, иначе переход к следующему шагу.

2. В зависимости от значения текущей ячейки и состояния головки:

   (a) в текущую ячейку записывается символ из A или ячейка становится пустой;

[2]Под перемещением мы понимаем сдвиг головки на соседнюю ячейку.

(b) головка переходит в одно из состояний из множества Q;

(c) головка либо остается на месте, либо перемещается вправо или в лево.

3. Переход к шагу 1.

Входные данные для алгоритма, реализованного машиной Тьюринга, кодируются значениями в ячейка на ленте. Если машина Тьюринга остановилась, что означает конец работы алгоритма, то результат этого алгоритма будет записан в виде конфигурации символов на ленте.

При этом логика, которая отвечает за шаг 2, называется программой для машины Тьюринга. А сама головка машины есть конечный автомат.

Существует большое количество эквивалентных модификаций, например, можно использовать лишь одно направленную ленту или машину с несколькими параллельными лентами и т.д. Можно в качестве алфавита рассматривать лишь один символ $A = \{\times\}$, которым помечаются ячейки[3].

Программирование на машина Тьюринга — занятия весьма трудоемкое и нудное, поэтому мы предпочитаем иметь дело с языками программирования высокого уровня. Но машина Тьюринга важна с теоретической точки зрения, поскольку формализует не только язык программирования, но и сам процесс вычислений.

Почти каждый программист в свое время писал собственный эмулятор машины Тьюринга и если считать, что программа на языке программирования может иметь бесконечные массивы, то легко видеть, что любой алгоритм, реализованный на машине Тьюринга, может быть реализован и на языке программирования. На самом деле верно и обратное — любую программу можно реализовать на подходящей машине Тьюринга.

Теперь можно сформулировать принципиально важное заявление, которое называется тезис Тьюринга-Черча: *Всякий алгоритм может быть реализован в виде соответствующей машине Тьюринга.*

Это утверждение на может быть доказанным, поскольку использует неформальное понятие алгоритма. По этой же причине тезис не

[3]По-сути, наш алфавит содержит еще один символ — «пустой символ».

является и аксиомой. Тезис Тьюринга-Черча следует понимать как строгое определение алгоритма, которое может быть уже использовано в математических доказательствах.

Впрочем, этот тезис приводит к небезобидным выводам. Например, утверждению, что даже в арифметике существуют теоремы, которые нельзя ни доказать, ни опровергнуть[4]! При этом эти теоремы могут быть верными, но недоказуемыми. Не говоря уже о том, что нас окружает огромное число алгоритмически неразрешимых проблем. Тезис Тьюринга-Черча ставит четкую границу возможностей нашего разума с точки зрения логики.

Вообще говоря, неочевидно — ограничиваются ли вычислительные способности человека тезисом Тьюринга-Черча также, как ограничены возможности для числовых компьютеров.

4. Разрешимость и перечислимость

Используя понятие алгоритма согласно тезису Тьюринга-Черча, мы будем говорит, что функция

$$f : D(f) \subset \mathbb{N} \to \mathbb{N}$$

заданная на множестве $D(f)$ является вычислимой или эффективно вычислимой, если существует реализация этой функции с помощью алгоритма. При этом для значений $x \in \mathbb{N} \setminus D(f)$ алгоритм никогда не заканчивает свою работу.

Множество $A \subset \mathbb{N}$ называется разрешимым, если индикаторная функция

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

вычислима. Для вычислимого множества существует алгоритм, с помощью которого можно определить принадлежит конкретное число этому множеству или нет.

Множества не являющиеся вычислимыми называется невычислимыми множествами. А существуют ли невычислимые множества? Конечно, поскольку всех возможных подмножеств $\mathbb{N}$ несчетное число, а множество алгоритмов, как раз, именно счетно, то существует несчетное количество неразрешимых множеств.

[4]Теорема Геделя.

Легко видеть, что объединение и пересечение разрешимых множеств есть разрешимое множество. Также дополнение к разрешимому множеству является разрешимым множеством.

Если для для какого-нибудь множества $A \subset \mathbb{Z}$ существует такой алгоритм, т.е. вычислимая функция $f(n)$ такая, что

$$A = \{f(n) : n \in D(f)\},$$

то множество A называется перечислимым. Это означает, что существует такая вычислимая функция, которая перечисляет все элементы множества. Порядок перечисления множества, разумеется, значения не имеет.

Любое разрешимое множество является также и перечислимым множеством, поскольку перебирая все числа из $\mathbb{N}$ с помощью характеристической функции можно перечислить все элементы разрешимого множества.

Пусть теперь перечислимое множество $A \subset \mathbb{N}$ имеет перечислимое дополнение $\mathbb{N} \setminus A$, тогда множество A является разрешимым множеством. Действительно, для построения вычислимой характеристической функции необходимо использовать два алгоритма, перечисляющих множество A и его дополнение. Любое $n \in \mathbb{N}$ обязательно будет перечисленно одним из этих алгоритмов, поэтому существует вычислимая процедура для выяснения принадлежит ли n множеству A.

Принципиальным моментом является тот факт, что существуют множества перечислимые, но не разрешимые.

5. Конструктивные действительные числа

Поскольку математика должна быть объективной, то возникает соблазн использовать только вычислимые процедуры, которые потенциально могут быть выполнены (проверены) автоматически. Таким образом возникает идея конструктивной математики. Конструктивная математика, развитая, в частности, в работах А.А. Маркова, признает только конструктивные объекты и конструктивные (вычислимые) процедуры. Конструктивная математика имеет свои важные достижения, но в ряде вопросов является менее удобной. Рассмотрим вопросы построения конструктивных действительных чисел.

Заметим, что множества $\mathbb{N}$ и $\mathbb{Z}$ являются конструктивными, поскольку любое рациональное число представляется в виде дроби из

целых чисел, то и множество рациональных чисел является конструктивным. Как известно, любое действительное число можно с любой наперед заданной точностью аппроксимировать рациональным числом. Но в этом определении есть неконструктивный момент, связанный с фразой «можно... аппроксимировать». Для конструктивного определения этой фразе нужно придать значение «существует алгоритм».

Действительное число a называется конструктивным, если существует такой алгоритм, который каждому номеру n соотносит рациональное число q_n такое, что

$$|a - q_n| \leq \frac{1}{n}.$$

Таким образом, конструктивное действительное число — это такое число, для которого существует конструктивная рациональная аппроксимация с заданной погрешностью.

Возникают два вопроса. Первый — а существуют ли неконструктивные действительные числа? Действительно, все приходящие на ум иррациональные числа (рациональные, очевидно, конструктивные): π, e, $\sqrt{2}$, $\sin(1)$ являются конструктивными. Ответ — неконструктивные действительные числа существуют и их значительно больше, чем конструктивных! Любое случайное действительное число с вероятностью единица будет неконструктивным. Почему? Очень просто — количество конструктивных чисел не больше чем множество всех возможных алгоритмов, но множество алгоритмов (согласно тезису Черча-Тьюринга) счетно, а множество действительных чисел — несчетно.

Второй вопрос, который возникает, такой — если неконструктивных чисел так много, то можно ли привести пример? Но если бы можно было на бумаге написать такое число, то оно было бы конструктивным.

К проблемам, которые возникают при использовании конструктивных числе, относится простейшая задача определения равенства двух конструктивных действительных чисел друг другу. В самом деле, не существует алгоритма, который для любых двух конструктивных действительных чисел сможет определить равны они или нет.

Глава VIII

Теория игр

1. Понятие игры

Многие математические проблемы могут быть сформулированы, как задачи оптимизации, т.е. поиска оптимального решения среди возможных вариантов. Однако в теории игр рассматриваются такие постановки задач, когда необходимо решать задачи оптимизации в условиях противодействия. И это кардинально меняет ситуацию.

В теории игр рассматриваются два и более игроков, каждый из которых имеет собственные возможности в игре, которые называются стратегиями игроков. Далее, каждый игрок имеет собственную цель, которая в свою очередь описывается функцией выигрыша игрока. Коллизия состоит в том, что результат игры для каждого игрока зависит не только от его выбора собственной стратегии, но и от выбора стратегий всех игроков.

Истоки теории игр лежат в анализе салонных игр, но достаточно быстро модели на основе теории игр стали применяться в экономических и военных проблемах. В настоящее время теория игр кроме того применяется в социально-политических науках, в биологии, в математической статистики, искусственном интеллекте, математическом моделировании и других дисциплинах.

Введем формальное определение игры.

Обозначим через I множество всех игроков. Мы будем рассматривать конечное число игроков. Мы будем различать игроков по номерам

$$I = \{1, 2, \ldots, N\}.$$

Предположим, что каждый игрок $i \in I$ имеет в своем распоряжении определенное множество стратегий, которое мы обозначим через S_i.

Процедура игры происходит следующим образом: каждый игрок выбирает одну стратегию из своего множества стратегий $s_i \in S_i$. Вектор выбранных стратегий всех игроков обозначим через

$$s = (s_1, s_2, \ldots, s_N).$$

Вектор s называется ситуацией в игре. Множество всех возможных ситуаций можно ввести по формуле

$$S = \prod_{i \in I} S_i.$$

В каждой сложившейся ситуации игроки получают определенные выигрыши. Договоримся считать, что выигрыш может быть и отрицательным, что означает проигрыш. Выигрыш игрока i в ситуации s обозначим через $H_i(s)$. Функция H_i, определенная на множестве всех ситуаций

$$H_i : S \to \mathbb{R}$$

называется функцией выигрыша i-го игрока. Мы будем измерять выигрыши действительными числами, хотя не всегда выигрыш может быть измерен числом.

Бескоалиционной игрой называется система

$$\Gamma = \langle I, \{S\}_{i \in I}, \{H_i\}_{i \in I} \rangle,$$

где I, S_i являются множествами, а H_i — функции на множестве S, принимающие вещественные значения.

Наиболее часто встречается ситуация, когда сумма выигрышей всех игроков во всех ситуациях является постоянной, что соответствует тому, что игроки по сути делят между собой фиксированную сумму. Игра называется игрой с постоянной суммой, если

$$\sum_{i \in I} H_i(s) = const$$

при всех ситуациях $s \in S$.

2. Антагонистичные игры

Важнейшим классом игр являются антагонистичные игры. Игра называется антагонистичной, если число игроков равно двум, т.е. $I = \{1, 2\}$, а значения функций выигрыша в сумме равны нулю

$$H_1(s) = -H_2(s),\ s \in S.$$

Если в теории оптимизации основной задачей является нахождения оптимальных решений, то в теории игр аналогом этого является нахождения ситуации равновесия. Ситуация $s^* \in S$ называется ситуацией равновесия в игре, если ни одному из игроков не выгодно отступать от этой стратегии. формально это можно записать следующей формулой $s^* = (s_1^*, s_2^*)$

$$H_1(s_1, s_2^*) \leq H_1(s_1^*, s_2^*) \leq H_1(s_1^*, s_2),\ s \in S.$$

Если множества стратегий конечны, то антагонистичные игры удобно записывать в матричном виде. Пусть множество стратегий первого игрока равно $n > 1$, а второго — $m > 1$, тогда запишем в виде матрицы значения функции выигрышей

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Игра в этом случае состоит в том, что первый игрок выбирает строку, а второй игрок (одновременно!) выбирает столбец. Число, стоящее на пересечении выбранных строки и столбца, означает выигрыш первого игрока и проигрыш второго игрока.

В матричной игре ситуация (i^*, j^*) называется равновесной, если

$$a_{ij^*} \leq a_{i^*j^*} \leq a_{i^*j}$$

для всех $i = 1, \dots, m$ и $j = 1, \dots, n$. В теории игр доказывается, что для существования ситуации равновесия необходимо и достаточно, чтобы было выполнено равенство

$$\max_i \min_j a_{ij} = \min_j \max_i a_{ij} = c.$$

Число c в этом случае называется ценой игры. Если бы в каждой игре существовала бы ситуация равновесия, то игры бы не имели смысл. К счастью или к сожалению, но во многих играх ситуации равновесия не существует. Самый простой пример — игра в «чет–нечет». Матрица этой игры такова

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Фундаментальным результатом теории игр является тот факт, что любая матричная игра имеет ситуацию равновесия в смешанных стратегиях. Смешанной стратегией называется случайная величина, значениями которой являются стратегии игрока. Смешанная стратегия — это распределение вероятностей на множестве допустимых стратегий, которую можно представить вектором с неотрицательными компонентами, сумма которых равна единице.

При смешанном расширении понятия матричной игры, игроки выбирают свои смешанные стратегии: первый игрок

$$X = (x_1, \ldots, x_m),\ x_i \geq 0,\ \sum_{i=1}^{m} x_i = 1,$$

$$Y = (y_1, \ldots, y_m),\ y_i \geq 0,\ \sum_{i=1}^{m} y_i = 1,$$

Выигрыш в смешанных расширениях рассчитывается как математическое ожидание. Выигрыш первого игрока равен

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_i y_j.$$

Использование смешанных стратегий основывается на возможном многократном повторении игр, поскольку в качестве функции выигрыша используется математическое ожидание, т.е. среднее значение выигрыша.

В матричной игре с матрицей выигрышей A имеет место

$$\max_X \min_j (X A_{\cdot j}) = v_\Gamma = \min_Y \max_i (A_{i\cdot} Y),$$

где $A_{i\cdot}$ обозначает i-ую строку, а $A_{j\cdot}$ — j-ый столбец. В этом равенстве внешние экстремумы достигаются на оптимальных смешанных стратегиях, а величина v_Γ называется ценой игры.

3. Методы решения игр

Решить игру — это означает найти оптимальные, вообще говоря, смешанные стратегии и цену игры. Мы будем рассматривать антагонистичные игры, которые задается платежной матрицей.

Пусть A есть платежная $(m \times n)$ матрица с положительными элементами

$$a_{ij} > 0, \quad i = 1, \ldots, m; j = 1, \ldots, n.$$

Этого условия всегда можно добиться, если к каждому элементу матрицы прибавить достаточно большое число.

Через $e^{(k)}$ обозначим вектор из $\mathbb{R}^k$, состоящий из единичных элементов. Рассмотрим двойственные задачи линейного программирования

$$\min xe^{(m)}, \quad xA \geq e^{(n)}, \quad x \geq 0 \tag{VIII.1}$$

и

$$\min ye^{(n)}, \quad Ay \leq e^{(m)}, \quad y \geq 0. \tag{VIII.2}$$

Можно показать, что эта задача имеет решение x^* и y^*, причем

$$x^*e^{(m)} = y^*e^{(n)} = V > 0.$$

Тогда оптимальные смешанные стратегии находятся по следующим формулам

$$\overline{x} = \frac{x^*}{V}, \quad \overline{y} = \frac{y^*}{V}.$$

Цена игры находится по формуле

$$v_\Gamma = \frac{1}{V}.$$

Решение задач линейного программирования при большой размерности имеет значительные технические трудности, поэтому рассмотрим еще один метод, позволяющий находить решение матричной игры.

Рассмотрим итеративный метод фиктивного разыгрывания[1]. Идея этого метода состоит в том, что на каждой итерации игрок выбирает такую стратегию, которая максимизирует результат при условии, что соперник выбирает стратегию ту же, что на предыдущей итерации.

[1]Этот метод также называется методом Брауна-Робинсона.

Пусть A произвольная платежная $m \times n$ матрица. На первой итерации оба игрока выбирают произвольные стратегии, например, первые по номеру. Далее, предположим, что за k итераций первый игрок использовал i-ю стратегию p_i^k раз, а второй игрок использовал j-ю стратегию q_j^k раз. Тогда в $(k+1)$ итерации первый игрок будет использовать i_{k+1}-ю стратегию, а второй игрок j_{k+1} стратегию так, что выполнены равенства

$$\overline{v}^k = \max_i \sum_{j=1}^{m} a_{ij} q_j^k = \sum_{j=1}^{m} a_{i_{k+1}j} q_j^k$$

и

$$\underline{v}^k = \min_j \sum_{i=1}^{n} a_{ij} p_i^k = \sum_{i=1}^{n} a_{ij_{k+1}} p_i^k.$$

Можно показать, что векторы

$$x^k = (p_1^k/k, p_2^k/k, \ldots, p_m^k/k)$$

и

$$y^k = (q_1^k/k, q_2^k/k, \ldots, q_n^k/k)$$

сходятся к оптимальным смешанным решениям при $k \to \infty$.

Этот процесс несложно реализовать на компьютере, поэтому этот метод можно рекомендовать для задач с большой размерностью. Сходимость этого метода, очевидно, не высока, но за счет простоты можно выполнять большое количество итераций и добиваться хороших результатов.

Литература

[1] *Андерсон Дж.А.* Дискретная математика и комбинаторика. М. Издательский дом «Вильямс», 2004.

[2] *Курош А.Г.* Лекции по общей алгебре. М. Наука, 1973.

[3] *Нефедов В.Н., Осипова В.А.* Курс дискретной математики. М. Изд-во МАИ, 1992.

[4] *Шамин Р.В.* Функциональный анализ от нуля до единицы. М.: ЛЕНАНД/URSS, 2016.

[5] *Шамин Р.В.* Математические вопросы волн-убийц. М.: ЛЕНАНД/URSS, 2016.

[6] *Яблонский С.В.* Введение в дискретную математику. М.: Наук, 1986.