Государственный комитет связи и информатизации Украины
Одесская национальная академия связи им. А.С.Попова

Кафедра высшей математики

КУРС ДИСКРЕТНОЙ МАТЕМАТИКИ

Учебное пособие для всех специальностей

УТВЕРДЖЕНО
Методическим советом академии
Протокол № 15 от 09. 04. 2002г.

Одеса 2002

УДК 681.3

План УМИ 2002 г.

Составитель – ***В.М. Плотников***

Рецензент – ***Л.И. Соколов***

Редактор – ***А.М. Харсун***

Компьютерная правка
и макетирование – ***Е.С. Корнейчук***

В курсе изучаются фундаментальные понятия, лежащие в основе математической кибернетики и таких разделов математики, как алгебра, теория графов, математическая логика. Хотя данные разделы и являются отдельными математическими теориями, они не только используют методы друг друга, но и тесно связаны между собой. Все их можно объединить условным названием «дискретная математика». Этот термин характеризует конструктивный характер теории, алгоритмические, комбинаторные методы. Задачи дискретной математики, как правило, тесно связаны с компьютерными проблемами, естественно выражаются в виде различных алгоритмов. Настоящее пособие содержит также задания для самостоятельной работы, вопросы к зачету по курсу, примеры и алгоритмы решения задач с использованием компьютера.

ОДОБРЕНО
на заседании кафедры
высшей математики и
рекомендовано к печати
Протокол № 5
« 11 » октября 2002 г.

СОДЕРЖАНИЕ

Бог всемогущ,
поэтому он может создать такой камень,
который сам не сможет поднять.
Но Бог всемогущ,
поэтому он может поднять любой камень.

1. ТЕОРИЯ МНОЖЕСТВ

Под множеством понимается совокупность каких-либо объектов, называемых элементами множества. Теория множеств занимается изучением свойств как произвольных множеств, так и множеств специального вида независимо от природы образующих их элементов. Терминология и многие результаты этой теории широко используются в математике, например, в математическом анализе, геометрии и теории вероятностей.

Множество является основным строительным материалом математики. Создатель теории множеств Г. Кантор определил множество как *«объединение в одно целое объектов, хорошо различимых нашей интуицией или нашей мыслью»*. Математическое понятие множества постепенно выделилось из привычных представлений о совокупности, собрании, классе, семействе и т.д. На место наивного подхода к понятию множества пришел аксиоматический подход [1, 2]. Аксиомы теории множеств постулируют существование некоторых неопределяемых или примитивных объектов, называемых множествами, вместе с символами и аксиомами, регулирующими их использование. Здесь существует аналогия с геометрией, где задаются неопределяемые объекты, называемые точками и плоскостями, вместе с системой аксиом, связывающих эти объекты друг с другом. Наиболее распространенными являются аксиомы теории множеств Цермело-Френкеля [1].

Терминология. Если каждый элемент множества B является элементом множества A, то множество B называется подмножеством множества A. Например, если множество A состоит из чисел 1, 2 и 3, то у него существует 8 подмножеств (три из них содержат по 1 элементу, три – содержат по 2 элемента, одно подмножество, по определению, есть само множество A и восьмое подмножество – это пустое множество, не содержащее ни одного элемента). Запись $x \in A$ означает, что x – элемент множества A, а $B \subset A$ – что B является подмножеством множества A. Если универсальное множество, из которого мы берем элементы всех множеств, обозначить через I, то элементы, принадлежащие I, но не входящие в A, образуют множество, называемое дополнением множества A и обозначаемое $C(A)$ или A'. Множество, не содержащее ни одного элемента, называется пустым множеством.

Над множествами можно производить операции, напоминающие операции, производимые в арифметике над числами. Объединением $A \cup B$ множеств A и B называется множество, состоящее из всех элементов, принадлежащих хотя бы одному из множеств A и B (элемент, принадлежащий множествам A и B одновременно засчитывается при включении в $A \cup B$ только один раз). Пересечением $A \cap B$ множеств A и B называется множество, состоящее из всех элементов, принадлежащих как A, так и B. Предположим, например, что множество I состоит из всех букв русского алфавита, A – из всех согласных, а множество B – из букв, встречающихся в слове «энциклопедия». Тогда объединение $A \cup B$ состоит из всех букв алфавита, кроме *а*, *ё*, *у*, *ъ*, *ь*, *ы*, *ю*, пересечение $A \cap B$ – из букв *д*, *к*, *л*, *н*, *п*, *ц*, а дополнение $C(A)$ – из всех гласных. Раздел теории множеств, который занимается исследованием операций над множествами, называется алгеброй множеств. Пустое множество играет в алгебре множеств роль нуля, и поэтому его часто обозначают символом $\varnothing$; например, $A \cup \varnothing = A$, $A \cap \varnothing = \varnothing$.

Сравнение множеств. Если из элементов двух множеств можно составить пары таким образом, чтобы каждому элементу первого множества соответствовал определенный элемент второго множества, а каждому элементу второго множества соответствовал один и только один элемент первого множества, то говорят, что между такими двумя множествами установлено взаимно однозначное соответствие. Чтобы установить взаимно однозначное соответствие, необязательно пересчитывать элементы множеств. Например, мы знаем, что американские штаты находятся во взаимно однозначном соответствии с их столицами, хотя можем оставаться в неведении относительно общего их числа. Мы могли бы утверждать: «Столиц штатов ровно столько, сколько штатов». Между двумя конечными множествами можно установить взаимно однозначное соответствие тогда и только тогда, когда оба множества состоят из одного и того же числа элементов. В теории множеств аналогичные утверждения используются, даже когда множества содержат бесконечно много элементов. Если между двумя множествами можно установить взаимно однозначное соответствие, то говорят, что они имеют одинаковое количество элементов или равномощны. Если же при любом способе образования пар некоторые элементы из первого множества остаются без пары, то говорят, что первое множество содержит больше элементов, чем второе, или, что первое множество имеет большую мощность. С понятием мощности связаны, казалось бы, удивительные результаты. Например, на первый взгляд положительных целых чисел в два раза больше, чем четных положительных чисел, так как четно каждое второе число. Но, согласно теории множеств, четных положительных чисел столько же, сколько всех положительных целых чисел. Действительно, можно образовать пары чисел 2 и 1, 4 и 2, 6 и 3 и, вообще каждому четному числу $2n$ поставить в соответствие целое число n. Именно это обстоятельство имел в виду Б. Рассел (1872–1970), сформулировав факт, названный им парадоксом Тристрама Шенди. Герой романа Стерна сетовал на то, что ему потребовался целый год, чтобы изложить события первого дня его жизни, еще один год понадобился, чтобы описать второй день, и что при таком темпе он никогда не завершит свое жизнеописание. Рассел возразил, заметив, что если бы Тристрам Шенди жил вечно, то смог бы закончить свое жизнеописание, так как события n-го дня Шенди мог бы описать за n-й год и, таким образом, в летописи его жизни ни один день не остался бы не запечатленным. Иначе говоря, если бы жизнь длилась бесконечно, то она насчитывала бы столько же лет, сколько дней. Эти примеры показывают, что бесконечное множество можно поставить во взаимно однозначное соответствие со своим бесконечным подмножеством. Иногда это свойство принимают за определение бесконечного.

Если можно установить взаимно однозначное соответствие между некоторым множеством и множеством положительных целых чисел, то говорят, что такое множество счетно. Для обозначения количества элементов в счетном множестве часто используют символ $\aleph_0$ (алеф-нуль). Так называемые «трансфинитные» числа, например $\aleph_0$, могут не подчиняться обычным законам арифметики. Например, так как существует $\aleph_0$ четных чисел, $\aleph_0$ нечетных и $\aleph_0$ целых чисел, то приходится признать, что $\aleph_0 + \aleph_0 = \aleph_0$. Идея сравнения множеств путем установления взаимно однозначного соответствия между ними используется в различных разделах математики. Число всех действительных чисел, как показал основатель научной теории множеств Г.Кантор (1845–1918), больше, чем $\aleph_0$ чисел. Следовательно, если можно показать, что множество действительных чисел, обладающих некоторым особым свойством, является всего лишь счетным множеством, то заведомо должны существовать действительные числа, этим свойством не обладающие. Например, так как множество алгебраических чисел счетно, должны существовать неалгебраические числа. Такие числа называются трансцендентными.

Поразительная и далеко не очевидная теорема, высказанная в качестве гипотезы Кантором и доказанная Э. Шрёдером и Ф. Бернштейном около 1896, утверждает, что если можно установить взаимно однозначное соответствие между множеством A и подмножеством множества B, и между множеством B и подмножеством множества A, то существует взаимно однозначное соответствие между всем множеством A и всем множеством B.

Парадоксы. Мы уже упоминали о том, что в теории множеств встречаются такие утверждения, как парадокс Тристрама Шенди, которые выглядят противоречащими здравому смыслу. Эти парадоксы возникают просто потому, что теория множеств, подобно многим математическим и физическим теориям, облекает свои идеи в обычные слова, вкладывая в них особый смысл. Однако существуют и парадоксы, возникающие из-за внутренних логических трудностей самой теории множеств. Обильным источником парадоксов такого типа служит широко распространенная практика задания множества путем указания некоторого свойства его элементов, например, «множество, состоящее из английских слов, содержащих менее 19 букв».

Некритическое использование такого рода определений может привести к трудностям. Например, некоторые статьи в этой энциклопедии содержат ссылки на себя, другие таких ссылок не содержат. Мы могли бы включить в нашу энциклопедию дополнительную статью, состоящую только из перечня статей, не содержащих ссылок на себя. Принадлежала бы такая статья множеству статей, не содержащих ссылок на себя, или не принадлежала бы? Любой ответ противоречил бы отличительному свойству, которым по их определению наделены элементы множества. Это – одна из форм так называемого парадокса Рассела, названного в честь своего автора Бертрана Рассела. «Множество всех множеств» – еще одно понятие, также приводящее к парадоксу. Существование парадоксов показывает, с какой осторожностью следует пользоваться терминологией теории множеств. Тем не менее теория множеств настолько полезна, что большинство математиков не хотели бы отказываться от нее. Было затрачено много усилий, чтобы развить методы, позволяющие исключить возникновение парадоксов в теории множеств. В приложениях теории множеств к другим разделам математики универсальное множество I обычно само является некоторым определенным множеством и парадоксальные ситуации здесь не возникают.

Аксиома выбора. Неожиданные трудности в теории множеств могут возникнуть, казалось бы, в самых простых случаях. Если, например, задано семейство непересекающихся множеств, ни одно из которых не пусто, то интуитивно кажется очевидным, что мы можем построить новое множество, содержащее ровно по одному элементу из каждого множества, входящего в это семейство. Но если наше семейство содержит бесконечно много множеств, то для построения нового множества может потребоваться бесконечное число произвольных выборов, а законность такого процесса при тщательном анализе становится отнюдь не очевидной. Аксиома выбора, утверждающая, что такое множество существует, была впервые сформулирована в 1904 г. Э. Цермело (1871–1953). До сих пор не удалось показать, что аксиома выбора следует из остальных аксиом теории множеств. Но около 1938 г. К. Гёдель (1906–1978) показал, что если теория множеств непротиворечива (т.е. не содержит внутренних противоречий) без аксиомы выбора, то она остается непротиворечивой и после присоединения к ней аксиомы выбора.

1.1. ***Отношение* –** произвольное подмножество R множества A^n всех кортежей (упорядоченных наборов) вида $\langle a_1,...,a_n\rangle$, где $a_1,...,a_n$ – элементы некоторого множества A; в этом случае говорят, что R есть n-местное отношение на A. Понятие *отношения* служит в математике для выражения на теоретико-множественном языке связей между объектами. Множество всех таких элементов a, которые входят хотя бы в один кортеж, принадлежащий отношению R называется *полем* этого отношения. Двухместные отношения называются *бинарными*. Если R – бинарное отношение, то вместо $\langle a, b\rangle \in R$, часто пишут aRb. Частным случаем понятия *отношения* является *соответствие*.

Через $\in$ обозначается *отношение принадлежности*, т.е. $x \in A$ означает, что элемент x принадлежит множеству A.

Если x не является элементом множества A, то это записывается $x \in A$.

Два множества A и B считаются равными, если они состоят из одних и тех же элементов. Пишется $A = B$, если A и B равны, и $A \gamma B$ в противном случае.

Через $\subseteq$ обозначается *отношение включения* множеств, т.е. $A \subseteq B$ означает, что каждый элемент множества A является элементом множества B. В этом случае A называется *подмножеством* B, а B – *надмножеством* A. Если $A \subseteq B$ и $A \gamma B$, то A называется *собственным подмножеством* B, и в этом случае пишем $A \subset B$.

Множество, не содержащее элементов, называется пустым и обозначается через $\varnothing$.

Объединением (ДИЗЪЮНКЦИЕЙ, СУММОЙ) множеств A и B называется множество

$$A \cup B = \{x \,/\, x \subseteq A \text{ или } x \in B\}.$$

Пересечением (КОНЪЮНКЦИЕЙ) множествA и B называется множество

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

Разностью множеств A и B называется множество

$$A \,|\backslash\, B = \{x \,/\, x \in A \text{ и } x \notin B\}.$$

Симметрической разностью множеств A и B называется множество

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

Мощностью (*или кардинальным числом*) множества называется количество элементов в нем.

Некоторое, общее для всех множеств данной мощности, надмножество, называется универсальным множеством или УНИВЕРСУМОМ и обозначается обычно как U. Разность $U \setminus A$ называется *дополнением* множества A и обозначается через $\overline{A}$.

Соответствие, бинарное отношение между двумя множествами A и B – произвольное подмножество R *декартова произведения* $A \beta B$.

Если $a \notin A$, $b \notin B$ и $(a, b) \notin R$, то пишут также $R(a, b)$ или aRb. Если $R = \varnothing$ – пустое множество, то соответствие называется *пустым*, а если $R = A \beta B$, то соответствие называется *полным*.

Пусть $R \subseteq A$ β B. *Областью определения* Dom R называется множество элементов $a \in A$, для каждого из которых найдется хотя бы один элемент $b \in B$ такой, что aRb. *Областью значений, или образом*, Im R соответствия R называется множество элементов $b \in B$, для каждого из которых найдется хотя бы один элемент $a \in A$ такой, что aRb. Соответствие R называется *всюду определенным*, если Dom $R = A$, и *сюръективным*, если Im $R = B$.

Для каждого $a \in A$ множество элементов $b \in B$ таких, что aRb, называется *образом* относительно R и обозначается $\mathrm{im}_R\, a$. *Прообразом элемента* $b \in B$ относительно R называется множество элементов $a \in A$ таких, что aRb; прообраз обозначается $\mathrm{coim}_R\, b$. Ясно, что

$$\mathrm{Im}\, R = \cup_{a \in A} \mathrm{im}_R\, a, \quad \mathrm{Dom}\, R = \cup_{b \in B} \mathrm{coim}_R\, b.$$

Каждое соответствие однозначно определяет функцию a (ρ) $\mathrm{im}_R\, a$, которая отображает множество A в множество подмножеств B. Обратно, всякая функция f из множества A в множество подмножеств B определяет некоторое соответствие $R(f)$: $aR(f)b$ тогда и только тогда, когда $b \in f(a)$. Указанные сопоставления взаимно однозначны, что позволяет рассматривать соответствия как частично определенные многозначные функции.

Для конечных множеств A и B широко используются матричное и графовое представления соответствия. Пусть $A = \{a_1, a_2, ..., a_n\}$, $B = \{b_1, b_2, ..., b_m\}$ и $R \subseteq A$ βB. Соответствию R сопоставляется матрица размера n β m, строки которой помечены элементами из A, столбцы – элементами из B, а на пересечении строки a_i и столбца b_j стоит 1, если $a_i\, R\, b_j$, и 0 в противном случае. Например,

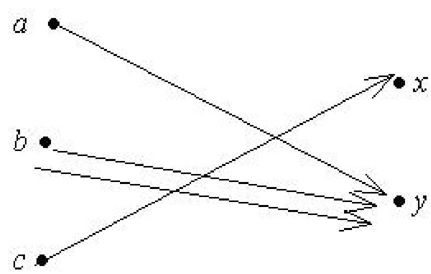
$$A = \{a, b, c\},\ B = \{x, y\},$$
$$R = \{(a, x), (a, y), (b, y), (c, x)\}.$$

Тогда R сопоставляется матрица

$$\begin{array}{c|c} & \begin{matrix} x & y \end{matrix} \\ \hline \begin{matrix} a \\ b \\ c \end{matrix} & \begin{Vmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{Vmatrix} \end{array}$$

Каждая матрица подобного вида однозначно определяет соответствие между A и B.

При графовом представлении элементы множеств A и B изображаются точками на плоскости. Обычно эти точки обозначаются теми же символами, что и соответствующие элементы. Точки a и b соединяются направленной дугой от a к b, если aRb. Например, соответствие $R = \{(a, x), (a, y), (b, y), (c, x)\}$ изобразится следующим ориентированным графом:



Всякое соответствие $R \subseteq A \subseteq \beta\ B$ устанавливает так называемое ***соответствие Галуа*** между подмножествами множества A и подмножествами множества B. Именно, если $X \subseteq A$, то через $\Gamma\ (X)$ обозначается пересечение $\cap_{a \in X} \mathrm{im}_R\, a$; аналогично, для $Y \subseteq B$ вводится множество $\Gamma^{-1}(Y) = \cap_{b \cap Y} \mathrm{coim}_R\, b$.

Пусть $X^* = \Gamma^{-1}(\Gamma\ (X))$, $Y^* = \Gamma\ (\Gamma^{-1}(Y))$,

тогда $X \subseteq X^*$, $Y \subseteq Y^*$;

из $X_1 \subseteq X_2$ следует

$$\Gamma\ (X_1) \supseteq \Gamma\ (X_2);$$

из $Y_1 \subseteq Y_2$ следует $\Gamma^{-1}(Y_1) \supseteq \Gamma^{-1}(Y_2)$; $X^{**} = X^*$; $Y^{**} = Y^*$.

Подмножество $X \subseteq A$ ($Y \subseteq B$) называется *замкнутым*, если $X = X^*$ ($Y = Y^*$). Соответствие Галуа устанавливает биективное соответствие между замкнутыми подмножествами в A и B.

Множество A: {Лена, Петя, Маша, Вася, Женя, Эллочка}

Множество B: {Горький, Достоевский, Лермонтов, Некрасов, Пушкин, Толстой, Фет}

Декартово произведение AxB – это множество всех пар вида (Лена, Горький) и т.д.

Множество R (подмножество множества AxB): (Лена, Некрасов); (Лена, Фет); (Петя, Горький); (Петя, Пушкин); (Петя, Толстой); (Маша, Пушкин); (Маша, Лермонтов); (Вася, Пушкин); (Вася, Достоевский); (Женя, Фет); (Вася, Толстой); (Женя, Горький).

Поставим вопрос об общности интересов. Выберем:

Множество X: {Маша, Вася}.

Множество im $_R$ Маша: {Пушкин, Лермонтов}, Множество im $_R$ Вася: {Достоевский, Толстой, Пушкин}.

Кстати, множество im $_R$ Эллочка: $\varnothing$

Множество $\Gamma(X) = \mathrm{im}_R$ Маша $\cap\ \mathrm{im}_R$ Вася: {Пушкин}

Найдем подмножество со сходными интересами:

Множество $X^* = \Gamma^{-1}(\Gamma(X)) = \Gamma^{-1}$({Пушкин}) = coim $_R$ Пушкин: {Петя, Маша, Вася}.

Поскольку соответствия между A и B являются подмножествами одного и того же множества $A\ \beta\ B$, можно говорить об их объединении, пересечении и дополнении. Таким образом, все соответствия между A и B образуют булеву алгебру, изоморфную булевой алгебре соответствия между B и A. Этот изоморфизм задается с помощью операции инволюции, или обращения соответствия: если $R \subseteq A\ \beta\ B$, то инволюция $R^{\#}$ состоит из таких пар (b, a), что $(a, b) \in R$. Иногда вместо $R^{\#}$ пишут R^{-1}. Ясно, что $R^{\#\#} = R$.

Помимо теоретико-множественных операций для соответствия вводится операция умножения. Если $R \subseteq A\ \beta\ B$, $S \subseteq B\ \beta\ C$, то произведение RS состоит из таких пар (a, c), для которых найдется элемент $b \in B$ такой, что $(a, b) \in R$,

$(b, c) \in S$. Умножение соответствий ассоциативно и обладает следующими свойствами: если $R_1 \subseteq R_2$, то $R_1S \subseteq R_2S$; $(RS)^{\#} = R^{\#}S^{\#}$. Кроме того, специальные соответствия – *отношения тождества*, или *диагонали*, Δ_A, состоящие из всех пар (a, a), $a \in A$, играют роль единиц для умножения соответствий. Именно, если $R \subseteq A\ \beta\ B$, то $\Delta_A R = R = R\Delta_B$.

Для всякого соответствия R выполнено включение $R \subseteq RR^{\#}R$. Если вместо включения выполняется равенство, то соответствие называется *дифункциональным*. Соответствие $R \subseteq A\ \beta\ B$ называется *функциональным*, если $RR^{\#} \supseteq \Delta_A$ и $R^{\#}R \subseteq \Delta_B$. Функциональные соответствия – это в точности графики функций из A в B. Всякое соответствие R представимо в виде $R = F^{\#}G$, где F, G – функциональные соответствия.

Пусть $R \subseteq A\ \beta\ A$. Соответствие R называется: a) *рефлексивным*, если $R \supseteq \Delta_A$; b) *симметричным, если* $R = R^{\#}$; c) *антисимметричным, если* $R \cap R^{\#} \subseteq \Delta_A$; d) *асимметричным, если* $R \cap R^{\#} = \varnothing$; e) *транзитивным, если* $R^2 \subseteq R$. Отношение R называется: a) *предпорядком*, если оно рефлексивно и транзитивно; b) *порядком*, если оно рефлексивно, транзитивно и антисимметрично; c) *толерантностью*, если оно рефлексивно и симметрично; d) *эквивалентностью*, если оно рефлексивно, симметрично и транзитивно.

Если R – эквивалентность, то множество A распадается на непересекающиеся классы эквивалентных элементов, или на классы эквивалентности. Множество классов эквивалентности называется *фактормножеством* множества A по отношению R. Всякое дифункциональное соответствие $R \subseteq A\ \beta\ B$ задает на своих областях определения и значений отношения эквивалентности, фактормножества по которым равномощны.

В определении соответствия можно считать A и B однотипными математическими структурами, для которых можно построить декартово произведение с той же структурой, и считать R подструктурой декартова произведения $A\ \beta\ B$. Тогда получаются определения соответствия между группами, кольцами, векторными пространствами и т.п.

2.ИЗОМОРФИЗМ, АВТОМОРФИЗМ И ГОМОМОРФИЗМ

Изоморфизм **(от греческих слов** τσοζ *– равный и* μορφη *– образ, вид, форма) это одно из основных понятий современной математики, которое исторически возникло сначала в пределах алгебры в применении к таким алгебраическим системам, как группы, кольца, поля и др., но оказавшееся принципиально существенным для общего понимания строения и структуры самых разных систем.*

Пусть даны две системы объектов S и $S^{/}$, причем в первой системе S определены отношения $F_k\,(x_1, x_2, \ldots)$, $\forall k=1, 2, \ldots, n$, а во второй системе $S^{/}$ определены отношения $F^{/}_k\,(x^{/}_1, x^{/}_2, \ldots)$, $\forall k=1, 2, \ldots, n$. Системы S и $S^{/}$ с указанными для них здесь отношениями называются *изоморфными*, если между ними существует такое взаимно однозначное соответствие $x'=\varphi(x)$, $x=\psi(x')$, где x произвольный элемент системы S, а x' произвольный элемент системы S', что из наличия $F_k(x_1, x_2, ...)$ вытекает $F'_k(x'_1, x'_2, ...)$, и наоборот. Отображение φ называется в этом случае *изоморфным отображением* или *изоморфизмом* системы S на систему S', а обратное ему отображение ψ *изоморфизмом системы* S', на систему S. Факт изоморфности систем S и S' *обозначается следующим образом:* $S \cong S'$.

Термин *изоморфизм* был введён в середине XIX в.; окончательно современные представления утвердились с 1918 года благодаря работам немецкой ученой Эмми (Амали) Нётер (*Noether Amalie Emmy*, родилась 23.3.1882 г. в Эрлангене, умерла 14.4.1935 г. в Брин-Море, США).

Первоначально понятие изоморфизма возникло в теории групп, где впервые был понят тот факт, что изучение внутренней структуры двух изоморфных систем объектов представляет собой одну и ту же задачу. Существование этого свойства впервые подметил ещё в начале XVII века великий французский философ и математик Рене Декарт (Descartes Rene, родился 31.3.1596 г., в Лаэ, Турень, умер 11.2.1650 г., в Стокгольме, автор, среди прочего, декартовой системы координат и фразы «*я мыслю, следовательно, я существую*»).

Аксиомы любой математической теории определяют систему объектов, изучаемую этой теорией всегда всего лишь с точностью до изоморфизма: аксиоматически построенная математическая теория, применимая к какой-либо одной системе объектов, всегда полностью применима к другой, изоморфной ей. Поэтому каждая аксиоматически изложенная математическая теория допускает не одну, а много интерпретаций или моделей.

Автоморфизм (от греческих слов αυτοζ – *сам и* μορφη) *изоморфизм некоторой системы объектов на себя.*

Все автоморфизмы алгебраической системы S образуют *группу* относительно умножения отображений. Эта группа обозначается *Aut S*.

Например, всякое невырожденное линейное преобразование векторного пространства V является автоморфизмом этого пространства. При этом *Aut V* подгруппа в полугруппе всех (т.е. в том числе вырожденных) линейных преобразований пространства *V*.

Среди автоморфизмов групп выделяются *внутренние автоморфизмы*. Внутренний автоморфизм группы G это такой автоморфизм φ, для которого $\varphi(x)=g^{-1}xg$, где g некоторый фиксированный элемент из G. Автоморфизм, не являющийся внутренним, называется *внешним*. Совокупность всех внутренних автоморфизмов группы G является *нормальной подгруппой* в *Aut G*.

Аналогично с помощью обратимых элементов определяются внутренние автоморфизмы *ассоциативного кольца с единицей* и внутренние автоморфизмы *полугруппы с единицей* (*моноида*). Группа автоморфизмов *Aut S* несет в себе существенную информацию о строении системы *S*.

Гомоморфизм (от греческих слов ομοζ – равный, одинаковый и μορφη) отображение множества элементов одной алгебраической системы в (в т. ч. на) другую, сохраняющее все значимые отношения (в частности, все операции).

Понятие гомоморфизма является обобщением понятия изоморфизма (в общем случае *гомоморфизм* это отображение одной алгебраической системы, вообще говоря, *в* другую; в частном же случае, когда имеет место взаимно-однозначное отображение одной алгебраической системы *на* другую, это и есть *изоморфизм*).

Например, гомоморфизм группы G в группу *H* это такое отображение φ, при котором каждому элементу $g \in G$ поставлен в соответствие определенный элемент $h=\varphi(g)\in H$ (который называется *образом g*, а *g* в этом случае называется *прообразом h*), причем произведению двух элементов g_1 и g_2 из *G* соответствует произведение их образов: $\varphi(g_1*g_2)=\varphi(g_1)*\varphi(g_2)$.

Если алгебраическая система имеет нулевую подсистему (т.е., например, в группе это единица точнее: 1 в мультипликативной группе или 0 в аддитивной группе), то множество элементов первой системы, отображающихся отображением φ в нулевую подсистему второй, называется *ядром гомоморфизма* φ.

Термин *гомоморфизм* ввел в употребление еще в конце XIX в. профессор Берлинского университета Георг (Фердинанд) Фробениус (Frobenius Ferdinand Georg), современное определение дано в 1929 г. профессором МГУ Эмми Нётер.

Обобщением понятия гомоморфизма является понятие морфизма в ***теории категорий***. В некоторых разделах математики термин гомоморфизм употребляется вместо термина *морфизм* и наоборот.

Категория (от греческого κατηγορια) – совокупность однотипных математических структур (объектов) и отображений (морфизмов) между этими структурами, в которых выполняется ряд естественных дополнительных условий.

Ob – класс объектов категории ***C***.

Mor C – класс морфизмов категории ***C***.

Категория множеств: ***St*** (или ***Ens***).

Эпиморфизм (от греческих слов επι – *на, над, при, после и* μορη) или, что то же самое, *сюръективное отображение* (*сюръекция) множества A* на множество *B* отображение *f* такое, что образ *A* есть все *B*, т.е. $f(A)=B$. Вместо *f* «сюръективно» говорят также « *f* есть отображение множества *A на* множество *B*». Вообще, для сохранения общности часто говорят: «*эпиморфизм в*», «*изоморфизм в*», «*автоморфизм*», но, разумеется, подразумевают: «*эпиморфизм на*», «*изоморфизм на*», «*автоморфизм на*», что непосредственно вытекает из определений соответствующих морфизмов. Так, во всех этих случаях образом отображения является все множество, т.е. для каждого элемента *B* существует прообраз в отображаемом множестве *A*. *Эпиморфизм это также морфизм категории, на который можно сокращать слева*.

Эндоморфизм (от греческих слов ενδον – внутри и μορφη) гомоморфизм алгебраической системы *в* (в т. ч. и *на*) себя. Все эндоморфизмы произвольной алгебраической системы относительно операции последовательного выполнения отображений образуют полугруппу с единицей (тождественный эндоморфизм). *Эндоморфизм векторного пространства* называется также *линейным преобразованием*.

Мономорфизм (от греческих слов μονοζ – *один и* μορφη) или, что то же самое, *инъективное отображение* (*инъекция) множества A* в множество *B* отображение, при котором различные элементы из *A* имеют различные образы в *B. Инъективное отображение называют также взаимно однозначным отображением множества A в множество B или вложением. Мономорфизм это также морфизм категории, на который можно сокращать справа*.

Биморфизм (от латинского *bi двойной, двоякий и* μορφη) или, что то же самое, *биективное отображение* (*биекция) морфизм категории, на который можно сокращать как слева, так и справа, то есть мономорфизм и эпиморфизм одновременно.*

Не следует путать термин гомоморфизм с созвучным математическим термином ≈ гомеоморфизм, который является частным случаем изоморфизма и представляет собой одно из основных понятий топологии, не являющейся отраслью дискретной математики. Тем не менее, полезно знать, что два топологических пространства называются гомеоморфными, если существует взаимно однозначное непрерывное отображение одного из них на другое такое, что обратное отображение тоже непрерывно; при этом само отображение называется гомеоморфизмом (от греческого ομοιοζ – подобный). *Например*, любой круг гомеоморфен любому квадрату, любые два отрезка гомеоморфны, но отрезок не гомеоморфен ни окружности, ни прямой. С другой стороны, прямая гомеоморфна любому интервалу. Свойства фигур, которые не меняются при переходе к гомеоморфным фигурам, называются *топологическими*. Примеры топологических свойств: *компактность* (*бикомпактность*) и *связность*:

гомоморфизм
все *A*] (в, на) *B*
эндоморфизм
все A] (в, на) *A*
изоморфизм
A Z B (взаимно-однозначно)
автоморфизм

А *Z A*(взаимно-однозначно)
эпиморфизм (сюръекция)
все *A*] (на) *B*
мономорфизм *(инъекция, вложение)*
все A] (в, на) *B* (взаимно-однозначно)
биморфизм* (биекция) = *мономорфизм* + *эпиморфизм

A	B	C	D	E	F	G	H
№ зач.	Фам.	Имя	Гр.	Студ	Нап.	Задолж.	баллы
96001	Айрян	Анна	1	Айрян	Брунь	---	---
96004	Асеев	Петр	1	Айрян	Вовк	---	75
96007	Баева	Алла	1	Айрян	Верба	---	88
96010	Бакин	Иван	1	Айрян	Гизе	---	---
96013	Бежин	Антон	1	Айрян	Даева	---	44
96016	Белов	Илья	1	Айрян	Деева	ма, ин, ф	12
96019	Богач	Раиса	1	Айрян	Доля	---	90
96022	Брунь	Елена	2	Деева	Елина	---	---
96025	Вовк	Елена	2	Деева	Ершов	---	---
96031	Гизе	Марта	2	Деева	Жук	---	87
96034	Даева	Анна	2	Деева	Жуков	---	21
96037	Деева	Алина	2	Деева	Жучко	---	---
96043	Елина	Яна	3	Рыжов	Янова	---	66
96046	Ершов	Яков	3	Рыжов	Айрян	ин, ф	64
96049	Ерина	Инна	3	Рыжов	Асеев	---	65
96052	Жук	Мария	3	Рыжов	Баева	---	14
96061	Рыжов	Павел	3	Рыжов	Белов	ма	59
96064	Янова	Цвета	3	Рыжов	Богач	ин	---

Изоморфизмы: A Z B; *A Z F*; множество {старосты этих групп} *Z* {номера этих групп}.

Эндоморфизмы: *A*] *F*; *F*] *A*; *B*] множество {старосты этих групп}.

Эпиморфизмы: *A*] *F*; *F*] *A*; *B*] множество {старосты этих групп}.

С каждым гомоморфизмом φ: *G*] *H* связана однозначно определенная *конгруэнция* системы *G*. *Конгруэнция (от латинского congruentia – согласие,* *соответствие) на универсальной алгебре A отношение эквивалентности* π на множестве элементов алгебры *A*, перестановочное с любой из операций. Последнее означает, что для любой *n*-арной операции ω, заданной в *A*, если

$a_{i\pi} a_i'$, i = 1, ... , n, $a_i, a_i' \in A$,

то и $(a_1, \ldots , a_n\omega)\pi\ (a_i', \ldots , a_n'\omega)$.

Конгруэнция π задает на множестве классов, эквивалентных по π элементов, структуру универсальной алгебры (однотипной с исходной алгеброй *A*). Эта алгебра обозначается *A*/π , и называется *факторалгеброй* алгебры *A* по конгруэнции π. С конгруэнцией π связан естественный эпиморфизм *A*] *A*/π , ставящий в соответствие каждому элементу $a \in A$ тот класс *A*/π , которому этот элемент принадлежит. Обратно, всякий гомоморфизм φ: *A*] *B* однозначно определяет конгруэнцию, классами которой служат полные прообразы элементов алгебры *B*. Аналогично определяется конгруэнция в случае алгебраической системы.

3. БИНАРНЫЕ ОПЕРАЦИИ

Бинарная операция – это, по определению, отображение множества *A* β *A* в множество *A*, при этом образ пары (*x*, *y*) обозначим, например, *x* ♥ *y*, где ♥ – символ операции. Здесь *A* – произвольное непустое множество и *A* β *A* множество всех упорядоченных пар (*x*, *y*) – таких, что $x, y \in A$ (т. е., декартов квадрат множества *A*). Непустое множество *A* называется *основным множеством* операции.

Можно составить следующую иерархию множеств с бинарной операцией (разумеется, вместо ♥ может быть вставлена любая: +, √, *, ∪, ∩, ⊕, ⊗, ∇, – в зависимости от необходимости и вкуса автора.

Группоид, обозначаемый символом (*A*, ♥) – множество *A*, на котором задана некоторая бинарная операция, обозначаемая ♥. Если множество группоида конечно, то есть к*A*к = *card* (*A*) = *n*, то таблица Кэли операции группоида есть таблица *n* β *n*, в которой элемент *x* ♥ $y \in A$ находится в клетке пересечения строки *x* и столбца *y*. Конечный группоид можно считать заданным, если выписана его таблица Кэли.

У Саши и Даши авторитет Даша.
У Саши и Маши авторитет Саша.
У Саши авторитет Саша.
У Даши и Маши авторитет Саша.
У Даши авторитет Даша.
У Маши авторитет Петя.
У Пети и Даши авторитет Петя.
У Пети и Маши авторитет Петя.
У Пети и Саши авторитет Саша.
У Пети авторитет Саша.

ТАБЛИЦА КЭЛИ ДЛЯ ОПЕРАЦИИ «АВТОРИТЕТ»

АВТОРИТЕТ	**Даша**	**Маша**	**Петя**	**Саша**
Даша	Даша	Саша	Петя	Даша
Маша	Саша	Петя	Петя	Саша
Петя	Петя	Петя	Саша	Саша
Саша	Даша	Саша	Саша	Саша

ТАБЛИЦА КЭЛИ, КОРЕЙСКИЙ ВАРИАНТ

АВТОРИТЕТ	**Ким**	**Пак**	**Чжо**
Ким	**Ким**	**Ким**	**Ким**
Пак	**Ким**	**Ким**	**Ким**
Чжо	**Ким**	**Ким**	**Ким**

4. ТЕОРИЯ ГРУПП

Квазигруппа (от латинского слова *quasi* – как-будто, почти и слова *группа*) – группоид, бинарная операция которого (например, ♥) такова, что каждое из уравнений a ♥ $x = b$, y ♥ $a = b$ имеет единственное решение для любых элементов a, b этого множества. Квазигруппа – одно из обобщений понятия группа. Особенно близки к группам квазигруппы с единицей – *лупы*, определение которых получается из аксиом групп отбрасыванием требования ассоциативности. Квазигруппу можно рассматривать и как *унивесальную алгебру* с тремя бинарными операциями (дополнительно левое и правое деление).

Гомоморфный образ квазигруппы, вообще говоря, не квазигруппа, а *группоид* с делением. Гомоморфизмам квазигруппы на квазигруппе соответствуют конгруэнции специального типа (т.н. нормальные конгруэнции). Значительно большую роль, чем гомоморфизмы, в теории и классификации групп играют изотопии. *Изотопия* – отношение эквивалентности для бинарных операций на фиксированном множестве, определяемое с помощью трех подстановок этого множества. Оказывается, что всякий группоид, изотопный квазигруппе, сам квазигруппа, а всякая квазигруппа изотопна некоторой лупе. Для групп понятие изотопии совпадает с понятием изоморфизма.

Таблица умножения конечной квазигруппы (ее *таблица Кэли*) в комбинаторике известна под названием «*латинский квадрат*». Одна из задач комбинаторной теории квазигрупп – отыскание систем взаимно ортогональных квазигрупп на заданном множестве – важна для построения конечных проективных плоскостей.

***Лупа*, или квазигруппа с единицей, определение которой получается из аксиом группы отбрасыванием требования ассоциативности, особенно близка к группе.**

***Полугруппа* – множество, с определенной на нем бинарной операцией, удовлетворяющей закону ассоциативности, т.е. группоид (**A, ♥), в котором для каждой тройки элементов a, b и c выполняется условие a♥(b♥c)=(a♥b)♥c). *Понятие полугруппы есть обобщение понятия группы: из аксиом группы остается лишь одна; этим объясняется и термин* «полугруппа».

Теория полугрупп принадлежит к числу сравнительно молодых областей алгебры. Первые исследования, посвященные полугруппам, относятся к 20-м гг. XX в. и связаны с именем А. К. Сушкевича. Он, в частности, определил строение конечной полугруппы без собственных идеалов. К концу 50-х гг. теория полугрупп сформировалась в самостоятельную ветвь современной алгебры с богатой проблематикой, разнообразными методами и тесными связями со многими областями математики – как собственно алгебраическими (в первую очередь, с теорией групп и теорией колец), так и другими, например, с функциональным анализом, дифференциальной геометрией, алгебраической теорией автоматов. Примеры полугрупп чрезвычайно многочисленны. Это, например:

– различные множества чисел вместе с операцией сложения или умножения, замкнутые относительно рассматриваемой операции (т.е. содержащие вместе с любыми двумя своими элементами их сумму или, соответственно, произведение);
– полугруппа матриц относительно умножения;
– полугруппа функций относительно «поточечного» умножения *, задаваемого формулой $(f\,g)\,(x) = f(x)\,g(x)$;
– полугруппа матриц относительно операции пересечения или объединения;
– один из простейших примеров полугруппы – множество всех натуральных чисел относительно сложения; эта полугруппа является частью (подполугруппой) группы целых чисел по сложению или, как говорят, вложима в группу целых чисел.

Далеко не всякая полугруппа вложима в какую-нибудь группу: необходимым условием такой вложимости является закон сокращения – каждое из равенств $ac = bc$, $ca = cb$ влечет за собой $a = b$; выполнение закона сокращения не достаточно для такой вложимости, но, например, коммутативная полугруппа с законом сокращения вложима в группу.

– Если на множестве F_X всех конечных последовательностей элементов произвольного множества (алфавита) X задать операцию. формулой

$$(x_1, \dots, x_m).\ (y_1, \dots, y_n) = (x_1, \dots, x_m,\ y_1, \dots, y_n),$$

то F_X станет полугруппой; она называется свободной полугруппой над алфавитом X. Роль свободных полугрупп в общей теории определяется тем, что всякая полугруппа есть гомоморфный образ подходящей свободной полугруппы. Важную роль играют свободные полугруппы и в некоторых приложениях, прежде всего в теории формальных языков и кодов.

– Всякая совокупность пребразований произвольного множества M, замкнутая относительно операции композиции (последовательного выполнения), будет полугруппой относительно этой операции; такова, в частности, совокупность всех преобразований множества M, называемая *симметрической* полугруппой на множестве M. Многие важные совокупности преобразований оказываются полугруппами относительно композиции, причем часто они не являются группами. С другой стороны, всякая полугруппа изоморфна некоторой полугруппе преобразований. Все это определяет как роль полугрупп преобразований в общей теории полугрупп, так и роль теории полугрупп для изучения в самом общем виде преобразований с точки зрения их композиции. В большой степени через рассмотрение преобразований осуществляются связи теории полугрупп с другими областями математики.

Как и в других алгебраических теориях, одной из главных задач теории полугрупп является классификация всевозможных полугрупп, описание их строения. Это осуществляется прежде всего наложением на рассматриваемые полугруппы различных ограничений и выделение тем самым различных типов полугрупп. Среди важных типов – *регулярные* полугруппы, то есть полугруппы, в которых для любого элемента a существует такой элемент x, что $axa = a$. Регулярными являются, например, полугруппа всех матриц данного порядка над телом, симметрические полугруппы, полугруппы всех частичных преобразований множеств. Регулярные полугруппы принадлежат к числу наиболее активно изучаемых в теории полугрупп.

Заметную часть общей теории составляет теория представлений полугрупп преобразованиями и матрицами. Точка зрения теории представлений нередко проливает дополнительный свет на некоторые типы полугрупп, естественно определяемые с точки зрения аксиоматики. Внесение в полугруппы дополнительных структур, согласованных с полугрупповой операцией, выделяет особые разделы теории полугрупп такие, как теория топологических полугрупп, теория упорядоченных полугрупп.

***Моноид* – это, по определению, полугруппа с единицей.**

***Группа* (от немецкого** *Grupрe* – группа) – одно из основных понятий современной математики – есть лупа, являющаяся в то же время полугруппой.

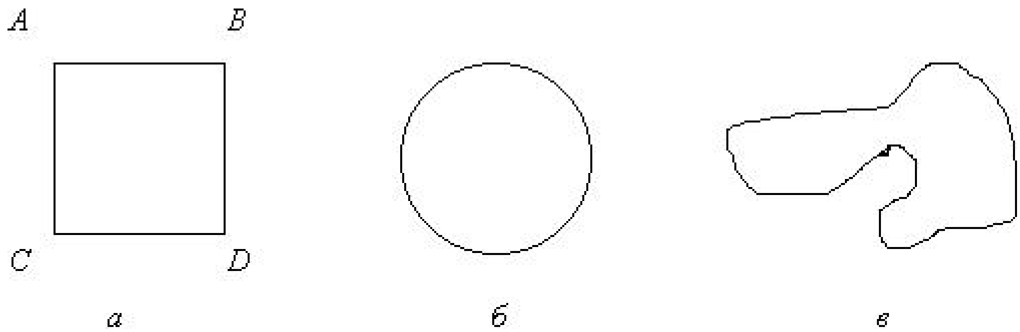
Теория групп изучает в самой общей форме операции, наиболее часто встречающиеся в математике и ее приложениях (примеры таких операций –

сложение чисел, умножение чисел, сложение векторов, последовательное выполнение преобразований и т.п.). При этом теория групп изучает не совсем произвольные операции, а лишь те, которые обладают рядом основных свойств, перечисляемых в определении группы.

Формальное определение группы таково. Пусть G – произвольное непустое множество, на котором задана *бинарная алгебраическая операция* $\circ$, т.е. для любых двух элементов a, b, из G определен некоторый элемент (обозначаемый, например, $a\circ b$) также из G. Если при этом выполняются условия: 1) $(a\circ b)\circ c = a\circ(b\circ c)$ для любых a, b и c из G; 2) в G существует такой элемент e (называемый *единицей*, иногда - *нейтральным элементом*), что $a\circ e = e\circ a = a$ для любого a из G; 3) для любого a из G существует такой элемент a^{-1} (обратный к a элемент), что $a\circ a^{-1} = a^{-1}\circ a = e$, то множество G с заданной на нем операцией $\circ$ назовем *группой*.

Примеры групп.

1) *Множество G* различных движений эвклидовой плоскости, самосовмещающих данную фигуру, операцией на котором служит композиция движений (если φ, ψ – два движения из G, то результатом их композиции назовем движение $\varphi\circ\psi$, равносильное последовательному выполнению сначала движения φ, а затем движения ψ), образует так называемую *группу симметрий фигуры*. Единицей в этой группе будет тождественное преобразование плоскости, а обратным к φ элементом ≈ обратное к φ преобразование. Группа G является характеристикой большей или меньшей симметричности фигуры: чем больше множество G, тем симметричнее фигура. Например, группа симметрий квадрата (рис.1, *а*) состоит из восьми движений (четыре поворота вокруг центра квадрата и четыре отражения: два – относительно диагоналей и два – относительно прямых, соединяющих середины противоположных сторон). Для круга (рис.1,*б*) группа симметрий уже содержит бесконечно много элементов (например, все повороты вокруг центра), а для фигуры, нарисованной на рис. 1, *в* группа симмерий состоит из одного тождественного преобразования.

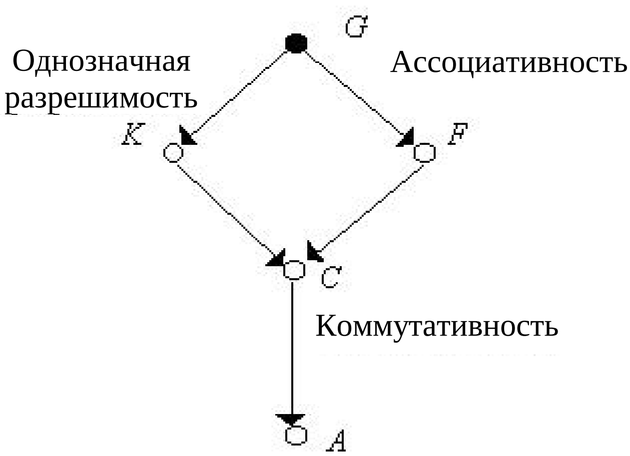


Рисунок

Если Z – множество всех целых чисел, а операция на Z – их обычное сложение, то Z – группа. Роль e будет играть число 0, а роль обратного к z элемента – число z^{-1}. Часть H множества Z, состоящая из четных чисел, сама будет группой относительно той же операции. В таком случае говорят, что H – *подгруппа группы* Z. Обе групы Z и H удовлетворяют следующему дополнительному условию: 4) $a + b = b + a$ для любых a и b из группы. Всякая группа, в которой выполняется условие 4) называется *коммутативной* или *абелевой*.

3) Множество всех *подстановок* n символов образует группу относительно умножения подстановок, называемую *симметрической группой*. При $\boldsymbol{n} = 3$ симметрическая группа неабелева. Порядок (число элементов) симметрической группы равен $\boldsymbol{n!}$.

Очередность образования вышеперечисленных понятий наглядно представлена в виде направленного графа. Буквы, приписанные точкам графа, означают: G – группоид, K – квазигруппа, F – полугруппа, C – группа, A – абелева группа.



Рисунок

Здесь, разумеется, можно нарисовать (попробуйте сами) два более точных графа, используя понятия *лупы* и *моноида*.

Абелева группа есть группа **(**A, (χ)), в которой для любых двух элементов $a, b \in A$ имеет место $a\,(\chi)\ b = b(\chi)\ a$. При одинаковом порядке элементов в заглавной строке и заглавном столбце таблица операции абелевой группы симметрична относительно главной диагонали.

Некоторые теоретико-групповые понятия и теоремы, играющие особо важную роль в конечных геометриях

Циклическая группа**.** Группа называется циклической, если она содержит элемент, множество степеней которого исчерпывает всю группу.

Если группа **G** содержит две такие подгруппы **A** и **B**, что каждый элемент $g \in$ **G** можно единственным образом представить в виде $g = a\ b$, где $a \in$ **A** и $b \in$ **B,** и для любых элементов $a \in$ **A** и $b \in$ **B** имеет место $a\ b = b\ a$, то группу **G** называют прямым произведением подгрупп **A** и **B** и пишут: **G**= **A** β **B**. Аналогично определяется прямое произведение нескольких подгрупп группы **G**.

Теорема Лагранжа. *Если* **A** *– подгруппа конечной группы* **G**, *то* |**G**| *делится нацело на* |**A**|.

Следствием этой теоремы является тот факт, что порядок любого элемента конечной группы служит делителем порядка группы. Другое следствие состоит в том, что группа простого порядка является циклической.

Примарной группой (p-группой) называется группа, порядки всех элементов которой являются степенями некоторого заданного простого числа p.

Элементарной группой называется группа, в которой порядок каждого элемента, отличного от единичного, есть некоторое заданное простое число. Очевидно, всякая элементарная группа является p-группой.

Нормальный делитель. Подгруппа **N** группы **G** называется нормальным делителем группы **G**, если для каждого элемента $x \in$ **G** множество **N** $x = \{nx \mid n \in$ **N**$\}$ совпадает с множеством x **N** $= \{xn \mid n \in$ **N**$\}$. Это записывается в виде **N > G**.

***Инвариантный элемент*. Элемент** c группы **G** называется инвариантным, если он перестановочен с каждым элементом группы **G**, т.е. если $c\ x = x\ c$, для всякого $x \in$ **G**.

***Центр группы*. Инвариантные элементы группы G** образуют ее нормальный делитель Z(**G**), называемый центром этой группы.

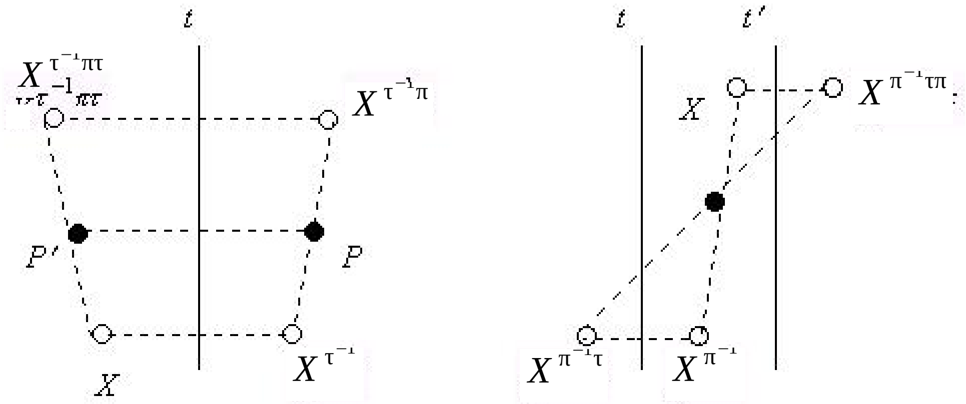
Теорема. *Если порядок группы является степенью простого числа, то ее центр содержит элементы, отличные от единичного элемента.*

Сопряженные элементы группы. Два элемента a и b группы **G** называются сопряженными, если существует такой элемент g, что $a\ g = g\ b$, т.е., если $g^{-1}\ a\ g = b$. Говорят также, что a получается ***трансформацией*** элемента b элементом g. Если $g^{-1}a\ g = a$, то элемент a называется ***инвариантным*** относительно трансформации элементом g.

Сопряженная подгруппа. Если для подгрупп **A** и **B** группы **G** найдется такой элемент $g \in$ **G**, что **A** $g = g$ **B**, т.е. g^{-1}**A** $g =$ **B**, то подгруппа **B** называется сопряженной подгруппе **A** в **G**. Говорят также, что подгруппа **B** получается ***трансформацией*** подгруппы **A** элементом g. Если g^{-1}**A** $g =$ **A**, то подгруппа **A** называется ***инвариантной*** относительно трансформации элементом g. Поскольку нормальный делитель группы инвариантен относительно трансформации любым ее элементом, то его называют также ***инвариантной подгруппой*** группы.

Покажем теперь на примере, как понятия теории групп позволяют сжато выражать некоторые геометрические факты. Чертеж слева (где P –

фиксированная точка, t – фиксированная прямая, X – переменная точка, пробегающая множество всех точек плоскости) приводит нас к следующей ***теореме***. Результат последовательного отражения точки X относительно прямой t, полученного образа относительно точки P и этого нового образа опять относительно прямой t совпадает с результатом отражения точки X относительно точки PP, симметричной точке P относительно прямой t. Сформулируем этот факт в терминах группы движений плоскости. Отражение относительно точки P обозначим через π. Очевидно, $\pi^2 = 1$, т.е. $\pi = \pi^{-1}$. Аналогично, обозначив отражение относительно прямой t через τ, получаем, что $\tau^{2} = 1$, откуда $\tau^{-1} = \tau$. Теперь наше геометрическое утверждение о трех последовательных отражениях может быть выражено следующей формулой: $\pi P = \tau^{-1\pi}\tau$. Чертеж справа показывает, что три последовательных отражения – относительно точки P, затем прямой t и, наконец, снова точки P – равносильны отражению относительно прямой tP, симметричной прямой t относительно точки P. Это можно выразить формулой: $\tau P = \pi^{-1\tau\pi}$.



Кроме того, легко проверить, что условие принадлежности точки P прямой t равносильно условию $\pi\tau = \tau\pi$. Чтобы убедиться в этом, достаточно заметить, что в первом случае условие $\tau^{-1\pi}\tau = \pi$ означает совпадение точек P и PP, тогда как во втором случае условие $\pi^{-1\tau\pi} = \tau$ означает совпадение прямых t и tP.

Преимущества языка теории групп становятся особенно убедительными, если выраженные на этом языке геометрические понятия и отношения попытаться выразить также средствами аналитической геометрии. Например:

$$\left.\begin{matrix} P:(x,y) \\ t:[m,b] \end{matrix}\right\} P\in t: y = mx+b, \quad \left.\begin{matrix} P:\pi \\ t:\tau \end{matrix}\right\} P\in t: \pi\tau=\tau\pi$$

Теорема Силова. *Если порядок конечной группы делится на степень p^r простого числа p, то эта группа имеет подгруппу порядка p^r.*

Силовская подгруппа. Если к Gк = $p^k s$ и s не делится на p, то подгруппы порядка p^k группы G называются силовскими p-подгруппами.

Следствие 1. Если порядок конечной группы делится на простое число *p*, то в группе имеется элемент порядка *p*.

***Следствие 2*. Порядок любой конечной** *p*-группы является степенью числа *p*.

Основная теорема об абелевых группах. *Любая конечная абелева группа разлагается в прямое произведение циклических групп, порядки которых являются степенями простых чисел; множество степеней простых чисел, являющихся порядками указанных групп, однозначно определяется порядком самой группы.*

Это надо понимать следующим образом: если $|G| = p_1^{y_1} p_2^{y_2} \mathrm{K} p_k^{y_s}$ *, то можно указать такие циклические группы* A_1, A_2, . . ., A_k порядков $p_1^{y_1}, p_2^{y_2}, \mathrm{K}, p_k^{y_s}$, соответственно, что $\mathrm{G} = \mathrm{A}_1 \times \mathrm{A}_2 \times \mathrm{K} \times \mathrm{A}_k$. Отсюда следует, что конечная абелева группа однозначно определяется с точностью до изоморфизма конечным набором степеней простых чисел.

Пусть даны множество $H = \{1, 2, \ldots, n\}$ и группа подстановок Γ этого множества. ***Стабилизатором*** элемента $x \in H$ называется следующая подгруппа группы Γ:

$$\Gamma_x = \{\pi \in \Gamma \text{ к} x\pi = x\}.$$

***Орбитой* элемента** *x* относительно группы Γ называется множество $x\Gamma = \{x\,\pi \text{ к } \pi \in \Gamma\}$. В обоих случаях через $x\pi$ обозначается элемент, в который подстановка π переводит *x*.

Группа подстановок Γ множества *H* называется ***регулярной***, если для каждого $x \in H$ имеет место $\Gamma_x = 1$.

Конечной целью собственно теории групп является описание всех возможных групповых операций. Теория групп распадается на ряд больших разделов, выделяемых чаще всего дополнительными условиями на групповую операцию или внесением в группу дополнительных структур, связанных определенным образом с групповой операцией. Перечислим важнейшие разделы теории групп.

а) *Теория конечных групп*. Основная проблема этой старейшей ветви теории групп – классификация так называемых простых конечных групп, играющих роль «строительных блоков» при построении произвольной конечной группы.

б) *Теория абелевых групп*. Отправной точкой многих исследований в этой области служит основная теорема о конечно порожденных абелевых группах, полностью выясняющая их строение.

в) *Теория разрешимых и нильпотентных групп*. Понятие разрешимой группы является обобщением понятия абелевой группы. Оно, по существу, идет от Э. Галуа и тесно связано с разрешимостью уравнений в радикалах.

г) *Теория групп преобразований*. Понятие группы возникло исторически именно как понятие группы преобразований, но в дальнейшем было освобождено от этой конкретной оболочки. Тем не менее теория групп преобразований осталась важной частью общей теории. Типичный вопрос в ней: какими абстрактными свойствами обладает группа, заданная как группа преобразований некоторого множества? Особое внимание привлекают, в частности, группа подстановок и группа.

д) *Теория представлений групп* – важное орудие изучения абстрактных групп. Представление абстрактно группы в виде некоторой конкретной группы (например, в виде группы подстановок или матриц) позволяет проводить тонкие вычисления и с их помощью обнаруживать важные абстрактные свойства. Особенно велики успехи теории представлений в теории конечных групп, где с ее помощью получен ряд результатов, недоступных пока абстрактным методам.

е) Из разделов теории групп, выделяемых внесением в группы дополнительных структур, согласованных с групповой операцией, отметим теорию *топологических групп* (в них групповая операция в некотором смысле непрерывна), в частности ее старейшую ветвь – теорию *групп Ли*.

Теория групп является одной из самых развитых областей алгебры и имеет многочисленные применения как в самой математике, так и за ее пределами. Например, с помощью теории групп Е. С. Федоров (1890) решил задачу классификации правильных пространственных систем точек, являющуюся одной из основных задач кристаллографии. Это был исторически первый случай применения теории групп непосредственно в естествознании. Большую роль играет теория групп в физике, например, в квантовой механике, где широко используются соображения симметрии и теория представлений групп линейными преобразованиями.

Историческая справка. Понятие группы послужило во многих отношениях образцом при перестройке математики на рубеже XIX-XX вв. Истоки понятия группы обнаруживаются в нескольких дисциплинах, главная из которых – теория решений алгебраических уравнений в радикалах. В 1771 г. Ж. Лагранж и А. Вандермонд впервые для нужд этой теории применили подстановки. Затем в ряде работ П. Руффини (1799 и позднее), посвященных доказательству неразрешимости уравнения 5-й степени в радикалах, систематически используется замкнутость множества подстановок относительно их умножения и, по существу, описаны подгруппы всех подстановок пяти символов. Глубокие связи между свойствами групп подстановок и свойствами уравнений были установлены Н. Абелем (1824) и Э. Галуа (1830). Э. Галуа принадлежат и многие достижения собственно в теории групп: открытие роли так называемых нормальных подгрупп в связи с задачей о разрешимости уравнений в радикалах, установление свойства простоты знакопеременных групп степени n ♦ 5 и др.; он же ввел термин «группа» (*le groupe*), хотя и не дал строгого определения. Важную роль в систематизации и развитии теории групп сыграл трактат К. Жордана (1870) о группах подстановок.

Независимо и из других соображений идея групп возникла в геометрии, когда в середине XIX в. На смену единой античной геометрии пришли многочисленные «геометрии» и остро встал вопрос об установлении связей и родства между ними. Выход из создавшегося положения был намечен исследованиями по проективной геометрии, посвященными изучению поведения фигур при различных преобразованиях. Постепенно интерес в этих исследованиях перешел на изучение самих преобразований и поиск их классификации. Таким «изучением геометрического родства» много занимался А. Мебиус. На более сознательном уровне классификацию геометрий дал А. Кэли (1854 и далее), он явно пользовался термином «группа», систематически использовал таблицы умножения, ныне называемые *таблицами Кэли, доказал представимость всякой конечной группы подстановками. Заключительным этапом на этом пути явилась* «Эрлангенская программа» Ф. Клейна (1872), положившая в основу классификации геометрий понятие группы преобразований: каждая геометрия определяется некоторой группой преобразований пространства, и только те свойства фигур принадлежат к данной геометрии, которые инвариантны относительно преобразований соответствующей группы. (См. также *Классическая группа*).

Третий источник понятия группы – теория чисел. Уже Л. Эйлер (1761), изучая «вычеты, остающиеся при делении степеней», по существу пользовался сравнениями и разбиениями на классы вычетов, что на теоретико-групповом языке означает разложение группы на смежные классы по подгруппе. К. Гаусс в «Арифметических исследованиях» (1801), занимаясь уравнением деления круга, фактически определил подгруппы его группы Галуа. Там же, изучая «композицию двоичных квадратичных форм», К. Гаусс, по существу, доказывает, что классы эквивалентных форм образуют относительно композиции конечную абелеву группу.

Осознание в конце XIX века принципиального единства теоретико-групповых идей, существовавших к тому времени независимо в разных областях математики, привело к выработке современного абстрактного понятия групп (С. Ли, Г. Фробениус и др.). Так, уже в 1895 г. С. Ли определял группу как совокупность преобразований, замкнутую относительно их композиции, удовлетворяющей условиям 1), 2), 3). Изучение групп без предположения их конечности и без каких бы то ни было предположений о природе элементов впервые оформилось в самостоятельную область математики с выходом книги О. Ю. Шмидта «Абстрактная теория групп» (1916).

5. КОЛЬЦА, ТЕЛА, ПОЛЯ

Кольцо – одно из основных понятий современной алгебры. В различных областях математики часто приходится иметь дело с разнообразными множествами, над элементами которых можно производить две операции, весьма похожие по своим свойствам на сложение и умножение обычных чисел. Предметом теории колец является изучение свойств обширного класса такого рода множеств.

Кольцом **называется непустое множество** R, для элементов которого определены две бинарные операции – сложение и умножение (обозначаемые $+$ и $\cdot$ соответственно; знак $\cdot$ обычно опускается), причем предполагаются выполненными следующие аксиомы колец ($a, b, c \in R$):

I. Коммутативность сложения: $a + b = b + a$.

II. Ассоциативость сложения: $a + (b + c) = (a + b) + c$.

III. Обратимость сложения (возможность вычитания): уравнение $a + x = b$ имеет решение $x = b - a \in R$.

IV. Дистрибутивность умножения относительно сложения: $a\,(b + c) = a\,b + a\,c$ и $(b + c)\,a = b\,a + c\,a$.

Кольцо является обобщением системы (Z, +, $\cdot$) целых чисел, однако в кольце условие $ab= ba$ может нарушаться и равенство $ab = 0$ может иметь место и тогда, когда

$$a\ \gamma\ 0 \text{ и } b\ \gamma\ 0.$$

Из свойств I – III ясно, что элементы кольца образуют абелеву группу относительно сложения: она называется ***аддитивной группой кольца***. Нуль 0 этой группы относительно умножения является «поглощающим» элементом, т.е. $a \cdot 0 = 0 \cdot a = 0$ для любого элемента a кольца. Кольцо, вообще говоря, может содержать и так называемые *делители нуля*, т.е. такие ненулевые элементы a и b, произведение которых равно 0. Единицей называется такой элемент 1 кольца R, что $a \cdot 1 = 1\cdot a = a$ для всех $a\in R$. Кольцо не обязано обладать единицей, но если она есть, то только одна.

Примеры колец:

1) множество всех целых чисел;
2) множество всех четных чисел и вообще целых чисел, кратных данному числу m;
3) множество всех рациональных чисел;
4) множество всех действительных чисел;
5) множество всех комплексных чисел;
6) множество всех гауссовых чисел, т.е. комплексных чисел $a + bi$ с целыми a и b;

7) множество всех многочленов от одного или нескольких переменных с рациональными, действительными или комплексными коэффициентами;
8) множество всех функций, непрерывных на данном отрезке чмсловой прямой;
9) множество всех квадратных матриц порядка *n* с действительными (или комплексными) элементами;
10) множество всех кватернионов;
11) множество всех чисел Кэли;
12) множество всех симметричных матриц порядка *n* с действительными элементами относительно сложения матриц и йорданова умножения a o b = к(ab + ba), где в правой части стоят обычные произведения матриц;
13) множество всех векторов трехмерного пространства относительно обычного сложения и векторного умножения.

Во многих случаях на умножение в кольцах накладывают дополнительные ограничения. Так, если a(bc) = (ab)c, то кольцо называют *ассоциативным* (примеры 1 – 10); если в кольце выполняются равенства (aa)b = a(ab), (ab)b = a(bb), то оно называется *альтернативным* (например, 11); если в кольце выполняются равенства ab = ba и (ab)(aa)=((aa)b)a, то оно называется *йордановым кольцом* (например, 12); если в кольце выполняются равенства $a^2=0$, a(bc) + b(ca) + c(ab) = 0, то оно называется *кольцом Ли* (пример 13); если ab=ba, то кольцо называется *коммутативным* (примеры 1 – 8, 12). Ассоциативно-коммутативное кольцо с единицей и без делителей нуля называется *областью целостности* (примеры 1 – 7). Ассоциативное кольцо, в котором при a <> 0 разрешимы оба уравнения ax = b и xa = b, называется *телом* (примеры 3 – 5, 10); тело обладает единицей и не имеет делителей нуля. Коммутативное тело называется *полем*.

Тело есть такая система (A, + , ·), что система (A, +) является абелевой группой, а система (AP, ·), где AP получается из A удалением нулевого элемента (т.е. нейтрального элемента абелевой группы), является группой и операция · дистрибутивна относительно операции + . Если система (AP, ·) также является абелевой группой, то тело называют коммутативным (или полем).

Теорема. *Кольцо является телом тогда и только тогда, когда оно содержит не менее двух элементов и оба уравнения ax = b* и *xa = b, разрешимы для любых элементов a, b∈* A*, где a γ* 0.

Тело служит обобщением системы (Q, + , ·) рациональных чисел, однако требование коммутативности умножения опускается.

Теорема Веддербёрна. *Всякое конечное тело коммутативно*.

Квазитело есть такая система (А, + , ·), что система (А, +) является абелевой группой, а система (А, ·) – квазигруппой с единичным элементом (т.е. лупой), причем имеет место левая дистрибутивность, т.е. для любых трех элементов $a, b, c \in$ А выполняется равенство $a\ (b + c) = (a\ b) + (a\ c)$. Если имеет место также и правая дистрибутивность, то квазитело называется *дистрибутивным.*

Если операция – ассоциативна, то квазитело называется *ассоциативным*.

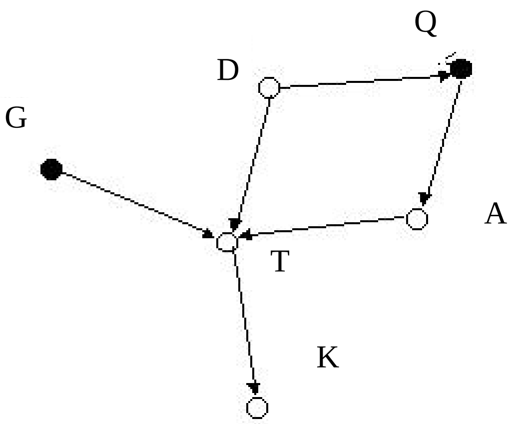
Альтернативное квазитело – это дистрибутивное квазитело, для которого выполняются следующие условия:

$a\ \ (aP\ x) = x$, если $a\,\gamma\ \ 0$ и $aaP\ \ = 1$;

$(x\ aP\ )aPP = x$, если $aPP\gamma\ \ 0$ и $aP\ aPP = 1$.

Теорема. *Дистрибутивное и ассоциативное квазитело является телом.*

Иерархия систем с двумя бинарными операциями изображена на рисунке. Обозначения: G – кольцо, Q – квазиполе, D – дистрибутивное квазиполе, A – ассоциативное квазиполе, T – тело, K – коммутативное тело.



6. ТЕОРИЯ АЛГЕБР

Весьма важны для многих разделов алгебры кольца многочленов над произвольным полем и кольца матриц над телами, определяемые аналогично кольцам примеров 7 и 9. Многие классы колец находят приложения и вне алгебры. Важнейшими из них являются кольца функций и кольца операторов, сыгравшие большую роль в развитии функционального анализа.

Пусть Ф – произвольное ассоциативное кольцо с единицей. Кольцо А (не обязательно ассоциативное) называется *алгеброй над* Ф или *операторным кольцом* с кольцом операторов Ф, если определено произведение любого элемента из Ф на элемент из А, лежащее в А, причем так, что для всех $\alpha\ \beta\ \ \in$ Ф, $a, b\in$A справедливы соотношения

(? + ?)a = ?a + ?a, ?(a + b) = ?a + ?b,

?(?a) = (??)a, 1a = a, ?(ab) = (?a)b.

А если кольцо Ф коммутативно, то принято требовать усиления последнего условия:

? (ab) = (?a)b = a(?b)

Любое кольцо можно считать алгеброй над кольцом целых чисел, если понимать произведение na (где n – целое число), как сумму n экземпляров элемента a: $a + a + \ldots + a$.

Если A – алгебра над полем (называемая также *линейной алгеброй*), то, по определению, А является векторным пространством над этим полем, а значит имеет базис. Это дает возможность строить алгебры над полем по базису, для чего достаточно задать таблицу умножения базисных элементов. Алгебра над полем называется *конечномерной*, если она конечномерна как векторное пространство. Размерность этого векторного пространства называется над *рангом* алгебры. Например, поле C комплексных чисел есть алгебра ранга 2 над полем *R* действительных чисел, кватернионы образуют алгебру ранга 4 над полем *R*, полное кольцо матриц порядка *n* с элементами из поля Ф – алгебра ранга n^2 над Ф.

В теории колец важную роль играют понятия гомоморфизма и изоморфизма. Многие рассуждения и описания проводятся «с точностью до изоморфизма», т.е. изоморфные кольца и алгебры не различаются. *Гомоморфизм* – это такое отображение φ кольца А в кольцо А', что для любых $a, b \in A$

$$(a + b)? = a? + b?, (ab)? = (a?)(b?),$$

т. е. φ перестановочно с кольцевыми операциями. Для алгебр (над одним и тем же Ф) требуют также, чтобы было (?*a*)? = ?(*a*?) ??я любого $\alpha \in$ Ф. Если при этом φ – биективное отображение (т.е. взаимно однозначное отображение на А'), то оно называется *изоморфизмом*, а кольца (алгебры) *А* и А' изоморфными.

Множество *М* элементов кольца (алгебры) *А* называется *подкольцом* (*подалгеброй), если М само является кольцом (алгеброй) относительно операций, определенных в А; М* называется *левым* (*правым или двусторонним*) ***идеалом*, если, кроме того, для любых элементов** $m \in M$ и $a \in A$ произведение *am* (соответственно, *ma* или как *am*, так и *ma*) лежит в *М*. Элементы $a, b \in A$ называются сравнимыми по идеалу *М*, если $b - a \in M$. Всё *А* разбивается на классы сравнимых элементов – классы вычетов по идеалу. Таким образом, всякий идеал определяет на множестве *А* отношение эквивалентности, причем для двустороннего идеала это отношение – конгруэнция, и можно определить сложение и умножение (умножение на элемент поля) классов вычетов по двустороннему идеалу *М* через сложение и умножение элементов этих классов. Относительно этих операций классы вычетов образуют кольцо (алгебру), называемое *факторкольцом* (*фактоалгеброй) А/М. Имеет место* ***теорема о гомоморфизмах****: если каждому элементу из А поставить в соответствие содержащий его класс, то получается гомоморфизм А на А/М; обратно, если А гомоморфно отображается на А'*, то множество *М* элементов из *А*, отображающихся в нуль кольца (алгебры) А', будет двусторонним *идеалом* в *А* и *А*/*М* изоморфно А'. Кольцо без двусторонних идеалов называется *простым*.

Переход от алгебры к ее подалгебрам и факторалгебрам является одним из способов получения новых алгебр. Например, из алгебры многочленов от достаточно большого числа переменных над полем Ф (в качестве гомоморфного образа) может быть получена любая ассоциативно-коммутативная алгебра над полем Ф.

Поле – алгебраическое понятие, широко используемое во многих разделах математики. Поля составляют особый подкласс колец.

Поле может быть определено как множество, содержащее не менее двух элементов, на котором заданы две бинарные алгебраические операции – сложение и умножение, обе ассоциативные и коммутативные, связанные между собой законом дистрибутивности, т.е. для любых a, b, c из поля справедливо:

$a + b = b + a$, $ab = ba$,

$(a + b) + c = a + (b + c)$, $(ab)\,c= a\,(bc)$,

$(a + b)\,c = ac + bc$.

Кроме того, в поле требуется существование нулевого элемента 0 (*нуля*), для которого $0 + a = a$, и для каждого элемента a противоположного элемента –a, то есть такого элемента, что $a + (–a) = 0$, а также существование единичного элемента e (*единицы), для которого $ae = a$*, и для каждого ненулевого элемента a существование *обратного элемента* a^{-1}, т.е. такого элемента, что $aa^{-1} = e$. Отсюда следует, что в поле выполнимы операция вычитания, а также операция деления на ненулевой элемент. Таким образом, все элементы поля образуют – абелеву группу по сложению (*аддитивная группа поля*), а все ненулевые элементы – абелеву группу по умножению (*мультипликативная группа поля*).

Примерами полей (относительно естественных операций сложения и умножения) являются: множество всех рациональных чисел Q, множество всех действительных чисел R, множество всех комплексных чисел C, множество всех чисел вида $a + b$?2, где a и b – рациональные числа, множество всех алгебраических чисел, множество всех рациональных функций от одного или нескольких переменных с действительными коэффициентами (а также с коэффициентами из произвольного поля). Множество элементов поля может быть конечным. Такие поля называют ***полями Галуа***. Простейшие примеры конечных полей – поля вычетов кольца по простому модулю.

Может оказаться, что в поле равно нулю целое кратное na какого-либо отличного от нуля элемента a. В этом случае существует такое простое число p, что p-кратное любого элемента этого поля равно нулю. Говорят, что в этом случае *характеристика* поля равна p (таковы поля вычетов по модулю p). Если же naγ 0 ни для каких ненулевых n и a, то характеристика поля считается равной нулю. В поле не может быть делителей нуля. Всякое поле является *простым кольцом*, т.е. не содержит ненулевых идеалов, не совпадающих с ним. Обратно, всякое ненулевое ассоциативно-коммутативное простое кольцо с единицей есть поле. Поле называется *простым*, если оно не содержит *подполей*, т.е. таких подмножеств, которые сами являются полями относительно тех же операций сложения и умножения. Если F – подполе в G, то G называют также *надполем* или *расширением поля* F. Всякое поле содержит единственное простое подполе. Все простые поля характеристики 0 исчерпываются полем рациональных чисел, а простые поля характеристики p – полем вычетов по модулю p.

Основные задачи теории поля - это описание всех подполей данного поля, всех расширений данного поля, классификация полей с точностью до изоморфизма и изучение групп автоморфизмов полей.

Один из способов обозрения всех полей – отправляясь от простого поля – получать описания всех полей, изучив структуру расширений. Расширение, которое порождается присоединением к F одного элемента, называется простым. Возможны простые расширения двух типов: а) *простое трансцендентное расширение*, которое получается, если за G взять поле всех рациональных функций одного переменного с коэффициентами из F, и б) *простое алгебраическое расширение*, которое получается, если к F добавить корень некоторого неприводимого над F многочлена $f(x)$ и все те элементы, которые можно выразить через этот корень и элементы из F. Во втором случае каждый элемент получаемого надполя G является корнем некоторого многочлена с коэффициентами из F. Расширения, обладающие этим свойством, называются *алгебраическими*, а все остальные – *трансцендентными*. Важный класс алгебраических расширений составляют *конечные расширения*, т.е. расширения, являющиеся конечномерными векторными пространствами над полем F. Любое расширение можно выполнить в два приема: сначала совершить *чисто трансцендентное расширение* (образовав поле рациональных функций, необязательно от одной переменной), а затем алгебраическое. Алгебраических расширений не имеют такие поля, в которых каждый многочлен разлагается на линейные множители (они называются *алгебраически замкнутыми*). Таково, например, поле C комплексных чисел (*основная теорема алгебры*). Любое поле содержится в некотором алгебраически замкнутом поле.

В теории чисел важную роль играет изучение конечных расширений полей Q, так называемых *полей алгебраических чисел*. Теория поля изучает также поля, несущие некоторые дополнительные структуры, например, нормированные поля, топологические поля, упорядоченные поля.

Решетка, структура, – частично упорядоченное множество, в котором каждое двухэлементное подмножество имеет как точную верхнюю (*sup*), так и точную нижнюю (*inf*) грани. Отсюда вытекает существование этих граней для любых непустых конечных подмножеств.

Примеры решеток:

1) множество всех подмножеств данного множества, упорядоченное по включению;
2) всякое линейно упорядоченное множество; причем если a ? b, то $sup\{a,b\} = b$, $inf\{a,b\} = a$;

3) множество всех надпространств векторного пространства, упорядоченных по включению, где *inf* – пересечение, а *sup* – сумма соответствующих надпространств;
4) множество всех неотрицательных целых чисел, упорядоченных по делимости: a ? b, если $b = ac$ для некоторого c. Здесь *sup* – наименьшее общее кратное, а *inf* – наибольший общий делитель данных чисел;
5) действительные функции, определенные на отрезке [0, 1], упорядоченные условием $f \div g$, если $f(t) \div g(t)$ для всех $t \in [0, 1]$. Здесь

$$sup\{f,g\} = u, \text{ где } u(t) = max\{f(t), g(t)\}.$$

Решетка может быть также определена как универсальная алгебра с двумя бинарными операциями (они обозначаются + и $\cdot$ или $\cup$ и $\cap$, а также $\vee$ и $\wedge$), удовлетворяющая следующим тождествам

(1) $a + a = a$, (1Р) $a \cdot a = a$ {*идемпотемпотентность*};

(2) $a + b = b + a$, (2Р) $a \cdot b = b \cdot a$ {*коммутативность*};

(3) $(a + b) = c = a + (b + c)$, (3Р) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ {*ассоциативность*};

(4) $a\ (a + b) = a$, (4Р) $a + a \cdot b = a$ {*поглощение*}.

Связь между этими двумя определениями устанавливается при помощи формул:

$a + b = sup\ \{a, b\}$, $\qquad a \cdot b = inf\ \{a, b\}$,

и обратно. При этом для любых элементов a и b эквивалентны следующие утверждения: (а) a ? b; (б) $a\ b = a$; (в) $a\ \ b = b$. Понятия изоморфизма решеток как универсальных алгебр и как частично упорядоченных множеств совпадают. Однако произвольное изотонное отображение решетки R в решетку RP не обязано быть гомоморфизмом этих решеток как универсальных алгебр.

Появление понятия «решетка» относится к середине XIX в. Четко его сформулировал Р. Дедекинд в работах 1894 и 1897 г.г. Термин «*lattice*», переведенный как «структура», был введен Г. Биркгофом в 1933 г. В настоящее время в русской терминологии (из-за многозначности слова «структура») он вытеснен переводом «решетка». Исторически роль теории решеток объясняется тем, что многие факты, касающиеся множества идеалов кольца и множества нормальных подгрупп группы, выглядят аналогично и могут быть доказаны в рамках теории ***дедекиндовых решеток***. Как самостоятельное направление алгебры эта теория сформировалась в 30-х годах XX в. Наиболее важные классы решеток, кроме дедекиндовых, - *это полные решетки, дистрибутивные решетки и булевы алгебры.*

7.ТОЖДЕСТВА, БИНАРНЫЕ ОПЕРАЦИИ

Бинарное отношение R в множестве M, обладающее следующими свойствами:

– *рефлексивности*: $(\forall a\in M)\ ((a,a)\in R)$;
– *антисимметричности*: $(\forall a,b\in M)\ (((a,b)\in R)$ и $(b,a)\in R) Z\ a=b)$;
– *транзитивности*: $(\forall a,b,c\in M)\ (((a,b)\in R)$ и $(b,c)\in R)(\rho)\ (a,c)\in R)$

называется ***отношением упорядоченности*** и может быть обозначено: $\div$.
Бинарное отношение R в множестве M, обладающее следующими свойствами:

– *антирефлексивности*: $(\forall a\in M)\ ((a,a)\notin R)$;
– *антисимметричности*: $(\forall a,b\in M)\ (((a,b)\in R)$ и $(b,a)\in R) Z\ a=b)$;
– *транзитивности*: $(\forall a,b,c\in M)\ (((a,b)\in R)$ и $(b,c)\in R)(\rho)\ (a,c)\in R)$

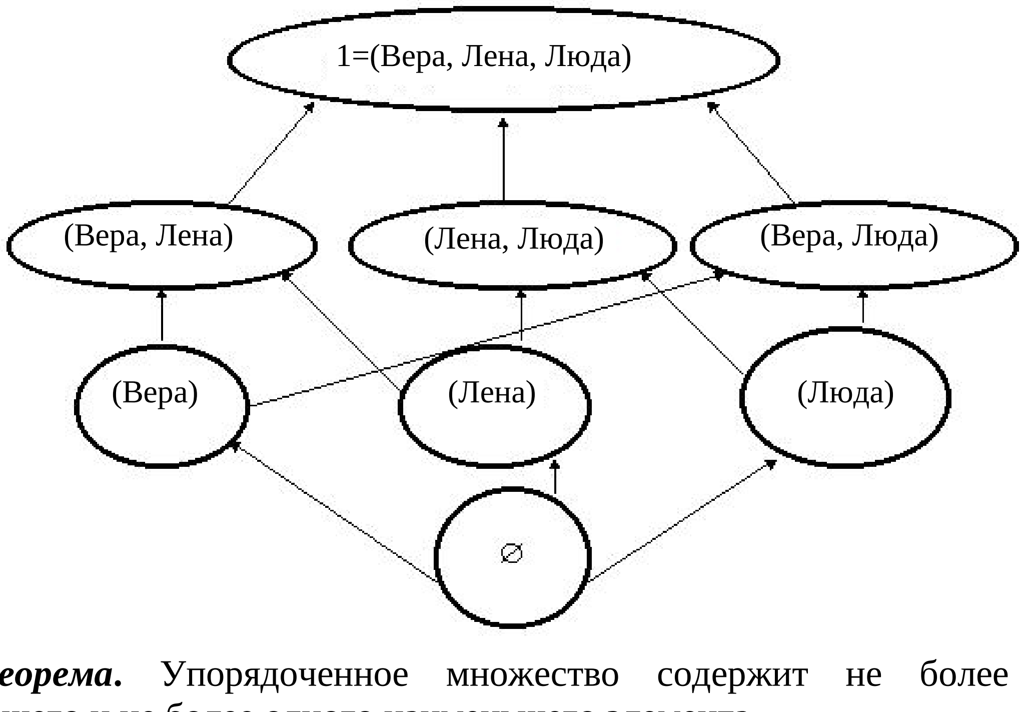
называется ***отношением строгой упорядоченности*** и может быть обозначено: <.

Бинарное отношение R в множестве M, обладающее следующими свойствами:

– *рефлексивности*: $(\forall a\in M)\ ((a,a)\in R)$;
– *транзитивности*: $(\forall a,b,c\in M)\ (((a,b)\in R)$ и $(b,c)\in R)(\rho)\ (a,c)\in R)$

называется ***отношением предпорядка***.

Диаграмма Хассе



Теорема. Упорядоченное множество содержит не более одного наибольшего и не более одного наименьшего элемента.

Теорема Стоуна. Булева алгебра изоморфна алгебре Кантора.

$A\ (\rho)\ x\in A\cup(\rho)$

$\cap(\rho)\wedge\lambda(\rho)\neg$

ТОЖДЕСТВА (ЗАКОНЫ), ОПРЕДЕЛЯЮЩИЕ ОПЕРАЦИИ ДИЗЪЮНКЦИИ, КОНЪЮНКЦИИ И ОТРИЦАНИЯ

- идемпотентностей $a \vee a = a$, $a \wedge a = a$;
- коммутативностей $a \vee b = b \vee a$, $a \wedge b = b \wedge a$;
- ассоциативностей $a \vee (b \vee c) = (a \vee b) \vee c$, $a \wedge (b \wedge c) = (a \wedge b) \wedge c$;
- дистрибутивностей $a \wedge (b \wedge c) = a \wedge b \vee a \wedge c$; $a \vee b \wedge c = (a \vee b) \wedge (a \vee c)$;
- двойного отрицания $\neg\neg a = a$;
- де-Моргана $\neg a \vee \neg b = \neg(a \wedge b)$, $\neg a \wedge \neg b = \neg(a \vee b)$;
- склеивания $a \wedge b \vee a \wedge \neg b = a$, $(a \vee b) \wedge (a \vee \neg b) = a$;
- поглощения $a \vee a \wedge b = a$, $a \wedge (a \vee b) = a$;
- законы нуля (лжи) $a \vee 0 = a$, $a \wedge 0 = 0$, $a \wedge \neg a = 0$;
- законы единицы (истины) $a \vee 1 = 1$, $a \wedge 1 = a$, $a \vee \neg a = 1$.

ТАБЛИЦЫ КЭЛИ ВСЕХ БУЛЕВЫХ БИНАРНЫХ ОПЕРАЦИЙ

<u>0</u> 0 1 Функция $f_0(x_1, x_2) = 0$
0 0 0 (константа нуль, *false*, ложь)
1 0 0 0000

$\wedge$ 0 1 $f_1(x_1, x_2) = x_1 \& x_2 = x_1 x_2 = x_{1*} x_2 = x_1 x_2 = x_{1\wedge} x_2$
0 0 0 (конъюнкция, *and*, и)
1 0 1 0001

|] 0 1 Функция $f_2(x_1, x_2) = x_{1\wedge\neg} x_2 = x_{1|]} x_2$
0 0 0 (левая коимпликация)
1 1 0 (*лат.* ***co**nversus = обратный*) 0010

<u>x_1</u> 0 1 Функция $f_3(x_1, x_2) = x_1$
0 0 0 (первый операнд)
1 1 1 0011

[| 0 1 Функция $f_4(x_1, x_2) = \neg x_{1\wedge} x_2 = x_{1[|} x_2$
0 0 1 (правая коимпликация)
1 0 0 (*лат.* ***co**nversus = обратный*) 0100

$\underline{x_2}$ 0 1 Функция $f_5(x_1, x_2) = x_2$ **0 0 1 (второй операнд)** **1 0 1** 0101
$\oplus$ 0 1 $f_6(x_1, x_2) = \neg x_1 \wedge x_2 \vee x_1 \wedge \neg x_2 = x_1 \oplus x_2$ **0 0 1 (неравнозначность, исключающее** **1 1 0 или, *xor*, сложение по модулю 2)** 0110
$\vee$ 0 1 $f_7(x_1, x_2) = x_1@x_2 = x_1 + x_2 = x_1 \vee x_2$ **0 0 1 (дизъюнкция, *or*, или)** **1 1 1** (*лат.* ***v****el* = *или*) 0111
° 0 1 $f_8(x_1, x_2) = \neg x_1 \wedge \neg x_2 = x_1 \circ x_2$ **0 1 0 (функция Вебба)** **1 0 0** 1000
~ 0 1 $f_9(x_1, x_2) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 = x_1 \eta x_2 = x_1 \sim x_2$ **0 1 0 (эквивалентность)** **1 0 1** 1001
$\underline{\neg x_2}$ 0 1 Функция $f_A(x_1, x_2) = f_{10}(x_1, x_2) = \neg x_2$ **0 1 0 (отрицание второго операнда)** **1 1 0** 1010
[0 1 $f_B(x_1, x_2) = f_{11}(x_1, x_2) = \neg x_2 \vee x_1 = x_1 \subset x_2 = x_1 [x_2$ **0 1 0 (правая импликация)** **1 1 1** 1011
$\underline{\neg x_1}$ 0 1 Функция $f_C(x_1, x_2) = f_{12}(x_1, x_2) = \neg x_1$ **0 1 1 (отрицание первого операнда)** **1 0 0** 1100

] 0 1 $f_D(x_1, x_2) = f_{13}(x_1, x_2) = \neg x_{1\vee} x_2 = x_{1\supset} x_2 = x_{1]} x_2$ **0 1 1 (импликация, левая импликация)** **1 0 1** 1101
к 0 1 $f_E(x_1, x_2) = f_{14}(x_1, x_2) = \neg x_1 \vee \neg x_2 = x_{1к} x_2$ **0 1 1 (несовместность, штрих Шеффера)** **1 1 0** 1110
<u>1</u> 0 1 Функция $f_F(x_1, x_2) = f_{15}(x_1, x_2)=1$ **0 1 1 (константа единица, *true*, истина)** **1 1 1** 1111

Встречаются и другие названия: f_8 называют также *стрелкой Пирса* и обозначают: $x_1 \downarrow x_2$.

Законом противоречия называют упоминавшееся уже тождество $x \wedge \neg x = 0$; *законом исключенного третьего* – тождество $x \wedge \neg x = 1$. Используют также тождества $\neg 1=0$ и $\neg 0=1$.

Бросается в глаза, что четыре из шестнадцати бинарных (то есть, двухместных) операций (или, что то же самое, бинарных функций) не зависят от одного из аргументов и являются по существу вырожденными – унарными или, что то же самое, одноместными. Действительно, имеется четыре унарных операции φ (или унарных функции):

x	φ_0	φ_1	φ_2	φ_3
0	0	0	1	1
1	0	1	0	1

Здесь функции (или, что тоже самое, унарные операции, отображения) представляют собой следующее:

$\varphi_0 = 0$ ≈ это константа 0, ложь;

$\varphi_1(x) = x$ – *это единичное отображение множества* {0, 1} *на себя*, x *остается неизменным;*

$\varphi_2(x) = \neg x$ – *это отрицание* x;

$\varphi_3(x)=1$ – *это константа* 1, *истина.*

Вообще, задание таблиц значений функций с перечислением всех возможных комбинаций значений аргументов довольно распространенная в логике форма. В том числе, уже рассмотренные ранее бинарные функции могут быть сведены в единую таблицу:

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_A	f_B	f_C	f_D	f_E	f_F
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Очевидно, столбцы этой таблицы, с одной стороны, представляют собой как бы «вытянутые» таблицы Кэли и, с другой стороны, являются обычными четырехмерными векторами, что позволяет говорить о базисах, взаимозависимостях и о том, что все бинарные логические операции могут быть представлены в виде комбинаций некоторой части логических операций (как мы, в частности, уже и убеждались ранее).

Суперпозицией функций f_1, ..., f_m называется функция f, полученная с помощью подстановок этих функций друг в друга и переименования переменных, а ***формулой*** называется выражение, описывающее эту суперпозицию.

Пусть дано множество (конечное или бесконечное) исходных функций $\sum = \{f_1, ..., f_m\}$. Символы переменных x_1, ..., x_n, ... и констант 0 и 1 считают формулами *глубины* 0. Любая формула имеет ***глубину*** k+1, если она имеет вид $f_i(F_1, ..., F_{nl})$, где f_{tO} S, ni ≈ количество аргументов f_i, а F_1, ..., F_{nl} ≈ формулы, максимальная из глубин которых равна k.

Знак функции (операции) может быть записан перед операндами (*префиксная* или *прямая польская* запись). Знак бинарной операции или функции часто записывают между операндами – такая нотация называется ***инфиксной***. Наконец, для удобства программирования используют и *обратную польскую* (или ***постфиксную***) запись, при которой знак функции или операции располагается после списка операндов. Этот вариант записи позволяет обходиться вообще без скобок, что бывает удобно при трансляции выражений.

Примеры различной записи одной и той же формулы:

1) *and*(x, *o*r(y, z));
2) $x \wedge (y \vee z)$ или x *and* (y *or* z);
3) $xy\ z \vee \wedge$.

8. ИСЧИСЛЕНИЕ ВЫСКАЗЫВАНИЙ

Исчисление высказываний как формальную теорию можно определить с помощью аксиоматического метода.

Аксиоматическая *(формальная) теория T считается определенной, если выполнены следующие условия:*

1) Задано некоторое ***счетное множество***, *т.е. множество, элементы которого могут быть взаимно однозначно сопоставлены элементам натурального ряда 1, 2, ... символов ≈ символов теории T*. Конечные последовательности символов теории называются ***выражениями*** теории T.

2) Имеется подмножество выражений теории T, называемых ***формулами теории*** T (формулами теории часто называют *правильно построенные формулы*). Чтобы определить, является ли выражение формулой в теории, существует эффективная процедура;

3) Выделено некоторое множество формул, называемых ***аксиомами теорииT***;

4) имеется конечное множество R_1, R_2, ..., R_n отношений между формулами, называемых ***правилами вывода***. Для каждого R_i существует натуральное j такое, что для всякого множества, состоящего из j формул, и для всякой формулы F эффективно решается вопрос о том, находятся ли данные j формул в отношении R_i с формулой F и если Z да ι, то F называется *непосредственным следствием данных j формул по правилу R_i*.

Выводом в теории T называется всякая последовательность F_1, F_2, ..., F_m формул такая, что для любого i формула F_i есть либо аксиома теории T, либо непосредственное следствие каких-либо предыдущих формул.

Формула F теории T называется ***теоремой теории*** T, если существует такой вывод в T, в котором последней формулой является F; этот вывод называется выводом формулы F. В общем случае может не существовать эффективной процедуры, с помощью которой можно определить по данной формуле, существует ли ее вывод в теории T.

Формула, для которой такая процедура существует, называется ***разрешимой*** в этой теории, в противном случае ≈ ***неразрешимой***. Иначе говоря, для неразрешимых формул нельзя построить алгоритм выяснения свойства формулы быть теоремой, для этого требуются все новые и новые озарения (изобретательства), не поддающиеся формализации.

Используя понятие аксиоматической теории T, определим исчисление высказываний в дизъюнктивном базисе Буля.

1. Символами T являются $\vee$, $\neg$, (,) и буквы m_i с целыми положительными числами в качестве индексов: m_1, m_2, ... Символы $\vee$, $\neg$, называются *связками*, а буквы m_i ≈ *пропозициональными буквами*.

2. а) Все пропозициональные буквы являются формулами;

б) если A и B ≈ формулы, то $(A \vee B)$ и $(\neg A)$ также формулы;

в) выражение является *формулой* тогда и только тогда, когда это может быть установлено с помощью пп. а) и б).

Таким образом, всякая *формула исчисления высказываний* есть некая пропозициональная форма, построенная из пропозициональных букв с помощью связок $\vee$ и $\neg$.

3. Каковы бы ни были формулы A, B и C теории T, следующие формулы есть *аксиомы* T:

$A \vee A$] A, $A \vee B$] $B \vee A$,

A] $A \vee B$, (B] C)] ($A \vee B$] $A \vee C$),

где запись α] β эквивалентна записи $\neg\, \alpha \vee\, \beta$.

4. Правилами вывода исчисления высказываний являются:

правило подстановки (если α ≈ выводимая формула и вместо любой переменной в этой формуле всюду произвести подстановку любой формулы, то новая формула также является выводимой);

правило заключения (если α] β и α ≈ выводимые формулы, то β также выводимая формула) ≈ правило modus ponens. Символически это правило записывается так: (α, α]β)/β.

Например, если высказывания A и A] (q]A) истинны, то высказывание q]A также является истинным согласно правилу заключения.

Аналогично можно определить и другие булевы алгебры:

алгебру Вебба $A = \langle$M,o$\rangle$;

алгебру Шеффера $A = \langle$M,к$\rangle$;

импликативную алгебру $A = \langle$M,],0$\rangle$;

коимпликативную алгебру $A = \langle$M,],1$\rangle$;

алгебру Жегалкина $A = \langle$M,&,$\oplus$,1$\rangle$.

Обобщением двузначных логик являются конечнозначные логики.

Функция $f(x_1, x_2, ..., x_n)$, отображающая n-мерный k-значный кортеж $(\sigma_1, \sigma_2, ..., \sigma_n), \sigma_i \in \{0,1,...,k-1\}, i = 1,2,...,n$, в множество {0, 1, ..., k √ 1}, называется *функцией k-значной логики*. Будем задавать функцию k-значной логики $f(x_1, x_2, ..., x_n)$ с помощью таблицы истинности (одно-мерной таблицы), число строк которой равно k^n, или двумерной таблицы, число клеток которой равно k^2.

Рассмотрим трехзначную функцию ≈ *функцию Вебба*, заданную таблицами:

x_a	x_b	y
0	0	1
0	1	2
0	2	0
1	0	2
1	1	2
1	2	0
2	0	0
2	1	0
2	2	0

x_a	x_b		
	0	1	2
0	1	2	0
1	2	2	0
2	0	0	0

Функция Вебба

$$y = x_a \lozenge x_b = \max(x_a x_b) + 1(\bmod k)$$

является полной в конечнозначной логике. Таким образом, конечнозначная *алгебра Вебба*

$$A_B = \langle M, o \rangle, M = \{0,1,2,...,k-1\}$$

определяет соответствующую k-значную логику.

Другими часто встречающимися k-значными логиками являются логики, определяемые:

алгеброй Поста

$A_П = \langle M, \vee, \sim \rangle, M = \{0,1,2,...,k-1\},$

где $x_a \vee x_b = \max(x_a, x_b)$ ≈ *дизъюнкция*; $\tilde{x} = x + 1(\bmod k)$ ≈ *цикл*;

алгеброй Россера ≈ *Тьюкетта*;

$A_{РТ} = \langle M, \vee, \&, j_i, i \rangle,\ M = \{0,1,2,...,k-1\}, 0 \le i \le k-1,$

где $x_a \,\&\, x_b = \max(x_a, x_b)$ ≈ *конъюнкция*,

$$j_i(x) = \begin{cases} k-1, & x = i \\ 0 \quad , & x \neq i \end{cases} \approx \textit{характеристические функции}, \ i = 0,1,\ldots,k-1.$$

Алгебраическая система (***алгебра***) **определяется как** $S = \langle A, O, R \rangle$**,** где A ≈ непустое множество, O ≈ семейство операций, R ≈ семейство отношений на множестве A. При этом A называется носителем или основным множеством; операции из O и отношения из R называются основными или главными.

Множество $\Omega = O \cup R$ называется ***сигнатурой алгебры S***.

Система S называется собственно ***алгеброй*** или ***универсальной алгеброй***, если множество R основных отношений пусто и называется ***реляционной системой*** или ***моделью***, если множество O основных операций пусто.

Сигнатура любой алгебры должна быть полной, независимой и непротиворечивой. Сигнатура является *полной*, если любая другая формула может быть представлена в виде пропозициональной формы с помощью ее элементов.

Сигнатура называется *независимой*, если в ней не найдется элемента, выводимого с помощью правил вывода из других элементов сигнатуры.

Сигнатура *непротиворечива*, если не найдется формулы F, которая одновременно справедлива с формулой $\neg F$.

Конечнозначные логики являются обобщением двузначных логик. Например, логика Поста $\langle M, \vee, \sim \rangle$ обобщает логику Буля $\langle M, \vee, \neg \rangle$.

При минимизации логических функций конечнозначных логик можно использовать результаты двузначной логики ≈ теорию ДНФ булевых функций. Для этого введем булеву переменную x_a^i, равную 1 при $x\alpha = i$ и 0 в противном случае, $x\alpha\gamma\ i$. Будем называть x_a^i i-й фазой переменной $x\alpha$, $x\alpha = \{0, 1, ..., k\text{-}1\}$:

$$x_a^i = \begin{cases} 1, & x_a = i \\ 0, & x_a \neq i \end{cases}.$$

Отрицание i-й фазы равно дизъюнкции остальных фаз этой переменной

$$\bar{x}_a^i = \bigvee_{j=0,\, j \neq i}^{k-1} x_a^j.$$

Очевидно, что $x_a^0 \vee x_a^1 \vee K \vee x_a^{k-1} = 1$.

Действительно, $x_a^0 \vee \left(x_a^1 \vee x_a^2 \vee K \vee x_a^{k-1}\right) = x_a^0 \vee \bar{x}_a^0 = 1$.

Предикат (от позднелат. praedicatum – сказанное) – высказывательная функция, определенная на некотором множестве M, т.е. такая n-местная функция P, которая каждому упорядоченному набору $\langle a_1, K, a_n \rangle$ элементов множества M сопоставляет некоторое высказывание, обозначаемое $P(a_1, \ldots, a_n)$; *P* ***называется n-местным предикатом*** **на** *M*. Под 0-местным предикатом понимается произвольное высказывание.

В математической логике высказывание обычно отождествляется с его *истинностным значением* 1 («истина») или 0 («ложь»). При этом понятие предиката получает следующее, наиболее общее определение: *n*-местным предикатом на множестве *M* называется произвольная *n*-местная функция, определенная на *M* и принимающая значения 0 и 1. Если на наборе значений аргументов $\langle a_1, \ldots, a_n \rangle$ предикат *P* принимает значение 1, то есть $P(a_1, \ldots, a_n) = 1$, то говорят, что этот набор значений аргументов удовлетворяет предикату *P*, а предикат *P* выполняется для набора $\langle a_1, \ldots, a_n \rangle$. Предикат называется тождественно истинным, если он выполняется для любого набора значений своих аргументов, и тождественно ложным, если никакой набор не удовлетворяет этому предикату. Предикат называется выполнимым, если он выполняется хотя бы для одного набора значений аргументов.

Множество тех наборов значений аргументов, которые удовлетворяют данному предикату, называется областью истинности этого предиката. Если *P* есть *n*-местный предикат на множестве *M*, то его область истинности является *n*-местным *отношением* на *M*. И наоборот, каждому *n*-местному отношению на множестве *M* однозначно соответствует *n*-местный предикат на *M*. Поэтому изучение предикатов и отношений тесно связано между собой.

С помощью логических операций из данных предикатов можно строить более сложные предикаты. Наряду с теми логическими операциями, которые действуют и над высказываниями, для образования новых предикатов из уже имеющихся применяются ***кванторы***.

Применение ***квантора всеобщности*** $\forall x_i$ к предикату $P(x_1, \ldots, x_n)$, где $1 \le i \le n$ дает $(n-1)$ – местный предикат $\forall x_i P(x_1, \ldots, x_n)$, который набору $\langle a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \rangle$ сопоставляет высказывание, истинное тогда и только тогда, когда для любого значения *a* переменной x_i истинно высказывание $P(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n)$.

Квантор существования $\exists x_i$ в применении к предикату $P(x_1, \ldots, x_n)$ при $1 \le i \le n$ дает (*n*–1)–местный предикат $\exists x_i\, P(x_1, \ldots, x_n)$, который набору $\langle a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \rangle$ сопоставляет высказывание, истинное тогда и только тогда, когда хотя бы для одного значения *a* переменной x_i истинно высказывание $P(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n)$.

Исследованием связей между предикатами, определяемых их логической структурой, занимается логика предикатов.

Предикатная переменная – переменная, возможными значениями которой являются предикаты.

Исчисление предикатов – это общее название *формальных систем*, служащих для формализации логических умозаключений, в которых учитывается как логическая структура суждений (то есть каким образом данное суждение получено из других с помощью *логических операций*), так и их субъектно-предикатная структура, то есть связь между субъектом суждения (о чем говорится в данном суждении) и предикатом (что говорится о субъекте). При этом для логического анализа суждений наряду с такими логическими операциями, как *конъюнкция*, *дизъюнкция*, *импликация*, *эквивалентность*, *отрицание, используются кванторы, а субъектно-предикатная структура уточняется с помощью понятия предиката*.

Поскольку в математической логике интересуются лишь структурой суждений, отвлекаясь от их конкретного смысла, а также во избежание двусмысленностей, свойственных естественным языкам, для построения логики предикатов используется *формализованный язык*, алфавит которого обычно содержит четыре группы символов:

1) предикатные переменные – выражения вида P_n^m, где m и n – неотрицательные целые числа;

2) предметные переменные $x_1, x_2, x_3\ K$;

3) логические символы & (или $\wedge$ – конъюнкция), $\vee$ (дизъюнкция),] (импликация), ~ (эквивалентность), (отрицание), $\exists$ (квантор существования), $\forall$ (квантор всеобщности);

4) вспомогательные символы (,) (скобки) и , (запятая).

Выражение P_n^m называется m-местной предикатной переменной; 0-местные предикатные переменные называются пропозициональными переменными.

Элементарной формулой называется всякая пропозициональная переменная, а также любое выражение вида $P(y_1, K, y_m)$, где P ≈ какая-либо m-местная предикатная переменная ($m > 0$), а y_1, K, y_m ≈ произвольные предметные переменные. Из элементарных формул следующим образом строятся предикатные формулы:

1) все элементарные формулы суть формулы;

2) если ℵ и ℜ ≈ формулы, то выражения (ℵ & ℜ), (ℵ ∨ ℜ), (ℵ] ℜ), (ℵ ~ℜ), ℵ считаются формулами;

3) если ℵ – формула, x – предметная переменная, то ∀ x ℵ и ∃ x ℵ суть формулы.

Например, предикатными формулами являются P_1^0, $P_0^1(x_1)$, $\exists x_1 P_1^2(x_1, x_3)$, $\left(P_0^1(x_2) \& x_1 P_1^2(x_1, x_2)\right)$.

Часть формулы ℵ, которая сама является формулой, называется подформулой формулы ℵ.

Областью действия квантора $\forall y$ или $\exists y$ в формуле $\aleph$ называется такая ее подформула $\Re$, что $\forall y \Re$ или $\exists y \Re$ является подформулой формулы $\aleph$.

Вхождение переменной y в формулу $\aleph$ называется связанным, если оно есть вхождение в квантор $\forall y$ или $\exists y$, или в область действия одного из этих кванторов.

Всякое вхождение переменной y, не являющееся связанным, называется свободным. Например, в формуле $\left(\forall x_1 P_0^1(x_1)\,\&\, P_1^1(x_1)\right)$ первые два вхождения переменной x_1 – связанные, а третье – свободное.

Переменная y называется свободной переменной формулы $\aleph$, если она имеет свободные вхождения в $\aleph$.

Говорят, что задана *интерпретация* формулы $\aleph$ на непустом множестве M, если каждой свободной переменной формулы $\aleph$ сопоставлен некоторый элемент из M, а каждой m-местной предикатной переменной из $\aleph$ – некоторый m-местный предикат на M. Истинностное значение к $\aleph$к формулы $\aleph$ в данной интерпретации определяется индукцией по построению формулы $\aleph$. Если $\aleph$ имеет вид $P(y_1, K, y_m)$, то ее значением является значение предиката, сопоставленного предикатной переменной P, на наборе значений переменных y_1, K, y_m. Если $\aleph$ имеет вид [$\Re$, то к $\aleph$ к = И тогда и только тогда, когда к $\Re$к = Л. Аналогично, в соответствии с истинностными таблицами для логических операций &, $\vee$,] , ~ определяются значения формул вида ($\aleph$ & $\Re$), ($\aleph \vee \Re$), ($\aleph$] $\Re$), ($\aleph \sim \Re$) через значения формул $\aleph$ и $\Re$. Например, к$\aleph$ & $\Re$к = И, тогда и только тогда, когда к $\aleph$к = И и к$\Re$ к = И. Значение формулы $\forall$ y $\aleph$ есть Л в том и только том случае, когда к$\aleph$ к = Л в некоторой интерпретации, полученной из данной приписыванием значения переменной y. Значение формулы $\exists$ $y\aleph$ есть И, если к$\aleph$ к = И в некоторой интерпретации, полученной из данной приписыванием значения переменной y. Если к$\aleph$к = И, то говорят, что формула $\aleph$ истинна в данной интерпретации.

Предикатная формула называется общезначимой на множестве M, если она истинна в любой интерпретации на M. Например, формула

$$\left(\exists x_1 P_0^1(x_1) \to \forall x_2 P_0^1(x_2)\right)$$

общезначима на любом множестве, содержащем ровно один элемент, и не будет общезначимой на M, если в M есть хотя бы два элемента. Формула называется общезначимой или тавтологией, или тождественно истинной, если она общезначима на любом непустом множестве. Тот факт, что формула $\aleph$ общезначима, обычно обозначают так: к=$\aleph$.

Целью логики предикатов является описание класса всех общезначимых формул. Одним из способов такого описания является построение исчисления предикатов, то есть *исчисления*, аксиомами и выводимыми формулами которого являются предикатные формулы. При этом в качестве аксиом выбираются некоторые общезначимые формулы, а правила вывода позволяют из общезначимых формул получать новые общезначимые формулы.

Обычно классическое исчисление предикатов строится на основе того или иного варианта *исчисления высказываний*: аксиомы классического исчисления высказываний считаются схемами аксиом исчисления предикатов, то есть любая предикатная формула, полученная из некоторой аксиомы исчисления высказываний подстановкой в нее каких-либо предикатных формул вместо пропозициональных переменных, объявляется аксиомой исчисления предикатов. Например, из аксиомы исчисления высказываний $A \to (B \to A)$ таким образом получается аксиома исчисления предикатов

$$\left(\forall x_2 P_0^1(x_2) \to \left(P_3^0 \to \forall x_2 P_0^1(x_2)\right)\right).$$

К этим аксиомам добавляются две новые схемы аксиом:

$$(\forall x \aleph(x) \supset \aleph(y)) \text{ и } (\aleph(y) \supset \exists x \aleph(x)),$$

где $\aleph$ (x) ≈ произвольная предикатная формула, в которой переменная x не находится в области действия кванторов $\forall y$ и $\exists y$, а формула $\aleph$ (y) получена заменой в $\aleph$ (x) каждого свободного вхождения переменной x на y.

Правилами вывода исчисления предикатов являются *правило модус поненс* и следующие два правила:

– $\forall$ – правило, позволяющее из формулы ($\Re$] $\aleph$) получить формулу ($\Re$] $\forall x$ $\aleph$), где $\Re$ и $\aleph$ произвольные предикатные формулы, причем $\Re$ не содержит свободную переменную x;
– $\exists$ – правило, позволяющее при тех же предположениях относительно формул $\aleph$, $\Re$ и переменной x перейти от формулы ($\aleph$] $\Re$) к формуле ($\exists x$ $\aleph$] $\Re$) .

Правило модус поненс (правило отделения, правило вывода в формальной логике) означает, что из истинности формулы A (малая посылка) и $A \supset B$ (большая посылка) следует истинность B.

Выводом формулы $\aleph$ в исчислении предикатов называется конечная последовательность формул $\aleph_1$, . . ., $\aleph_m$ такая, что каждая из формул $\aleph_i$ либо есть аксиома, либо получается из некоторых предшествующих ей формул по одному из перечисленных правил вывода, и $\aleph_m$ совпадает с $\aleph$. Формула $\aleph$ выводима в исчислении предикатов, или является теоремой, если можно построить вывод этой формулы. Согласно *теореме Геделя о полноте*, все общезначимые предикатные формулы и только они выводимы в классическом исчислении предикатов.

Дедуктивный аппарат исчисления предикатов, то есть система аксиом и правила вывода, используются при построении логико-математических исчислений (например, *формальной арифметики*, *аксиоматической теории множеств*)

9. ТЕОРИЯ КОДИРОВАНИЯ

В теории передачи информации чрезвычайно важным является решение проблемы кодирования и декодирования, обеспечивающее надежную передачу по каналам связи с «шумом».

Передача информации сводится к передаче по некоторому каналу связи символов некоторого алфавита. Однако в реальных ситуациях сигналы при передаче практически всегда могут искажаться, и переданный символ будет восприниматься неправильно. Например, в системе ЭВМ – ЭВМ одна из вычислительных машин может быть связана с другой через спутник. Канал связи в этом случае физически реализуется электромагнитным полем между поверхностью Земли и спутником. Электромагнитные сигналы, накладываясь на внешнее поле, могут исказиться и ослабиться. Для обеспечения надежности передачи информации в таких системах разработаны эффективные методы, использующие коды различных типов.

Рассмотрим одну из таких моделей, связанную с *групповыми кодами*.

Алфавит, в котором записываются сообщения, считаем состоящим из двух символов {0, 1}. Он называется двоичным алфавитом. Тогда сообщение есть конечная последовательность символов этого алфавита. Сообщение, подлежащее передаче, кодируется по определенной схеме более длинной последовательностью символов в алфавите {0, 1}. Эта последовательность называется *кодом* или *кодовым словом*. При приеме можно исправлять или распознавать ошибки, возникшие при передаче по каналу связи, анализируя информацию, содержащуюся в дополнительных символах. Принятая последовательность символов декодируется по определенной схеме в сообщение, с большой вероятностью совпадающее с переданным.

Блочный двоичный (m, n)–код определяется двумя функциями: E: $\{0, 1\}^m$] $\{0, 1\}^n$ и D: $\{0, 1\}^{n]}$ $\{0, 1\}^m$, где $m \div n$ и $\{0, 1\}^n$ H множество всех двоичных последовательностей длины n. Функция E определяет схему кодирования, а функция D – схему декодирования. Математическую модель системы связи можно представить в виде схемы:

Здесь T– «функция ошибок»; E и D выбираются таким образом, чтобы композиция $D \circ T \circ E$ была функцией, с большой вероятностью близкой к тождественной. Двоичный (m, n)–код содержит 2^m кодовых слов.

Коды делятся на два больших класса: коды с обнаружением ошибок, которые с большой вероятностью определяют наличие ошибки в принятом сообщении, и коды с исправлением ошибок, которые с большой вероятностью могут восстановить посланное сообщение.

Расстояние Хемминга

На множестве двоичных слов длины m расстоянием $d(a, b)$ между словами a и b называют число несовпадающих позиций этих слов, например: расстояние между словами $a = 01101$ и $b = 00111$ равно 2.

Определенное таким образом понятие называется *расстоянием Хемминга*. Оно удовлетворяет следующим *аксиомам расстояний*:

1) $d(a, b)$ ♦ 0 и $d(a, b) = 0$ тогда и только тогда, когда $a = b$;

2) $d(a, b) = d(b, a)$;

3) $d(a, b) + d(b, c)$ ♦ $d(a, c)$ (неравенство треугольника).

Весом $w(a)$ слова a называют число единиц среди его координат. Тогда расстояние между словами a и b есть вес их суммы $a \oplus b$: $d(a, b) = w(a \oplus b)$, где символом $\oplus$ обозначена операция покоординатного сложения по модулю 2.

Интуитивно понятно, что код тем лучше приспособлен к обнаружению и исправлению ошибок, чем больше различаются кодовые слова. Понятие расстояния Хемминга позволяет это уточнить.

Теорема *Для того, чтобы код позволял обнаруживать ошибки в k (или менее) позициях, необходимо и достаточно, чтобы наименьшее расстояние между кодовыми словами было* ♦ $k + 1$.

Доказательство этой теоремы аналогично доказательству следующего утверждения.

Теорема *Для того, чтобы код позволял исправлять все ошибки в k (или менее) позициях, необходимо и достаточно, чтобы наименьшее расстояние между кодовыми словами было* ♦ $2k + 1$.

В математической модели кодирования и декодирования удобно рассматривать строки ошибок. Данное сообщение $a = a_1a_2 \ldots a_m$ кодируется кодовым словом $b = b_1b_2 \ldots b_n$. При передаче канал связи добавляет к нему строку ошибок $e = e_1e_2 \ldots e_n$, так что приемник принимает сигнал $c = c_1c_2 \ldots c_n$, где $c_i = b_i + e_i$. Система, исправляющая ошибки, переводит слово $c_1c_2 \ldots c_n$ в ближайшее кодовое слово $b_1b_2 \ldots b_n$. Система, обнаруживающая ошибки, только устанавливает, является ли принятое слово кодовым или нет. Последнее означает, что при передаче произошла ошибка.

Матричное кодирование

При явном задании схемы кодирования в (m, n)–коде следует указать 2^m кодовых слов, что весьма неэффективно.

Одним из экономных способов описания схемы кодирования является методика матричного кодирования.

Пусть $G = \left\| g_{ij} \right\|$ матрица порядка m β n с элементами g_{ij}, равными 0 или 1. Символ + обозначает сложение по модулю 2. Тогда схема кодирования задается системой уравнений

$$b_j = a_1 g_{1j} + a_2 g_{2j} + \ldots + a_m g_{mj} = \sum_{i=1}^{m} a_i g_{ij}, \quad j = 1,\ldots.,n,$$

или в матричной форме $b = aG$, где $a = a_1 a_2 \ldots a_m$ вектор, соответствующий передаваемому сообщению; $b = b_1 b_2 \ldots b_n$ вектор, соответствующий кодированному сообщению; G порождающая матрица кода.

Порождающая матрица кода определена неоднозначно. Код не должен приписывать различным словам-сообщениям одно и то же кодовое слово. Можно доказать, что этого не произойдет, если первые m столбцов матрицы G образуют единичную матрицу.

Вместо 2^m кодовых слов достаточно знать m слов, являющихся строками матрицы G.

Групповые коды

Двоичный **(**m, n)–код называется *групповым*, если его кодовые слова образуют группу.

Заметим, что множество всех двоичных слов длины m образует коммутативную группу с операцией покоординатного сложения по модулю 2, в которой выполняется соотношение $a \oplus a = 0$. Следовательно, множество слов-сообщений a длины m есть коммутативная группа.

Пусть G порождающая матрица кода порядка m βn. Тогда множество кодовых слов $b = aG$ есть группа, так как если $b^1 = a^1 G$, $b^2 = a^2 G$, то $b^1 + b^2 = a^1 G + a^2 G = (a^1 + a^2)G$, т.е. матричные коды являются групповыми.

В групповом коде наименьшее расстояние между кодовыми словами равно наименьшему весу ненулевого кодового слова, что видно из соотношения $d(b^i, b^j) = w(b^i + b^j)$.

Пусть задан групповой код с порождающей матрицей G и кодирование происходит по схеме $b = aG$.

При передаче кодовые слова могут исказиться, в результате чего буден принято сообщение $c_1 c_2 \ldots c_n$, где $c = b + e$ и $e = e_1 e_2 \ldots e_n$ вектор (строка) ошибок.

Предложим схему декодирования, при которой вероятность того, что $D(aG)$ γ a, будет минимальной.

Обозначим через C множество всех слов, которые могут быть приняты. Это есть множество всех слов длины n. Оно образует коммутативную группу. Множество B всех кодовых слов есть подгруппа C. Рассмотрим множество смежных классов C по B, т.е. фактор-группу C/B. *Лидером смежного класса назовем слово, имеющее наименьший вес. Поскольку смежные классы либо не пересекаются, либо совпадают, то любой элемент* $c \in C$ однозначно представляется в виде суммы $c = e + b$ лидера e и кодового слова b. Декодирование слова c состоит в выборе кодового слова b в качестве переданного и в последующем переходе к слову a, где $b = E(a)$.

Данный метод кодирования удобно реализовать с помощью таблицы, первая строка которой представляет собой множество кодовых слов, т.е. смежный класс $0 + B$, состоящий из элементов $0, b^1, ..., b^{2^m-1}$, а остальные строки соответствуют остальным смежным классам по B, причем первый столбец этой таблицы есть столбец лидеров.

Покажем, что при таком способе декодирования:

1) исправляются все строки ошибок, являющиеся лидерами;

2) кодовое слово, стоящее в данном столбце, является ближайшим кодовым словом ко всем словам этого столбца.

Действительно:

1) Если при передаче слова b произошла ошибка, где e лидер, то $c = b + e$ и есть искомое представление слова c, и при декодировании считается, что передано кодовое слово b, т.е. ошибка исправляется.

2) Пусть c слово, стоящее в том же столбце, что и кодовое слово b^i. Тогда $c = b^i + e$, где e лидер соответствующего смежного класса. Имеем $d(c, b^i) = w(e)$. Если b^j другое кодовое слово, то $c = b^j + eP$ и, поскольку b^i $b^j \in B$, $eP = b^i\, b^j + e$ лежит в том же смежном классе, что и e. Следовательно, $d(c, b^j) = w(eP)$ ♦ $w(e)$.

Коды Хемминга

Опишем один из классов групповых кодов ≈ коды Хемминга, которые исправляют однократную ошибку, поскольку минимальный вес кодового слова равен 3. Это (m, n)-коды, где $m = 2^r - 1$, $n = (2^r - 1 - r)$ для любого r = m-n.

Схема кодирования:

1. Сообщения – слова длины $2^r - 1 - r$, где r ♦ 2; кодовые слова имеют длину $2^r - 1$.

2. В каждом кодовом слове $b = b_1..b_{2^r-1}$. символы, индексы которых являются степенью двойки, т.е. $b_{2^0}, b_{2^1}, K, b_{2^{r-1}}$ – контрольные, а остальные – символы сообщения, расположенные в том же порядке. Например, при $r = 4$ b_1, b_2, b_4, b_8 - контрольные символы, b_3, b_5, b_6, b_7, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15} – символы сообщения.

3. Рассмотрим матрицу M порядка r β $(2^r - 1)$ такую, что в i-м столбце этой матрицы стоят символы двоичного разложения числа i. Тогда матрицы M при r = 2, 3, 4 имеют соответственно вид

$$M_{2,3}=\begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix};\ M_{3,7}=\begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix};\ M_{4,15}=\begin{matrix} 0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \end{matrix}$$

4. Запишем систему уравнений $bM^T = 0$.

По построению матрицы M в каждое из уравнений системы входит один и только один символ b_{2^i}, индекс которого есть степень двойки.

5. При кодировании сообщения значения контрольных символов $b_{2^0}, b_{2^1}, K, b_{2^r-1}$ получим из этой системы (пункт 4).

Схема декодирования.

Пусть принято слово $c = b + e$, где b – кодовое слово, e – ошибка. Тогда $bM^T = 0$, и, следовательно, $(b+e)M^T = bM^T + eM^T = eM^T$.

Если $eM^T = 0$, то считается, что ошибки не было. Это действительно так при $e = 0$.

Если произошла ошибка ровно в одной позиции, т.е. вектор ошибок e имеет только одну единицу в i-й позиции, то $eM^T = 0$ есть вектор, совпадающий с i-м столбцом матрицы M, являющимся двоичным разложением числа i. В этом случае в переданном слове $c = b + e$ надо изменить символ в i-й позиции и вычеркнуть контрольные символы. Тогда полученное слово будет результатом декодирования.

Если ошибка допущена более чем в одной позиции, декодирование дает неверный результат. Например, если строка ошибок e будет кодовым словом, то $(b+e)M^T = 0$ и, следовательно, в результате декодирования слово не изменится.

10. ТЕОРИЯ ГРАФОВ

Граф (от греческого γραφω – пишу) – множество ***V*** вершин и набор ***E*** неупорядоченных и упорядоченных пар вершин; обычно граф обозначают как ***G*(*V*, *E*)**.

Неупорядоченная пара вершин называется ***ребром***, упорядоченная пара – ***дугой***.

Граф, содержащий только ребра, называется ***неориентированным***; граф, содержащий только дуги – ***ориентированным*** (или ***орграфом***).

Пара вершин может быть соединена двумя или более ребрами (или, соответственно, дугами одного направления), такие ребра (или дуги) называются ***кратными***.

Дуга (или ребро) может начинаться и заканчиваться в одной и той же вершине, в этом случае соответственно дуга (или ребро) называется ***петлей***.

Вершины, соединенные ребром или дугой, называются ***смежными***.

Ребра, имеющие общую вершину, тоже называются ***смежными***.

Ребро (или дуга) и любая из его вершин называются ***инцидентными***.

Принято говорить, что ребро (***u, v***) *соединяет* вершины ***u*** и ***v***, а дуга (***u, v***) *начинается* в вершине ***u*** и *заканчивается* в вершине ***v***. Каждый граф можно представить в евклидовом пространстве множеством точек, соответствующих вершинам, которые соединены линиями, соответствующими ребрам (или дугам – в последнем случае направление обычно указывается стрелочками) – такое представление называется ***укладкой*** графа.

Доказано, что в *3-мерном пространстве* любой граф можно представить в виде укладки таким образом, что линии, соответствующие ребрам (дугам) *не будут пересекаться* во внутренних точках. *Для 2-мерного пространства* это, вообще говоря, неверно. Допускают представление в виде укладки в 2-мерном пространстве графы, называемые ***плоскими***.

Существуют и другие способы задания графов. Пусть $\boldsymbol{v_1, v_2, \ldots v_n}$ – вершины графа $\boldsymbol{G(V, E)}$, а $\boldsymbol{e_1, e_2, \ldots e_m}$ – его ребра (дуги).

Матрицей смежности графа G называется матрица $\boldsymbol{A}=||\boldsymbol{a_{ij}}||$, i=1,...,n; j = 1, ..., n, у которой элемент $\boldsymbol{a_{ij}}$ равен числу ребер (или дуг), соединяющих вершины $\boldsymbol{v_i}$ и $\boldsymbol{v_j}$ (соответственно, идущих из вершины $\boldsymbol{v_i}$ в вершину $\boldsymbol{v_j}$).

Матрицей инцидентности* графа *G называется матрица $\boldsymbol{B}=||\boldsymbol{b_{ij}}||$, i=1, .., n; j = 1, ..., m, у которой элемент $\boldsymbol{b_{ij}}$ равен 1, если вершина $\boldsymbol{v_i}$ инцидентна ребру (дуге) $\boldsymbol{e_j}$ и равен 0, если не инцидентна.

Наконец, граф можно задать посредством **списков**. Например,

вариант 1: списком пар вершин, соединенных ребрами (или дугами);

вариант 2: списком списков для каждой вершины множества смежных с ней вершин. 0-граф

Графическое представление

1) A ° ; 2) A ° ° B; 3) A ° ° B ° C;

4) A ° ° B; 5) A ° ° B C ° ° D C °° D ° E;

Матрица смежности (квадратная!)

		A	2)		A	B
	A	0		A	0	0
				B	0	0

и так далее.

Матрица инцидентности (прямоугольная!)

1)		AA	2)		AA	AB	BA	BB
	A	0		A	0	0	0	0
				B	0	0	0	0

и так далее.

Список пар вершин (соответствие)

1) ∅ 2) ∅ и так далее.

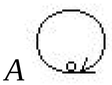
Список списков смежных вершин (простых «поддеревьев»)

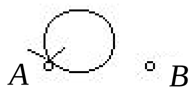
1) *A* ( ) 2) *A* ( ), *B* ( ) и так далее.

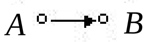
1-й граф

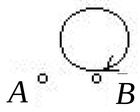
Варианты (с петлями, без петель, дуги, ребра и т.п.)

Графическое представление

1) 

2) а) 

4) б) 

5) 

и так далее.

Матрица смежности

1)		A	2) а)		A	B
	A	1		A	1	0
				B	0	0

2) б)		A	B	2) в)		A	B
	A	0	1		A	0	0
	B	0	0		B	0	1

и так далее.

Матрица инцидентности

1)		AA	2)		AA	AB	BA	BB
	A	1		A	1	0	0	0
				B	0	0	0	0

и так далее.

Список пар вершин

1) (*A*? *A*) 2) а) (*A*? *A*) 2) ,) (*A*? *D*) *b nfr lfktt.*

Список списков смежных вершин

1) *A* (*A*) 2) а) *A* (*A*) 2) б) *A* (*B*) 2) в) *B* (*B*) и так далее.

Два графа ***G*(*V, E*)** и ***H*(*W, I*)** называются ***изоморфными***, если существует взаимно-однозначное соответствие между множествами вершин ***V***, ***W*** и множествами ребер ***E***, ***I***, сохраняющее отношение инцидентности.

Подграфом *GP*(*VP*, *EP*) графа ***G*(*V, E*)** называется граф с множеством вершин $\boldsymbol{VP \subseteq V}$ и множеством ребер (дуг) $\boldsymbol{EP \subseteq E}$, – такими, что каждое ребро (дуга) из ***EP*** инцидентно (инцидентна) только вершинам из ***VP.***

Последовательность ребер (v_0, v_1), (v_1, v_2), ..., $(v_i\text{-}1, v_i)$, $(v_i, v_{i+}1)$, ..., $(v_{r\text{-}}1, v_r)$ называется ***маршрутом***, соединяющим вершины v_0 и v_r. Маршрут замкнут, если $v_0 = v_r$.

Маршрут называется ***цепью***, если все его ребра различны, и ***простой цепью***, если все его вершины различны.

Замкнутая (простая) цепь называется **(*простым*) *циклом***.

Граф называется ***связным***, если любая пара его вершин соединена маршрутом.

Любой максимальный связный подграф (то есть, не содержащийся в других связных подграфах) графа ***G*** называется ***компонентой связности***. Несвязный граф имеет, по крайней мере, две компоненты связности.

Связный граф с наименьшим числом ребер (или связный граф без циклов) называется «***деревом***».

Граф называется *k – связным* (*k – реберно – связным*), если удаление не менее *k* вершин (ребер) приводит к потере свойства связности.

Маршрут, содержащий все вершины или ребра графа и обладающий определенными свойствами, называется ***обходом*** графа.

Длина маршрута (цепи, простой цепи) равна количеству ребер в порядке их прохождения. Длина кратчайшей простой цепи, соединяющей вершины v_i и v_j в графе ***G***, называется ***расстоянием*** $d\ (v_i,\ v_j)$ между v_i и v_j. В связном неориентированном графе расстояние удовлетворяет аксиомам метрики.

С помощью различных операций можно строить графы из более простых, переходить от графа к более простому, разбивать графы на более простые и т.д.

Среди **одноместных операций наиболее употребительны: удаление и добавление ребра или вершины**, **стягивание ребра (отождествление пары смежных вершин), подразбиение ребра (т.е. замена ребра (**u, v) на пару (u, w), (w, v), где w – новая вершина) и др.

Известны **двуместные операции**: **соединение**, **сложение, различные виды умножений графов и др. Такие операции используются для анализа и синтеза графов с заданными свойствами**.

10.1. Эйлеровы пути

Эйлеровым путем в графе называется произвольный путь, проходящий через каждое ребро графа в точности один раз, т.е. путь $v_1, ..., v_m+1$, такой что каждое ребро $e \in E$ появляется в последовательности $v_1, ..., v_m+1$ в точности один раз как $e = \{v_i, v_i+1\}$ для некоторого i.

Если $v_1 = v_m+1$, то такой путь называется ***эйлеровым циклом***. Задача существования эйлерова пути в заданном графе была решена Эйлером в 1736 году, и представленное им необходимое и достаточное условие существования такого пути считается первой в истории теоремой теории графов.

Теорема. *Эйлеров путь в графе существует тогда и только тогда, когда граф связный и содержит не более чем две вершины нечетной степени.*

Доказательство. Доказательство достаточности условия теоремы будет следствием анализа алгоритма нахождения эйлерова пути, который будет описан ниже. Необходимость условия очевидна, так как если вершина v, отличная от вершины v_1 и v_m+1, появляется в эйлеровом пути k раз, то это означает, что степень этой вершины в графе составляет $2k$. Отсюда следует, что вершины нечетной степени, если они существуют, являются концами эйлерова пути. Здесь следует отметить, что не существует графов с одной только вершиной нечетной степени. Действительно, обозначая степень вершины v через $d(v)$, имеем

$$\sum_{v \in V} d(v) = 2m,$$

ибо в указанной сумме каждое ребро $\{u, \text{v}\}$ считается дважды: один раз в $d(u)$ и один раз в $d(\text{v})$. Отсюда следует, что число вершин всегда четно.

Если в связном графе нет вершин нечетной степени, то каждый эйлеров путь является циклом, так как концы эйлерова пути, не являющегося циклом, всегда вершины нечетной степени.

Предположим, что u и v – единственные вершины нечетной степени связного графа $G = < V, E>$, и образуем граф G^* добавлением дополнительной вершины t и ребер $\{u, t\}$ и $\{v, t\}$ (или просто $\{u, v\}$, если $\{u, v\} \notin E$). Тогда G^* – связный граф без вершин нечетной степени, а эйлеровы пути в G будут в очевидном взаимно однозначном соответствии с эйлеровыми циклами в G^*. В силу этого дальше мы будем заниматься только эйлеровыми циклами.

Алгоритм нахождения эйлерова цикла

***Исходные данные*:** связный граф $G = < V, E>$ без вершин нечетной степени, представленный списками *ЗАПИСЬ* $[v]$, $v \in V$.

Результаты*: Эйлеров цикл, представленный последовательностью вершин в стеке СЕ.*

begin

СТЕК := ∅ ; *СЕ* := ∅ ;

v:= произвольная вершина графа;

СТЕК⇐ *v*;

while *СТЕК* γ ∅**do**

begin *v*:= *top* (*СТЕК*); {*v* = верхний элемент стека}

if *ЗАПИСЬ* [*v*] γ ∅ **then**

begin *u*:= первая вершина списка *ЗАПИСЬ* [*v*];

СТЕК⇐ *u*;

{удалить ребро {*v*, *u*} из графа}

ЗАПИСЬ [*v*] := *ЗАПИСЬ* [*v*] \ {*u*};

ЗАПИСЬ [*u*] := *ЗАПИСЬ* [*u*] \ {*v*};

v := *u*

end

else { *ЗАПИСЬ* [*v*] = ∅}

begin *v* ⇐ *СТЕК* ; *СЕ*⇐ *v*;

end

end

end

Принцип действия этого алгоритма можно объяснить следующим образом: пусть v_0 – вершина, выбранная в третьей строке. Цикл начинает строить путь с началом в v_0, причем вершины этого пути помещаются в *СТЕК*, а ребра удаляются из графа.

Эти действия продолжаются вплоть до того момента, когда путь нельзя удлинить, включив в него новую вершину, т.е. когда *ЗАПИСЬ* [*v*] = ∅.

Отметим, что тогда должно быть $v = v_0$, так как в любом другом случае это означало бы, что степень вершины *v* нечетная.

Таким образом, из нашего графа был удален цикл, а вершины этого цикла находятся в стеке *СТЕК*.

Отметим, что в графе, модифицированным таким способом, степень произвольной вершины останется четной.

Вершина $v = v_0$ переносится теперь из стека *СТЕК* в стек *СЕ*, а «очередной» вершиной *v* становится верхний элемент стека *СТЕК*.

Процесс повторяется, начиная с этой вершины (если *ЗАПИСЬ* [*v*] ≠ ∅), в результате чего вследствие четности степени всех вершин находится и помещается в стек *СТЕК* некоторый цикл, проходящий через вершину *v*.

Это продолжается до того момента, пока *СТЕК* не станет пустым.

Очевидно, что вершины, помещаемые в стек *СЕ*, образуют некоторый путь, причем вершина *v* переносится в стек *СЕ* только тогда, когда *ЗАПИСЬ* [*v*] = ∅, т.е. когда все ребра, инцидентные с этой вершиной, представлены (парами соседних вершин) в одном из стеков. Отсюда легко следует, что по окончании алгоритма стек *CE* содержит эйлеров цикл.

Оценим теперь вычислительную сложность этого алгоритма. Для этого отметим, что каждая итерация главного цикла либо помещает вершину в стек *СТЕК* и удаляет ребро из графа, либо переносит вершину из стека *СТЕК* в стек *CE*. Таким образом, число итераций этого цикла – O(m). В свою очередь число шагов в каждой итерации ограничено константой. Здесь предполагаем, что каждый из списков инцидентности *ЗАПИСЬ* [v], $v \in V$, реализован таким образом, что каждая вершина в этом списке содержит указатели на предыдущую и последующую вершины, и вершина u в списке *ЗАПИСЬ* [v] содержит указатель на вершину v в списке *ЗАПИСЬ* [u]. Это дает возможность устранить ребро $\{u, v\}$ за время, ограниченное константой. Из приведенных выше рассуждений можно сделать вывод, что общая сложность алгоритма есть O(m). Очевидно, что этот алгоритм оптимальный, так как уже одно выписывание эйлерова цикла требует Ω (m) шагов. Первые задачи теории графов были связаны с решением математических развлекательных задач и головоломок, например:

– ***задача о кенигсбергских мостах*** (задача Эйлера), развитие которой привело к циклу задач об обходах графов;
– ***задачи о перевозках***, решение которых привело к созданию эффективных методов решения ***транспортных задач*** и др.

Попытки решить сформулированную в середине XIX в. ***задачу четырех красок*** привели к появлению некоторых исследований графов, имеющих теоретическое и прикладное значение.

Многие результаты середины XIX в., относящиеся к теории графов, были получены при решении практических проблем.

Так, Г. Кирхгоф при составлении ***полной системы уравнений для токов и напряжений в электрической схеме*** предложил, по существу, представлять такую схему графом и находить в нем деревья, с помощью которых выделяются линейно независимые системы контуров.

А. Кэли, исходя из задач подсчета числа изомеров предельных углеводородов, пришел к задачам ***перечисления и подсчета «деревьев», обладающих заданными свойствами*****, и решил некоторые из них.**

В XX в. задачи, связанные с графами, начали возникать не только в физике, электротехнике, химии, биологии, экономике, социологии и т.д., но и внутри математики, в таких ее разделах, как алгебра, топология, теория вероятностей, теория чисел и др. Методы этих разделов стали успешно использоваться для решения задач теории графов.

Наряду с термином *граф* в начале XX в. употреблялись в качестве синонимов и другие термины, например, *карта, комплекс, диаграмма*, *сеть, лабиринт*.

В проблематике теории графов можно выделить направления, носящие более ***комбинаторный*** или более ***геометрический*** характер.

К комбинаторным относятся, например, задачи о ***построении графов с заданными свойствами***, задачи о ***подсчете и перечислении графов с фиксированными свойствами***.

Геометрический (топологический) характер носят, например, задачи, связанные с ***обходами графа***, и задачи, возникающие ***при укладке графа на различных поверхностях***.

Примером результата о существовании графа с фиксированными свойствами может служить ***критерий существования графа с заданными степенями вершин*** (***степень вершины*** v есть число ребер, инцидентных v): набор целых чисел $0 \div d_1 \div d_2 \div \ldots \div d_n$, сумма которых четна, является набором степеней вершин графа без петель и кратных ребер тогда и только тогда, когда для любого числа r, $1 \div r \div n - 1$, выполняется условие

$$\sum_{i=1}^{r} d_i \le r(r-1) + \sum_{i=r+1}^{n} \min(r, d_i)$$

Примерами задач о подсчете являются задачи о нахождении ***количеств неизоморфных графов с одинаковым числом вершин и (или) ребер***. Получены асимптотические формулы для числа неизоморфных «деревьев» с n вершинами и для числа неизоморфных графов без петель и кратных ребер с n вершинами.

Примером результатов геометрического направления является ***выделение маршрутов, содержащих все вершины или все ребра графа***. Известен ***критерий существования обхода всех ребер графа***: в связном графе цикл, содержащий все ребра и проходящий по каждому ребру только один раз, существует тогда и только тогда, когда все вершины графа имеют четные степени.

Для аналогичного обхода вершин известен только ряд ***достаточных условий существования цикла***. Таковы, например, все плоские 4-связные графы.

Получено ***необходимое и достаточное условие вложения графа в плоскость***: граф является плоским тогда и только тогда, когда он не содержит подграфов, получаемых с помощью подразбиения ребер из полного 5-вершинного графа и полного двудольного графа с тремя вершинами в каждой доле.

Наряду с проблемами, носящими общий математический характер, в теории графов имеются специфические задачи. Например, изучаются различные свойства связности графа, исследуется строение графа по свойствам связности. При *анализе надежности сетей связи, электронных схем, коммуникационных сетей* возникает задача о *нахождении количеств непересекающихся цепей, соединяющих различные вершины графа*. Так, например, наименьшее число вершин, разделяющих две несмежные вершины графа, равно наибольшему числу непересекающихся по вершинам простых цепей, соединяющих эту пару вершин.

Найдены критерии и построены эффективные алгоритмы установления k – связности графа.

Характерным специфическим направлением теории графов является цикл проблем, связанных с ***раскрасками графов***, в которых изучаются разбиения множества вершин (ребер), обладающие определенными свойствами, например, смежные вершины (ребра) должны принадлежать различным множествам (вершины и ребра из одного множества окрашиваются одним цветом). Так, было доказано, что наименьшее число цветов, достаточное для раскраски ребер любого графа без петель с максимальной степенью вершин σ, равно $[\ 3\ \sigma/\ 2\ ]$, а для раскраски вершин любого графа без петель и кратных ребер достаточно $\sigma + 1$ цветов.

С привлечением методов топологии изучаются ***вложения графов в различные поверхности***. Построены алгоритмы нахождения минимального рода (при условии, что он ограничен константой) ориентируемой поверхности, на которой можно расположить граф. Для решения задач, связанных с ***печатным монтажем электронных схем***, модельной является задача о ***разбиении данного графа на минимальное число плоских графов***.

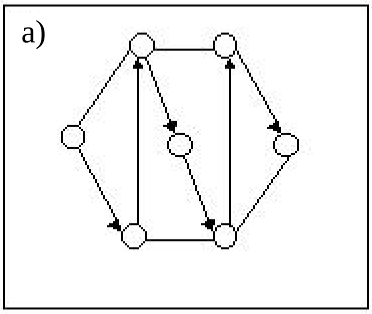
Средствами алгебры изучаются ***группы автоморфизмов графов*** (т.е. изоморфных отображений графов на себя). Было доказано, что каждая конечная абстрактная группа есть группа автоморфизмов (симметрий) некоторого графа.

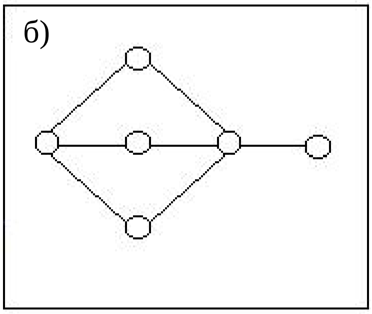
Построены ***алгоритмы установления изоморфизма графов***: графы, у которых степени вершин ограничены константой; графы, у которых собственные значения матрицы смежности ограничены константой; графы, допускающие вложение в поверхность ограниченного константой рода.

Как одна из математических моделей теории вероятностей изучаются ***случайные*** графы. Это понятие возникает, например, в результате задания для каждой пары вершин v_i и v_j вероятности p_{ij} существования ребра между ними (с вероятностью $q_{ij} = 1 - p_{ij}$ этого ребра нет). Многие свойства были изучены для «почти всех» графов, т.е. когда всех $p_{ij} = 1/2$.

10.2. Гамильтоновы пути

Перейдем теперь к задаче, рассмотренной выше, с той разницей, что на этот раз нас будут интересовать пути, проходящие в точности один раз через каждую вершину (а не каждое ребро) данного графа. Эта небольшая модификация приводит к значительному изменению характера проблемы. Путь с указанным свойством называется ***гамильтоновым путем*** (*гамильтонов цикл определяется очевидным образом). Пример гамильтонова пути в графе и графа, в котором такого пути не существует, показан на рис.* (*а*), (*б*) *соответственно.*





В отличие от эйлеровых путей не известно ни одного простого необходимого и достаточного условия для существования гамильтоновых путей и это несмотря на то, что эта задача - одна из самых центральных в теории графов.

Не известен также алгоритм, который проверял бы существование гамильтонова пути в произвольном графе, используя число шагов, ограниченное многочленом от переменной n (числа вершин в графе).

Имеются некоторые основания полагать, что нет математического доказательства того, чтобы предполагать, что такое положение вещей вызвано не столько нашим незнанием, а скорее тем фактом, что такого алгоритма не существует.

Проблема существования гамильтонова пути принадлежит к классу так называемых *NP-полных задач.* Это широкий касс задач, включающий фундаментальные задачи из теории графов, логики, теории чисел, дискретной оптимизации и других дисциплин, ни для одной из которых не известен полиномиальный алгоритм (т.е. с числом шагов, ограниченным полиномом от размерности задачи), причем существование полиномиального алгоритма для хотя бы одной из них автоматически влекло бы за собой существование полиномиальных алгоритмов для всех этих задач. Именно факт фундаментальности многих *NP*-полных задач в различных областях и то, что, несмотря на независимые друг от друга усилия специалистов в этих областях, не удалось найти полиномиального алгоритма ни для одной из этих задач, склоняет к предположению, что такого алгоритма не существует.

Вернемся к конкретной задаче *существования гамильтонова пути*. Очевидный алгоритм, который мы можем применить, это *полный перебор всех возможностей*: генерируем все $n!$ различных последовательностей вершин и для каждой из них проверяем, определяет ли она гамильтонов путь. Такие действия требуют по меньшей мере $n!\ n$ шагов, но функция от n подобного вида растет быстрее, чем произвольный многочлен, и даже быстрее, чем произвольная экспоненциальная функция вида a^n, $a > 1$, ибо

$$n!n[n/2]^{[n/2]} = a^{[n/2]\log_a[n/2]}.$$

Опишем теперь общий метод, позволяющий значительно сократить число шагов в алгоритмах типа полного перебора всех возможностей. Чтобы применить этот метод, искомое решение должно иметь вид последовательности $\langle x_1, ..., x_n\rangle$.

Основная идея метода состоит в том, что мы строим наше решение последовательно, начиная с пустой последовательности ε (длины 0). Вообще, имея данное частичное решение $\langle x_1, ..., x_i\rangle$, мы стараемся найти такое допустимое значение x_i+1, относительно которого мы не можем сразу заключить, что $\langle x_1, ..., x_i, x_i+1\rangle$можно расширить до некоторого решения (либо $\langle x_1, ..., x_i, x_i+1\rangle$ уже является решением).

Если такое предполагаемое, но еще не использованное значение x_i+1 существует, то мы добавляем его к нашему частичному решению и продолжаем процесс для последовательности $\langle x_1, ..., x_i, x_i+1\rangle$. Если его не существует, то мы возвращаемся к нашему частичному решению $\langle x_1, ..., x_i–1\rangle$ и продолжаем наш процесс, отыскивая новое, еще не использованное допустимое значение xP_i – отсюда название «*алгоритм с возвратом*» (англ.: *backtracking*).

Точнее говоря, мы предполагаем, что для каждого $k > 0$ существует некоторое множество A_k, из которого мы будем выбирать кандидатов для k-й координаты частичного решения.

Очевидно, что множества A_k должны быть определены таким образом, что для каждого целочисленного решения $\langle x_1, ..., x_n\rangle$нашей задачи и для каждого $k\div n$ множество A_k содержало элемент x_k (на практике мы не можем вообще исключить ситуацию, когда множество A_k содержит некоторые «лишние» элементы, не появляющиеся в k-й координате ни одного целочисленного решения). Мы предполагаем также, что существует некоторая простая функция, которая произвольному частичному решению $\langle x_1, ..., x_i\rangle$ ставит в соответствие значение $P(x_1, ..., x_i)$ (истина либо ложь) таким образом, что если $P(x_1, ..., x_i)$ = ложь, то последовательность $\langle x_1, ..., x_i\rangle$ несомненно нельзя расширить до решения. Если $P(x_1, ..., x_i)$ = истина, то мы говорим, что значение x_i *допустимо* (*для частичного решения* $\langle x_1, . . ., x_i$-1$\rangle$), но это отнюдь не означает, что $\langle x_1, . . ., x_i$-1$\rangle$ обязательно расширяется до полного решения. Этот процесс можно записать в виде следующей схемы:

begin
k:= 1;
while $k > 0$ **do**
if существует еще неиспользованный элемент $y \in A_k$
такой, что $P(X[1], ..., X[k - 1], y)$ **then**
begin $X[k] := y$; { элемент y уже использован }
if $\langle X[1], . . ., X[k]\rangle$ является целочисленным решением **then**
write(X[1], . . ., X[k]);
k:= k +1;
end

else {возврат на более короткое частичное решение; все элементы множества A_k вновь становятся неиспользованными}

k:= k - 1

end

Этот алгоритм находит все решения в предположении, что множества A_k конечные и что существует n, такое что $P(x_1, . . ., x_n)$ = ложь для всех $x_1 \in A_1$, ..., $x_n \in A_n$ (последнее условие означает, что все решения имеют длину меньше n).

Покажем наличие более общего свойства:

Пусть $s > 0$, и пусть $\langle x_1, ..., x_s -1\rangle$ – некоторое частичное решение, построенное алгоритмом. Рассмотрим первую итерацию цикла, для которой $k = s$, $X[i] = x_i$, $1 \div i \div s$. Начиная с этой итерации, алгоритм генерирует все целочисленные решения, являющиеся расширением последовательности $\langle x_1, ..., x_s -1\rangle$и приходит к состоянию, когда $k = s - 1$.

Очевидно, что для $s = 1$ приведенное выше свойство означает просто корректность алгоритма.

Доказательство этого свойства в общем виде проводится по «индукции назад» относительно s. Оно имеет место для $s = n$, потому что не существует ни одного допустимого элемента для $\langle x_1, ..., x_s–1\rangle$ и сразу же выполняется второе условие условного оператора (по **else**), т.е. переменная k принимает значение $s – 1$.

Предположим теперь правильность нашего свойства для некоторого $s > 1$. Покажем его правильность для $s – 1$. Пусть $\langle x_1, ..., x_s–2\rangle$ – произвольное частичное решение, построенное нашим алгоритмом; рассмотрим первую итерацию цикла, для которой $k = s – 1$, $X[i] = x_i$, $1 \div i \div s – 1$. Если существует элемент y, допустимый для $\langle x_1, ..., x_s–2\rangle$, то построение дает частичное решение $\langle x_1, ..., x_s –2, y\rangle$ и переменная k принимает значение s. Затем согласно нашему индуктивному предположению будут сгенерированы все решения, являющиеся расширением последовательности $\langle x_1, ..., x_s–2, y\rangle$, и мы приходим к состоянию, когда $k = s – 1$, после чего процесс повторяется для следующего неиспользованного элемента y, допустимого для $\langle x_1, ..., x_s–2\rangle$, пока не будут использованы все такие элементы (такая ситуация может появиться сразу с самого начала). Переменная k уменьшается тогда на 1 и принимает значение $s – 2$. Из приведенных выше рассуждений следует, что алгоритм правильно порождает все решения, являющиеся расширением последовательности $\langle x_1, ..., x_s –2\rangle$, что завершает доказательство шага индукции.

Доказанное свойство приводит непосредственно к следующему простому и очевидному рекурсивному варианту схемы алгоритма с возвратом: **procedure** $AP(k)$

{генерирование всех решений, являющихся расширением последовательности

X[1], ..., X[k – 1]: массив X – глобальный}
begin
for $y \in A_k$ такого, что $P(X[1], ..., X[k - 1], y)$ **do**
begin $X[k] := y$;
if X[1], ..., X[k] есть целочисленное решение **then** write(X[1], ..., X[k]);
AP(k +1);
end
end

Генерирование всех целочисленных решений можно вызвать вызовом *AP*(1). Представление алгоритма с возвратом мы начали с несколько более сложного нерекурсивного варианта только потому, что в рекурсивном варианте «возврат» не появляется в явном виде, будучи частью реализации механизма рекурсии.

Применим теперь алгоритм с возвратом для нахождения гамильтонова цикла в графе $G = <V, E>$. Каждый такой цикл можно представить последовательностью $\langle x_1, x_2, ..., x_n+1\rangle$, причем $x_1 = x_n+1 = v_0$, где v_0 – некоторая фиксированная вершина графа, $x_i – x_i+1$ для $1 \div i \div n$ и $x_i\, \gamma x_j$ для $1 \div i < j \div n$. Согласно этим условиям мы можем положить:

$A_k = V$,

$P \langle x_1, ..., x_k - 1, y\rangle \Leftrightarrow y \in$ ЗАПИСЬ $[x_k - 1] \wedge y \in \{x_1, ..., x_k - 1\}$.

Алгоритм нахождения всех гамильтоновых циклов в графе

Данные: Граф $G = <V, E>$, представленный списками инцидентности ЗАПИСЬ[v], $v \in V$.

Результаты: Список всех гамильтоновых циклов графа G

procedure *ГАМИЛЬТ*(k)
{генерирование всех гамильтоновых циклов, являющихся расширением последовательности X[1], ..., X[k – 1]: массив X – глобальный}
begin
for $y \in$ ЗАПИСЬ [X[k – 1]] **do**
if ($k = n$ +1) **and** ($y = v0$) **then** write(X[1], ..., X[n], $v0$)
else if *DOP*[y] **then**
begin $X[k] := y$; *DOP*[y] := **ложь**;
ГАМИЛЬТ (k +1);
DOP[y] := **истина**
end
end; { *ГАМИЛЬТ* }
begin { главная программа }
for $v \in V$ **do** *DOP*[v] := **истина; {инициализация}**
$X[1] := v0$; {$v0$ = произвольная фиксированная вершина графа}
DOP[$v0$] := **ложь**;

ГАМИЛЬТ (2);

end

Работу этого алгоритма, так же как и произвольного алгоритма с возвратом, можно проиллюстрировать процессом поиска в некотором «дереве». Каждая вершина «дерева» естественным образом соответствует некоторой последовательности $\langle x_1, ..., x_k \rangle$, причем вершины, соответствующие последовательностям вида $\langle x_1, ..., x_k, y \rangle$ являются «сыновьями» этой вершины (корень соответствует пустой последовательности ε). Рассмотрим полное «дерево» D, состоящее из всех возможных последовательностей вида $\langle x_1, ..., x_k \rangle$, где $0 \div k \div n$ и $x_i \in A_i$ для $1 \div i \div k$, и временно допустим, что каждый элемент $y \in A_k$ является допустимым для $\langle x_1, ..., x_k - 1 \rangle$, если $k \div n$, и ни один элемент не является допустимым для $\langle x_1, ..., x_k - 1 \rangle$, если $k > n$; другими словами, $P(x_1, ..., x_k - 1, x_k)$ – **истина, если** $k \div n$, и $P(x_1, ..., x_k - 1, x_k) =$ **ложь, если** $k \div n$ ($x_i \in A_i$ для $1 \div i \div k$). Тогда нетрудно отметить, что вызов $AP(1)$ вызывает поиск в глубину во всем дереве D (начиная от корня ε). Для случая менее тривиальной функции P, определяющей допустимость вершин, процесс поиска «опускает» рассмотрение вершин поддерева с корнем в каждой встреченной «недопустимой» вершине, т.е. вершине, соответствующей последовательности $\langle x_1, ..., x_k \rangle$, для которой $P(x_1, ..., x_k)$ – ложь). В этом, собственно говоря, заключается эффективность алгоритма с возвратом в отношении полного перебора всех возможностей.

Следует отметить, что для большинства приложений число шагов алгоритма с возвратом хотя и будет меньше, чем в случае полного перебора, однако же в наихудшем случае растет по экспоненте с возрастанием размерности задачи. Это справедливо и для случая, если мы ищем только одно, а не все решения (тогда мы просто прерываем выполнение алгоритма после получения первого решения; отметим, что когда задача не имеет решения, эта модификация не влияет на ход выполнения алгоритма).

10.3. Кратчайшие пути в графах

Начальные понятия

Рассмотрим ориентированные графы $G = \langle V, E \rangle$, дугам которых приписаны веса. Это означает, что каждой дуге $\langle u, v \rangle \in E$ поставлено в соответствие некоторое вещественное число $a(u, v)$, называемое *весом* данной дуги.

Полагаем, кроме того, $a(u, v) =$ Т , если u -а v.

Если последовательность вершин $v_0, v_1, ..., v_p$ определяет путь в G, то его *длина* определяется как сумма

$$\sum_{i=1}^{p} a(v_{i-1}, v_i).$$

(Отметим, что если в произвольном графе мы примем вес каждой дуги равным единице, то мы получим обычное определение длины пути как числа дуг; длина пути равна 0 при $p = 0$).

Нас будет интересовать нахождение кратчайшего пути между фиксированными вершинами $s, t \in V$. Длину такого кратчайшего пути мы будем обозначать $d\ (s, t)$ и называть *расстоянием* от s до t (расстояние, определенное таким образом, может быть отрицательным). Если не существует ни одного пути из s в t, то полагаем $d\ (s, t) =$ Т. Если каждый контур нашего графа имеет положительную длину, то *кратчайший* путь будет всегда *элементарным* путем, т.е. в последовательности $v_1,..., v_p$ не будет повторов.

С другой стороны, если в графе существует контур отрицательной длины, то расстояние между некоторыми парами вершин становится неопределенным, потому что, обходя этот контур достаточное число раз, мы можем показать путь между этими вершинами с длиной, меньшей произвольного вещественного числа. В таком случае, можно было бы говорить о длине кратчайшего элементарного пути, однако задача, поставленная таким образом, вероятно будет значительно более сложной, так как, в частности, она содержит в себе задачу существования гамильтонова пути.

Можно дать много практических интерпретаций задачи о кратчайших путях. *Например*, вершины могут соответствовать городам, а каждая дуга – некоторому пути, *длина* которого представлена весом дуги. Мы ищем затем кратчайшие пути между городами. Вес дуги также может соответствовать *стоимости* (или *времени*) передачи информации между вершинами. В таком случае мы ищем самый дешевый (или самый скорый) путь передачи информации. Еще одну ситуацию получаем, когда вес дуги $<u, v>$ равен *вероятности* $p(u, v)$ безаварийной работы канала передачи информации. Если предположить, что аварии каналов не зависят друг от друга, то вероятность исправности пути передачи информации равна произведению вероятностей составляющих его дуг. Задачу нахождения наиболее надежного пути легко можно свести к задаче о кратчайшем пути, заменяя веса $p(u, v)$ на $a\ (u, v) = -\log p(u, v)$.

Сначала рассмотрим *алгоритмы нахождения расстояния между вершинами*, а не самих путей. Однако, зная расстояние, мы можем при условии положительной длины всех контуров легко определить кратчайшие пути. Для этого достаточно отметить, что для произвольных $s, t \in V$ ($s\ \gamma\ t$) существует вершина v, такая что $d\ (s, t) = d\ (s, v) + a\ (v, t)$.

Действительно, таким свойством обладает предпоследняя вершина произвольного кратчайшего пути из s в t.

Далее мы можем найти вершину u, для которой $d\ (s, v) = d\ (s, u) + a\ (u, v)$, и т.д.

Из положительности длины всех контуров легко следует, что созданная таким образом последовательность t, v, u, ... не содержит повторений и оканчивается вершиной s.

Очевидно, что она определяет (при обращении очередности) кратчайший путь из s в t.

Таким образом, мы получаем следующий алгоритм:

Алгоритм нахождения кратчайшего пути

Данные: Расстояния $D[v]$ от фиксированной вершины s до всех остальных вершин $v \in V$, фиксированная вершина t, матрица весов ребер, $A[u, v]$, $u, v \in V$.

Результаты: *СТЕК* содержит последовательность вершин, определяющую кратчайший путь из s в t.

begin
СТЕК := $\varnothing$; *СТЕК* $\Leftarrow t$; $v := t$;
while $v \gamma s$ **do**
begin
u := вершина, для которой $D[v] = D[u] + A[u, v]$;
СТЕК $\Leftarrow u$;
$v := u$
end
end.

Пусть $<V, E>$ – ориентированный граф, $|V| = n$, $|E| = m$. Если выбор вершины u происходит в результате просмотра всех вершин, то сложность нашего алгоритма – $O(n^2)$. Если мы просматриваем только список *ПРЕДШ* $[v]$, содержащий все вершины u, такие что $u\ (\rho)\ v$, то в этом случае сложность будет $O(m)$.

Отметим, что в случае положительных весов ребер задача о кратчайшем пути в *неориентированном* графе легко сводится к аналогичной задаче для некоторого ориентированного графа. С этой целью достаточно заменить каждое ребро $\{u, v\}$двумя дугами $\langle u, v\rangle$ и $\langle v, u\rangle$, каждая с таким же весом, что и $\{u, v\}$. Однако в случае неположительных весов это приводит к возникновению контуров с неположительной длиной.

Далее будем всегда предполагать, что $G = < V, E>$ является ориентированным графом, $|V| = n$, $|E| = m$. В целях упрощения изложения и избежания вырожденных случаев при оценке сложности алгоритмов будем исключать ситуации, при которых «большинство» вершин изолированные.

Будем также предполагать, что веса дуг запоминаются в массиве $A[u, v]$, $u, v \in V$ ($A[u, v]$ – содержит вес $a\ (u, v)$).

Кратчайшие пути от фиксированной вершины

Большинство известных алгоритмов нахождения расстояния между двумя фиксированными вершинами s и t опирается на действия, которые в общих чертах можно представить следующим образом: при данной матрице весов дуг $A[u, v]$, $u, v \in V$, вычисляются некоторые верхние ограничения $D[v]$ на расстояния от s до всех вершин $v \in V$. Каждый раз, когда мы устанавливаем, что

$$D[u] + A[u, v] < D[v],$$

оценку $D[v]$ улучшаем: $D[v] = D[u] + A[u, v]$.

Процесс прерывается, когда дальнейшее улучшение ни одного из ограничений невозможно.

Легко можно показать, что значение каждой из переменных $D[v]$ равно тогда $d(s, v)$ – расстоянию от s до v.

Заметим, что для того чтобы определить расстояние от s до t, мы вычисляем здесь расстояния от s до всех вершин графа.

Не известен ни один алгоритм нахождения расстояния между двумя фиксированными вершинами, который был бы существенным образом более эффективным, нежели известные алгоритмы определения расстояния от фиксированной вершины до всех остальных.

Описанная общая схема является неполной, так как она не определяет очередности, в которой выбираются вершины u и v для проверки условия минимальности расстояния. Эта очередности, как будет показано ниже, очень сильно влияет на эффективность алгоритма. Опишем теперь более детально методы нахождения расстояния от фиксированной вершины, называемой *источником*, его всегда будем обозначать через s, до всех остальных вершин графа.

10.4. Виды графов

Сначала представим алгоритм для общего случая, в котором предполагается только отсутствие контуров с отрицательной длиной. С эти алгоритмом обычно связывают имена *Л.Р. Форда* и *Р.Е. Беллмана*.

Алгоритм нахождения расстояния от источника до всех вершин – метод Форда – Беллмана

Данные: Ориентированный граф $<V, E>$ с n вершинами с выделенным источником $s \in V$, матрица дуг $A[u, v]$, $u, v \in V$ (граф не содержит контуров отрицательной длины).

***Результаты*:** Расстояния от источника до всех вершин графа: $D[v]=d(s, v)$, $v \in V$.

1 begin
2 for $v \in V$ **do** $D[v] := A[s, v]$; $D[s] := 0$;
3 for $k := 1$ **to** $n - 2$ **do**
4 for $v \in V \setminus \{s\}$ **do**

5 for $u \in V$ **do** $D[v] := \min(D[v], D[u] + A[u, v])$
6 end

Докажем правильность **этого алгоритма.** Для этого обозначим через $d^{(m)}(v)$ длину кратчайшего из путей из s в v, содержащих не более m дуг ($d^{(m)}(v)$ = Т , если не существует ни одного такого пути). Тогда получим

$$d^{(m+1)}(v) = \min\{d^{(m)}(u) + a[u, v]: u \in V\}, v \in V \qquad (1)$$

На самом деле, для каждого $u \in V$ очевидно имеем $d^{(m+1)}(v) \div d^{(m)}(u) + a[u, v]$, причем равенство появляется, когда u является предпоследней вершиной произвольного кратчайшего пути из s в v.

Покажем теперь, что если при входе в очередную итерацию цикла 3

$$d(s, v) \div D[v] \div d^{(m)}(v) \text{ для всех } v \in V, \qquad (2)$$

то по окончании этой итерации

$$d(s, v) \div D[v] \div d^{(m+1)}(v) \text{ для всех } v \in V. \qquad (3)$$

Действительно, предполагая выполнение условия (2) и анализируя действие оператора в строке 5, мы видим, что по окончании итерации цикла 3 имеем

$$d(s, v) \div D[v] \div \min\{d^{(m)}(u) + a[u, v]: u \in V\},$$

что, принимая во внимание равенство (1) дает условие (3).

Отметим, что при входе в цикл 3 имеем $D[v] = d\,1\,(v)$, $v \in V$, следовательно, после выполнения $n - 2$ итераций этого цикла будут выполняться неравенства

$$d(s, v) \div D[v] \div d^{(n-1)}(v), v \in V.$$

Теперь достаточно показать, что $d^{(n-1)}(v) = d(s, v)$. Это справедливо, поскольку каждый путь более чем с $n - 1$ дугами содержит контур, устранение которого не увеличивает длины пути (ведь мы предполагаем отсутствие контуров отрицательной длины). Тем самым закончено ***доказательство корректности алгоритма***.

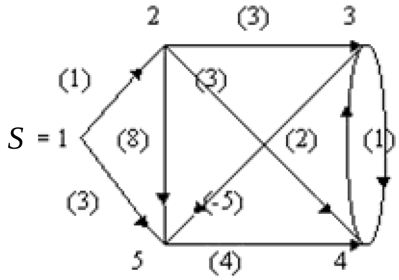
Очевидно, что ***временная сложность алгоритма*** есть $O(n^3)$. Мы можем, конечно, закончить вычисления, когда выполнение цикла 4 не вызывает изменения ни одной из переменных $D[v]$, $v \in V$. Это может наступить для $k < n - 2$, однако такая модификация алгоритма не изменяет существенным образом его сложности. Для редких графов ($m << n^2$) удобнее представлять граф списками инцидентности $ПРЕДШ[v]$, $v \in V$. Заменяя строку 5 на

for $u \in ПРЕДШ[v]$ **do** $D[v] := \min(D[v], D[u] + A[u, v])$,

получаем алгоритм со сложностью $O(nm)$.

Работа алгоритма Форда - Беллмана проиллюстрирована на следующем рисунке ($V = \{1, ..., 5\}$, веса дуг даны числами в скобках, циклы 4 и 5 выполняются в порядке возрастания номеров вершин).

$$A = \begin{bmatrix} \infty & 1 & \infty & \infty & 3 \\ \infty & \infty & 3 & 3 & 8 \\ \infty & \infty & \infty & 1 & -5 \\ \infty & \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & 4 & \infty \end{bmatrix}$$



k	D[1]	D[2]	D[3]	D[4]	D[5]
	0	1	Т	Т	3
1	0	1	4	4	–1
2	0	1	4	3	–1
3	0	1	4	3	–1

Случай неотрицательных весов – алгоритм Дейкстры

Известны более эффективные алгоритмы для двух важных случаев, а именно: когда веса всех дуг неотрицательны или когда граф бесконтурный. Сначала опишем алгоритм для первого случая – алгоритм Дейкстры.

Алгоритм нахождения расстояния от источника до всех остальных вершин в графе с неотрицательными весами дуг – метод Дейкстры

Данные: Ориентированный граф $\langle V, E\rangle$ с выделенным источником $s \in V$, матрица весов дуг $A[u, v]$, $u, v \in V$ (все веса неотрицательны).

***Результаты*:** Расстояния от источника до всех вершин графа $D[v] = d(s, v)$, $v \in V$.

1 begin
2 for $v \in V$ **do** $D[v] := A[s, v]$; $D[s] := 0$;
3 $T := V \setminus \{s\}$;
4 while T γ $\varnothing$ **do**
5 begin
6 u := произвольная вершина $r \in T$, такая что $D[r] = \min(D[p]: p \in T\}$;
7 $T := T \setminus \{u\}$;
8 for $v \in T$ **do** $D[v] := \min(D[v], D[u] + A[u, v])$
9 end
10 end.

Чтобы понять действие алгоритма, покажем, что следующее условие является *инвариантом* цикла 4: для каждой $v \in V \setminus T$ $D[v] = d(s, v)$, для каждой $v \in T$ $D[v]$ = длине кратчайшего из тех путей из s в v, для которых предпоследняя вершина принадлежит множеству $V \setminus T$. (4).

На самом деле, в строке 6 мы находим вершину $u \in T$ такую, что значение $D[u]$ является минимальным (из всех) значением $D[t]$, для $t \in T$.

Покажем, что $D[u] = d(s, u)$. Это именно так, потому что если кратчайший путь из s в u имеет длину меньше $D[u]$, то в силу второй части условия (4) его предпоследняя вершина принадлежит множеству T.

Пусть t будет первой вершиной пути, принадлежащей множеству T. Начальный отрезок пути из s в t составляет кратчайший путь из s в t, причем его предпоследняя вершина не принадлежит множеству T. По второй части условия (4) имеем $D[t] = d(s, t)$. Используя предположение о неотрицательности весов, получаем

$$D[t] = d(s, t) \div d(s, u) < D[u]$$

вопреки принципу, по которому была выбрана вершина u.

Таким образом, $D[u] = d(s, u)$ и мы можем в строке 7 удалить u из множества T, не нарушая первой части условия (4). Чтобы обеспечить выполнение также и второй части этого условия, следует еще проверить пути из из s в $v \in T$, предпоследняя вершина в которых есть u, и выполнить актуализацию переменных $D\,[v]$, $v \in T$. Именно это выполняет цикл 8.

Очевидно, что условие (4) выполняется при входе в цикл 4. По окончании действия алгоритма $T = \varnothing$, а следовательно, согласно условию (4), $D[v] = d(s, v)$, $v \in V$.

Оценим сложность алгоритма Дейкстры. Цикл 4 выполняется $n - 1$ раз, причем каждое его выполнение требует $O(n)$ шагов: $O(n)$ шагов для нахождения вершины u в строке 6 (предполагаем, что множество T представлено списком) и $O(n)$ шагов для выполнения цикла 8. Таким образом, сложность алгоритма есть $O(n^2)$.

Тщательно подбирая структуры данных, можно получить вариант алгоритма со сложностью $O(m \log n)$. Для этого множество T нужно представить бинарным деревом с высотой $O(\log n)$ и с таким свойством, что для произвольных его вершин u и v:

если u – сын v, то $D[u] \div D[v]$.

Вершина u, для которой значение $D[u]$ минимально, является тогда корнем дерева. Этот корень можно устранить за $O(\log n)$ шагов, сохраняя свойство уменьшения значения $D[j]$ на каждом пути до корня. Достаточно сместить на место корня его сына s с большим (или равным) значением $D[j]$, затем на освободившееся место передвинуть сына вершины s с большим значением $D[j]$ и т.д. Если граф представлен списками *ЗАПИСЬ*$[u]$, $u \in V$, то строку 8 можно заменить на

for $v \in$ *ЗАПИСЬ* $[u]$ **do**
if $D[u] + A\,[u, v] < D[v]$ **then**
begin
$D[v] := D[u] + A\,[u, v]$;

передвинуть вершину в «дереве» в направлении корня так, чтобы сохранить условие если u – сын v, то $D[u] \div D[v]$

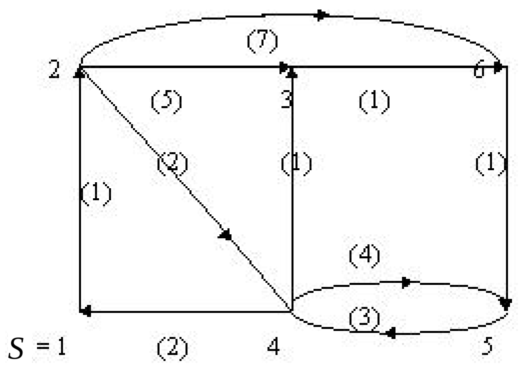
end.

Если предположить существование таблицы указателей на вершины нашего «дерева», то передвижение вершины v, о которой идет речь в данной части раздела, может быть осуществлено за O(log n) шагов. Достаточно заменять v поочередно вершинами, находящимися непосредственно над ней.

В алгоритме, модифицированном таким способом, каждая дуга графа анализируется в точности один раз, причем с этим связано O(log n) шагов на передвижение соответствующей вершины в дереве, представляющем множество T. Это дает в сумме O(m log n) шагов. Сюда нужно добавить O(n log n) шагов, необходимых для построения нашего дерева и для устранения n – 1 раз из него корня. Общая сложность алгоритма есть O(m log n).

Неизвестно, существует ли алгоритм сложности O(m) нахождения расстояния от фиксированной вершины до всех остальных вершин графа с неотрицательными весами дуг. Можно показать, что существует константа C, такая что эта задача для произвольного $k > 0$ может быть решена за время $Ck(m + n^{1+1/k})$.

Работа алгоритма Дейкстры проиллюстрирована на рисунке ($V = \{1,..., 6\}$, веса дуг даны в скобках, значения $D[v]$, $v \in T$, приведены со звездочкой (*), минимальные значения – с двумя звездочками.



D [1]	D[2]	D[3]	D[4]	D[5]	D[6]
0	1*	T *	T *	T *	T *
0	1	6*	3*	T *	8*
0	1	4**	3	7*	8*
0	1	4	3	7*	5**
0	1	4	3	6**	5

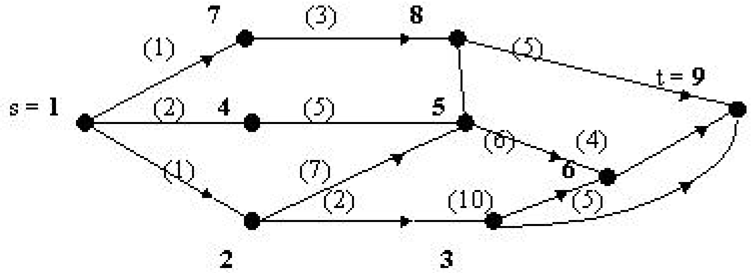
** = *min*

Пути в бесконтурном графе

Второй случай, для которого известен алгоритм нахождения расстояний от фиксированной вершины за время $O(n^2)$ – случай, когда граф является бесконтурным (веса дуг могут быть произвольными). Сначала докажем следующую лемму.

Лемма. *В произвольном бесконтурном графе вершины можно перенумеровать так, что каждая дуга будет иметь вид* $\langle v_i, v_j \rangle$, *где* $i < j$.

Пример такой нумерации приведен на рисунке.



Для доказательства предложим алгоритм, который конструирует такую нумерацию.

Алгоритм нумерации вершин бесконтурного графа

Данные: Ориентированный бесконтурный граф $<V, E>$, определяемый списками инцидентности *ЗАПИСЬ* [v], $v \in V$.

Результаты: Для каждой вершины $v \in V$ номер $NR[v]$, такой, что для произвольной дуги $\langle u, v\rangle \in E$ выполняется неравенство $NR[u] < NR[v]$.

begin
for $v \in V$ **do** *ЧЗАХ* [v] := 0;
{*ЧЗАХ* [v] = число дуг, заходящих в v }
for $u \in V$ **do**
for $v \in$ *ЗАПИСЬ* [u] **do** *ЧЗАХ* [v] := *ЧЗАХ* [v] + 1;
СТЕК := $\varnothing$;
for $v \in V$ **do**
if *ЧЗАХ* [v]= 0 **then** *СТЕК*$\Leftarrow v$;
num := 0;
while *СТЕК* γ$\varnothing$ **do**
begin $u \Leftarrow$ *СТЕК*;
num := *num* + 1; $NR[u]$:= *num*;
for $v \in$ *ЗАПИСЬ* [u] **do**
begin *ЧЗАХ* [v]:= *ЧЗАХ* [v] -1;
if *ЧЗАХ* [v]= 0 **then** *СТЕК*$\Leftarrow v$;
end
end
end.

Алгоритм основывается на следующем простом факте: в произвольном (непустом) бесконтурном графе существует вершина, в которую не заходит ни одна дуга. Чтобы убедиться в этом, выберем произвольную вершину w_1 графа, затем некоторую вершину w_2, такую что $w_{2\rho}\ w_1$, затем вершину w_3, такую что $w_{3\rho)}\ w_2$, и т.д. Через конечное число шагов мы должны дойти до некоторой вершины w_i, в которую не заходит ни одна дуга, ибо в силу бесконтурности ни одна вершина не может повторяться в последовательности w_1, w_2, w_3 , ...

В нашем алгоритме вершины, в которые не заходит ни одна дуга, накапливаются в стеке. В строке 10 выбирается верхний элемент стека u (это мог бы быть произвольный элемент стека), и этой вершине присваивается самый маленький из еще неиспользованных номеров. Таким образом, мы гарантируем то, что все дуги, выходящие из этой вершины, ведут к вершине с большими номерами. Затем вершина u вместе с выходящими из нее дугами удаляется из графа. Это приводит к уменьшению на единицу числа дуг, заходящих в каждую вершину v, такую что $u(\rho)\ v$; это число запоминается в *ЧЗАХ* [v]. Если для какой-нибудь из вершин v это число сводится к нулю, то v помещается в стек.

В силу бесконтурности графа и наших предыдущих замечаний полное опустошение стека, вызывающее окончание выполнения алгоритма (см. цикл 9), наступает не раньше, чем после присвоения номеров всем вершинам графа.

Легко заметить, что каждая дуга анализируется алгоритмом один раз в строке 4 и один раз в строке 12. Таким образом, сложность алгоритма есть $O(m)$ (остается в силе предположение $m = \Omega\ (n)$, в противном случае следовало бы написать $O(m + n)$).

Примечание:

При сравнении скорости роста двух функций $f(n)$ и $g(n)$ (с неотрицательными значениями) мы, как обычно, используем следующие обозначения:

$f(n) = O(g(n)) \Leftrightarrow$ существуют константы C, $N > 0$, такие что $f(n) \div C \cdot g(n)$ для всех n ♦ N

$f(n) = \Omega\ (g(n)) \Leftrightarrow$ существуют константы C, $N > 0$, такие что $f(n)$ ♦ $C \cdot g(n)$ для всех n ♦ N.

Т. е., в частности, $f(n) = \Omega\ (g(n))$ тогда и только тогда, когда $g(n) = O(f(n))$.

Согласно вышеизложенному алгоритму при описании алгоритма нахождения путей в бесконтурном графе мы можем предположить, что каждая дуга идет из вершины с меньшим номером в вершину с большим номером.

Алгоритм нахождения расстояний от источника до всех вершин в бесконтурном графе

Данные: Ориентированный граф $<V, E>$, где $V = \{v_1, ..., v_n\}$, и для произвольной дуги $\langle v_i, v_j\rangle \in E$ имеем $i < j$. Этот граф определен списками инцидентности ПРЕДШ[v], $v \in V$.

Результаты: Расстояния от v_1 до всех вершин графа:

$$D[v_i] = d(v_1, v_i),\ i = 1, ..., n.$$

begin
$D[v_1]$:= 0;
for j := 2 **to** n **do** $D[v_j]$:= T ;
for j := 2 **to** n **do**

for $v_i \in$ *ПРЕДШ* $[v_j]$ **do** $D[v_j] := \min(D[v_j]), D[v_i] + A[v_i, v_j])$

end

Нетрудно доказать индукцией по j, что после выполнения цикла 4 для некоторого значения j выполняется равенство $D[v_i] = d(v_1, v_i)$, $i = 1, ..., j$. Достаточно воспользоваться тем фактом, что все промежуточные вершины кратчайшего пути из v_1 в v_i имеют номера меньше j.

Сложность алгоритма порядка $O(m)$, так как каждая дуга $\langle v_i, v_j\rangle$ анализируется а строке 5 в точности один раз.

Описанные алгоритмы находят применения, в частности, в методах управления выполнением некоторого проекта. Эти методы основываются на построении графа, дуги которого соответствуют некоторым элементарным задачам, составляющим проект, а их веса указывают на время, необходимое для решения отдельных задач.

Кроме этого, мы предполагаем, что для произвольных дуг этого графа $\langle u, v\rangle$ и $\langle v, t\rangle$ задача, изображаемая дугой $\langle u, v\rangle$, должна быть закончена перед началом решения задачи, изображаемой дугой $\langle v, t\rangle$.

Легко заметить, что такой граф должен быть бесконтурным. Нашей задачей является нахождение самого длинного пути из вершины s, соответствующей началу проекта, до вершины t, соответствующей его окончанию. Такой путь называется *критическим путем*. Его длина определяет время, необходимое для реализации всего проекта. Очевидно, что задача сводится к задаче о кратчайшем пути путем изменения знака каждого веса $a(u, v)$, где $u\ (\rho)\ v$, на обратный.

Кратчайшие пути между всеми парами вершин, транзитивное замыкание отношения

Очевидно, что задачу определения расстояния между всеми парами вершин можно решить, используя n раз (поочередно для каждой вершины) один из ранее изложенных методов нахождения расстояний от фиксированной вершины. Таким образом, мы получаем алгоритм со сложностью $O(n^4)$ (при использовании метода Форда – Беллмана) или $O(n^3)$ для бесконтурных графов или неотрицательных весов. Однако оказывается, что в общем случае n-кратное использование метода Форда – Беллмана не является наилучшим методом. Рассмотрим два более эффективных метода.

Для этого рассмотрим ориентированный граф $G = \langle V, E\rangle$, где $V = \{v_1, ..., v_n\}$, и предположим, что $A = [a_{ij}]$ есть матрица весов ($a_{ij} = a(v_i, v_j)$). Обозначив через $d_{ij}^{(m)}$ длину кратчайшего пути из v_i в v_j, содержащего не более m дуг, получаем следующие очевидные уравнения:

$$d_{ij}^{(0)} = \begin{cases} 0, & i = j, \\ \infty, & i \neq j \end{cases} \tag{1}$$

$$d_{ij}^{(m+1)} = \min\{d_{ik}^{(m)} + a_{kj} : 1<k<n\} \tag{2}$$

Отметим, что последнее уравнение обнаруживает некоторое сходство с определением произведения двух квадратных матриц. Если операцию min трактовать как «сумму», операцию «+» – как «произведение», то матрица $[d_{ij}^{(m+1)}]$ является «произведением» матриц $[d_{ij}^{(m)}]$ и $A = [a_{ij}]$. Обозначим такое «произведение» двух матриц A и B через $A*B$ и отметим, что для этой операции единичным элементом служит матрица

$$U = \begin{matrix} 0 & \infty & \infty & L & \infty \\ \infty & 0 & \infty & L & \infty \\ \infty & \infty & 0 & L & \infty \\ M & M & M & & M \\ \infty & \infty & \infty & L & 0 \end{matrix}$$

Из уравнений (1) и (2) теперь легко следует, что $[d_{ij}^{(0)} = U]$ и

$$d_{ij}^{(m)} = ((...((\mathrm{A}*\mathrm{A})*\mathrm{A})...)*\mathrm{A})\ (m\blacklozenge\ 1). \tag{3}$$

Возможен один из следующих случаев:

(1) $d_{ij}^{(n\text{-}1)} = d_{ij}^{(n)}$ и в результате $d_{ij}^{(m)} = d_{ij}^{(n\text{-}1)}$ для каждого $m \blacklozenge n$. Тогда очевидно $d_{ij}^{(n\text{-}1)} = d(v_i, v_j)$.

(2) $d_{ij}^{(n\text{-}1)}\ \gamma\ d_{ij}^{(n)}$. Это означает, что существует контур отрицательной длины.

Произведение $A*B$ двух матриц размерности $n\ \beta\ n$ можно вычислить за время $\mathrm{O}(n^3)$ (n сложений и n – 1 сравнений на каждый из n^2 элементов произведения $A*B$). Следовательно, матрицу $[\mathrm{d}_{ij}^{(n\text{-}1)}]$ и тем самым расстояние между всеми парами вершин можно вычислить за время $\mathrm{O}(n^4)$.

Пока сложность этого алгоритма такая же, как и для случая n-кратного использования алгоритма Форда–Беллмана. Однако мы можем ее снизить, если заметим, что операция * ассоциативна (т.е. $(A * B) * C = A * (B * C)$). Этот факт позволяет вычислять произведение (3), поочередно возводя матрицу A в квадрат и тем самым заменяя (n – 1)– количества умножений матрицы количеством ($\log\ n$) – умножений. Таким образом, мы получаем алгоритм сложности $\mathrm{O}(n^3 \log n)$, отыскивающий расстояния между всеми парами вершин в графе без контуров отрицательной длины.

Авторами еще более эффективного алгоритма являются Уоршалл и Флойд. Идея этого алгоритма следующая. Обозначим через $d_{ij}^{(m)}$ длину кратчайшего из путей из v_i в v_j с промежуточными вершинами в множестве $\{v_1, ..., v_m\}$.

Тогда имеем следующие уравнения:

$$d_{ij}^{(0)} = a_{ij}, \tag{4}$$

$$d_{ij}^{(m+1)} = \min(d_{ij}^{(m)}, d_{im}^{(m)} + d_{mj}^{(m)}). \tag{5}$$

Обоснование второго уравнения достаточно простое. Рассмотрим кратчайший путь из v_i в v_j с промежуточными вершинами из множества $\{v_1, ..., v_m, v_m+1\}$. Если этот путь не содержит v_m+1, то $d_{ij}^{(m+1)} = d_{ij}^{(m)}$

Если же он содержит v_m+1, то деля путь на отрезки от v_i до v_m+1 и от v_m+1 до v_j, получаем равенство $d_{ij}^{(m+1)} = d_{im}^{(m)} + d_{mj}^{(m)}$.

Уравнения (4) и (5) дают возможность легко вычислить расстояния $d(v_i, v_j) = d_{ij}^{(n)}$, $1 \div i, j \div n$.

Алгоритм вычисления расстояний между всеми парами вершин – метод Флойда

Данные: Матрица весов дуг $A[i, j]$, $1 \div i, j \div n$, ориентированного графа без контуров отрицательной длины.

Результаты: Расстояния между всеми парами вершин $D[i, j] = d(v_i, v_j)$.

begin
for *i* := 1 **to** *n* **do**
for *j* := 1 **to** *n* **do** *D*[*i*, *j*] := *A*[*i*, *j*];
for *i* := 1 **to** *n* **do** *D*[*i*, *i*] := 0;
for *m* := 1 **to** *n* **do**
for *i* := 1 **to** *n* **do**
for *j* := 1 **to** *n* **do**
D[*i*, *j*] := min(*D*[*i*, *j*], *D*[*i*, *m*] + *D*[*m*, *j*];
end

Очевидно, что сложность этого алгоритма есть $O(n^3)$. Отметим, что такую же сложность имел алгоритм Форда – Беллмана нахождения расстояний от фиксированной вершины до всех остальных вершин. Любопытно, что для общего случая (т.е. без предположения о неотрицательности весов либо о бесконтурности графа) не известен ни один алгоритм нахождения расстояния между одной фиксированной парой вершин, который был бы значительно эффективнее алгоритма нахождения расстояний между всеми парами вершин.

С задачей определения кратчайших путей в графе тесно связана задача транзитивного замыкания бинарного отношения. Вспомним, что под бинарным отношением на множестве V мы понимаем произвольное подмножество $E \subseteq V \beta V$. Такое отношение является транзитивным, если выполняется условие если $\langle x, y \rangle \in E$ и $\langle y, z \rangle \in E$, то $\langle x, z \rangle \in E$ для произвольных $x, y, z \in E$. Заметим, что бинарное отношение $E \subseteq V \beta V$ можно однозначно представить ориентированным графом $G = \langle V, E \rangle$. Для произвольного такого отношения мы определяем

$E^* = \{\langle x, y \rangle$: в $\langle V, E \rangle$ существует путь ненулевой длины из x в $y\}$.

Нетрудно заметить, что E^* – транзитивное отношение на множестве V и $E \subseteq E^*$. Более того, E^* является наименьшим транзитивным отношением, содержащим E, т.е. для произвольного транзитивного отношения $F \supseteq E$ выполняется включение $E^* \subseteq F$. Отношение E^* называется *транзитивным замыканием* отношения E.

Если отношение E представить в виде графа $\langle V, E \rangle$, в котором каждая дуга имеет вес 1, то транзитивное замыкание E^* можно вычислить с помощью алгоритма Флойда за время $O(n^3)$; после завершения работы имеем

$$< v_i, v_{j>} \in E^* \Leftrightarrow D[i, j] < \mathrm{T}\ .$$

При вычислении транзитивного замыкания удобно принять

$$A[i, j] = \begin{cases} 0, & \langle v_i, v_j \rangle \notin E, \\ 1, & \langle v_i,\ v_j \rangle \in E \end{cases} \tag{6}$$

Тогда строку 8 в алгоритме можно заменить на

$D[i, j]{:=} D[i, j] \vee (D[i, m] \wedge D[m, j]\ )$,

где $\vee$ и $\wedge$ – обычные булевы операции.

После завершения работы алгоритма, модифицированного таким образом (принадлежащего Уоршаллу), очевидно, имеем

$$D[i, j] = \begin{cases} 0, & \langle v_i, v_j \rangle \in E, \\ 1, & \langle v_i,\ v_j \rangle \notin E \end{cases}$$

Известен алгоритм построения транзитивного замыкания, более эффективный, чем алгоритм Уоршалла. Он использует связь этой задачи с умножением матриц, обсуждавшуюся ранее, эта связь для матриц A, определяемых равенством (6), приводит к обычному умножению булевых матриц по формуле

$$c_{ij} = \mathop{\vee}_{k=1}^{n} \left(a_{ik} \wedge b_{jg}\right)$$

Такое умножение можно выполнить за время $O(n^{\log 7})$, что дает алгоритм построения транзитивного замыкания, имеющего сложность $O(n^{\log 7} \log n)$. Такой алгоритм имеет скорее теоретическое значение, поскольку метод умножения матриц за время $O(n^{\log 7})$ является довольно сложным и, следовательно, обнаруживает свое преимущество перед обычным «школьным» методом только при очень больших значениях n. На практике обычно самым эффективным оказывается алгоритм Уоршалла, соответствующим образом запрограммированный.

Другим способом построения транзитивного замыкания отношения E является применение поиска в глубину (или в ширину) в графе $<V, E>$. Этот способ особенно выгоден, когда отношение E симметрично; очевидно, что тогда транзитивное замыкание строится отдельно для каждой связной компоненты, на которые разбивается неориентированный граф, определяемый отношением E, и это построение может быть получено за время $O(m + n)$.

10.5. Знаковые графы и орграфы, их применение

Для начала введем терминологию, а заодно сравним ориентированные и неориентированные графы гносеологически, представив параллельные места билингвой – в виде следующей таблицы. Орграф здесь и далее – ориентированный граф.

ГРАФЫ И ОРИЕНТИРОВАННЫЕ ГРАФЫ (АНАЛОГИИ И ОТЛИЧИЯ)

Пусть $D = (V, A)$ – орграф	Пусть $G = (V, E)$ – граф
Путь **в** D – последовательность вершин и дуг u_1, a_1, u_2, a_2,..., u_t, a_t, u_t+1, где t♦ 0, причем каждая вершина $u_i \in V$, а каждая дуга $a_i \in A$, и a_i всегда яв-ляется дугой (u_i, u_i+1). Путь обычно записывается последовательностью вершин u_1, u_2, ..., u_t, u_t+1	***Цепь*** **в** G – последовательность вершин и ребер u_1, e_1, u_2, e_2,..., u_t, e_t, u_t+1, где t♦ 0, причем каждая вершина $u_i \in V$, а каждое ребро $e_i \in E$, и e_i всегда является ребром (u_i, u_i+1). Цепь обычно записывается последовательностью вершин u_1, u_2, ..., u_t, u_t+1. ***Цепь*** **= маршрут без повторений (каждое ребро проходится лишь один раз)**
Полупуть **в** D – последовательность вершин и дуг u_1, a_1, u_2, a_2,..., u_t, a_t, u_t+1, где t♦ 0, причем каждая вершина $u_i \in V$, а каждая дуга $a_i \in A$, и a_i всегда является либо дугой (u_i, u_i+1), либо дугой (u_i+1, u_i). Полупуть также обычно записывается последовательностью вершин u_1, u_2, ..., u_t, u_t+1	*Аналогия – та же цепь (см. выше)* В определении полуцепи нет смысла – полуцепь всегда совпадает с соответствующей цепью
Полный путь или полный полупуть **в** D – путь или полупуть, проходящий через все вершины D	***Полная цепь или полная полуцепь*** **в** G – цепь или полуцепь, проходящая через все вершины G
Простым путем или простым полупутем **в** D называется путь или полупуть без повторяющихся вершин	***Простой цепью*** **в** G называется цепь без повторяющихся вершин
Замкнутым путем или полупутем **в** D называется путь или полупуть u_1, a_1, u_2, a_2,..., u_t, a_t, u_t+1, в котором u_t+1=u_1	***Замкнутой цепью*** **в** G называется цепь u_1, u_2, ..., u_t, u_t+1, в которой $u_{t+1}=u_1$
Полный замкнутый путь или полный замкнутый полупуть **в** D – полный путь или полный полупуть, который замкнут	***Полная замкнутая цепь*** **в** G – полная цепь, которая замкнута
Контуром **в** D называется замкнутый путь u_1, u_2, ..., u_t, u_1, в котором все вершины различны	***Циклом*** **в** G называется замкнутая цепь u_1, e_1, u_2, e_2,..., u_t, e_t, u_1 , в которой все вершины различны

Полуконтуром **в** D называется замкнутый полупуть, u_1, a_1, u_2, a_2,..., u_t, a_t, u_1, в котором все вершины u_1, u_2, ..., u_t и все дуги a_1, a_2,..., a_t различны	*Аналогия – тот же цикл (см. выше)* *В определении полуцикла нет смысла – полуцикл всегда совпадает с соответствующим циклом*
Вершина $\boldsymbol{u_i}$, ***достижима*** **из вершины** u_j, если имеется путь из u_i в u_j	Вершина $\boldsymbol{u_i}$, ***достижима*** **из вершины** u_j, если имеется соединяющая их цепь
Вершины $\boldsymbol{u_i}$ и u_j ***соединимы***, *если имеется путь из вершины* u_i в вершину u_j или из вершины u_j в вершину u_i	Вершины $\boldsymbol{u_i}$ и u_j ***соединимы***, *если имеется соединяющая их цепь*
Длиной **пути, полупути, простого пути, простого полупути, контура или полуконтура называется число дуг, содержащихся в них**	***Длиной*** **цепи, простой цепи или цикла называется число ребер, содержащихся в них**
Расстояние $\boldsymbol{d}(u_i, u_j)$ от вершины u_i и до вершины u_j в D равно длине кратчайшего пути из u_i в u_j или не определено, если путь из u_i в u_j отсутствует Во взвешенном графе длиной и расстоянием обычно называют Σ с учетом веса (Σ весов)	***Расстояние*** $\boldsymbol{d}(u_i, u_j)$ от вершины u_i и до вершины u_j в G равно длине кратчайшей цепи между u_i и u_j или не определено, если цепь между ними отсутствует

ПОДГРАФЫ, ОРИЕНТИРОВАННЫЕ ПОДГРАФЫ И СВЯЗНОСТЬ (АНАЛОГИИ И ОТЛИЧИЯ)

Пусть D=(V, A) – орграф	Пусть G=(V, E) – граф
Орграф D называется ***сильно связным*** (***сильным, категории связности* 3**), если для каждой пары вершин u_i и u_j в $D u_i$ достижима из u_j и u_i достижима из u_i	Граф G называется ***связным***, если каждая пара вершин u_i и u_j в G *связана цепью*
Орграф D называется ***слабо связным*** (***слабым, категории связности* 1**), если каждая пара вершин u_i и u_j в D соединима (полупутем)	*Аналогия – см. выше*

Орграф D называется ***односторонне связным*** (***односторонним, категории связности* 2**), если для каждой пары вершин u_i и u_jв D либо u_i достижима из u_j, либо u_j достижима из u_i	*Аналогия – см. выше*
***Подграфом* в** D называется орграф (W, B), в котором $W \subseteq V$, $B \subseteq A$	***Подграфом* в** G называется граф (W, F), в котором $W \subseteq V$, $F \subseteq E$
***Порожденным подграфом* в** D называется подграф (W, B), в котором B содержит все дуги из A, соединяющие вершины в W	***Порожденным подграфом* в** G называется подграф (W, F), в котором F содержит все ребра из E, соединяющие вершины в W
***Сильной компонентой* в** D называется максимальный сильно связный (порожденный) подграф	***Компонентой* (*связности*)** в G называется максимальный связный (порожденный) подграф

Взвешенный граф, взвешенный орграф – граф (орграф), каждому ребру которого приписан некоторый вес.

Знаковый граф, знаковый орграф – граф (орграф), каждому ребру которого приписан некоторый знак.

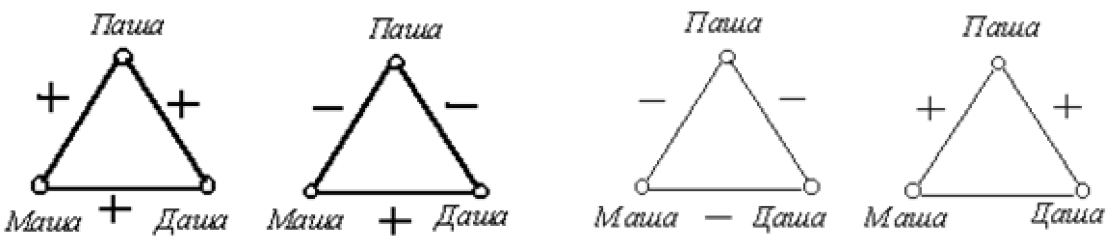
Знак пути, цепи, замкнутого пути, замкнутой цепи, контура, цикла и т.д. определяется как произведение знаков входящих в них дуг или ребер, если знак плюс заменить на +1, *а знак минус на* -1. *Очевидно, что путь, цепь и т.д. имеют знак минус, если число дуг или ребер, содержащихся в них, нечетно, иначе они имеют знак плюс.*

Хейдер изучал задачи из области **социологии малых групп** *людей (Heider F. Attitudes and Cognitive Organization.— J. of Phych.T.,– 21, 1946. – P. 107-112*).

Его результаты - полный обзор вариантов знаковых графов для группы из трех человек в условиях явно выраженной симпатии / антипатии представлены на рисунках.

Группы из трех лиц по Хейдеру психологически

сбалансированные несбалансированные



Анализ этого и огромного количества других примеров из самых разных областей человеческой деятельности привел Картрайта и Харари (*Cartwright D. and Harary F. Structural Balance: A Generalization of Heider's Theory. – Psych. Rev., 63, 1956. – P. 277-293*) к следующей **математической модели баланса**:

Малая группа является сбалансированной, если представляющий ее знаковый граф сбалансирован.

Знаковый граф называется ***сбалансированным****, если каждый цикл в нем положителен.*

Теорема *О структуре* (*теорема Харари о балансе*)

Для знакового графа G=(V,E) следующие утверждения эквивалентны:

a. *Граф G сбалансирован.*

b. *Каждая замкнутая цепь в G положительна.*

c. *Любые две цепи между любыми двумя вершинами u_i и u_j имеют одинаковый знак.*

d. *Множество вершин V можно разбить на два подмножества A и B так, что каждое положительное ребро соединяет вершины одного подмножества и каждое отрицательное соединяет вершины различных подмножеств.*

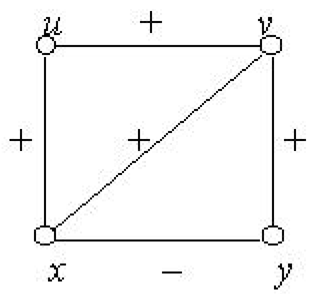
Последнее утверждение называют также ***критерием баланса.***

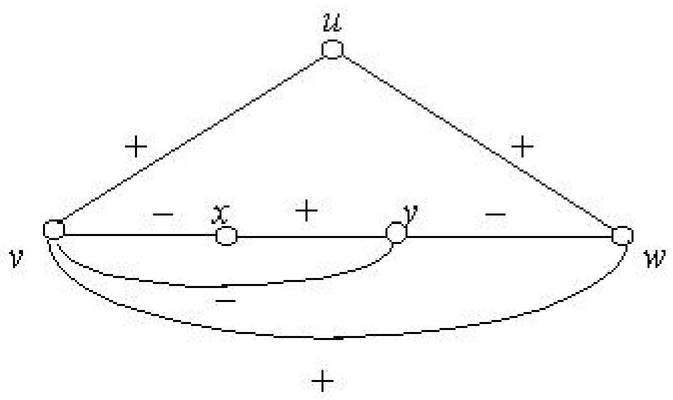
Для пункта *d* существует интерпретация и в области **политики** – устойчивым является представительный орган, основанный на одно- или двухпартийной основе (внутри фракции существуют отношения «симпатии», а соответствующие отношения между представителями разных фракций отрицательны).

Использование ***знакового орграфа*** в качестве модели сложной системы основано на следующем представлении. Наиболее существенные для рассматриваемой проблемы переменные считаются вершинами орграфа. От переменной ***u*** к переменной ***v*** проводится дуга, если изменение ***u*** оказывает непосредственное существенное воздействие на ***v***. И, наконец, эта дуга имеет знак плюс, если воздействие является «*усилением*» (при прочих равных условиях увеличение ***u*** приводит к увеличению ***v*** и уменьшение ***u*** приводит к уменьшению ***v***), и знак минус, если воздействие вызывает «*торможение*» (при прочих равных условиях увеличение ***u*** приводит к уменьшению ***v*** и уменьшение ***u*** приводит к увеличению ***v***).

Контуры в знаковом орграфе соответствуют ***контурам обратной связи***, причем контуры, усиливающие отклонение – контурам ***положительной обратной связи***, а контуры, противодействующие отклонению, – контурам ***отрицательной обратной связи.***

Контур **противодействующий отклонению –** это такой контур, в котором увеличение любой переменной приводит через другие переменные контура к уменьшению данной переменной и наоборот.





Класс A

u

v

w

Класс B

x

y

Контур ***усиливающий отклонение*** – это контур, в котором увеличение (уменьшение) любой переменной приводит к ее последующему увеличению (уменьшению). При этом оказывается, что знак контура замечательным образом совпадает со знаком соответствующей обратной связи!

Таким образом, в особенности, используя компьютер, совсем несложно, проанализировав соответствующий граф, определить, какие вершины-факторы «работают» на устойчивость текущего состояния системы, а какие – на изменение ситуации, то есть, например, на ее развитие или ее деградацию... Важно то, что, произведя соответствующий анализ и увидев, что возможные изменения негативны, можно попытаться ввести в модель новые вершины, новые факторы, которые повысят стабильность системы или, напротив, исключить из модели факторы, которые оказались дестабилизирующими. Очень многие из приведенных в конце статьи примеров разрабатывались именно для подобного анализа.

Оставаясь знаковым, орграф может быть в то же время взвешенным в том смысле, что его вершинам может быть приписан вес. Такие орграфы удобно использовать в моделях, позволяющих сделать некоторые допущения о влиянии изменений значения параметра одной вершины на параметры других вершин – правила изменений параметров вершин. На таких графах можно качественно и количественно оценивать изменения значений параметров вершин $\boldsymbol{V}(t)$, происходящие в системе в дискретные моменты времени под влиянием поступающего *импульса* (изменения) $\boldsymbol{P}(t)$, в том числе, этот импульс может быть внешним, отражающим изменение одного из параметров. Здесь знак дуги оказывается знаком изменения следующего фактора в текущий момент времени, поскольку численное значение веса вершины оказывается равным его предыдущему численному значению плюс изменение с учетом его знака, умноженного на знак дуги.

11. ТЕОРИЯ АВТОМАТОВ

Конечный автомат «в чистом виде» – это математическая модель устройства с конечной памятью, преобразующего дискретную информацию. Конечный автомат является одним из важнейших видов *управляющих систем*. Содержательно конечный автомат можно охарактеризовать как устройство, имеющее входной и выходной каналы и находящееся в каждый из моментов дискретного времени, называемых тактовыми моментами, в одном из состояний. По входному каналу в каждый тактовый момент в устройство поступают сигналы a – буквы входного алфавита A; в те же моменты по выходному каналу устройство выдает сигналы b – буквы выходного алфавита B, причем b определяется состоянием s из алфавита состояний S и буквой a; внутреннее состояние s' в следующий тактовый момент также определяется состоянием s и буквой a из предыдущего момента. Таким образом, для некоторых функций φ и ϕ имеет место

$$b = \varphi\ (a, s),\ s' = \phi\ (a, s).$$

Эти функции называются соответственно *выходной* и *переходной функциями; они определяют закон «переработки» слов в алфавите* A, подаваемых побуквенно на входной канал устройства при условии задания начального состояния устройства.

Для конечных автоматов предполагается конечность алфавитов A, S, B. Если считать указанную «переработку» слов главной характеристикой устройства, то его можно отождествить с набором (A, S, B, φ, ϕ), который и называют конечным автоматом. Для этой формы описания конечного автомата характерно отношение исследователя к устройству как внешнего наблюдателя. Само задание конечного автомата называется при этом *абстрактным* конечным автоматом. В случае, когда устройство рассматривается с учетом того, что оно собрано по некоторым композиционным правилам из абстрактных конечных автоматов, приходят к понятию *структурного* конечного автомата, который в итоге также реализует некоторый абстрактный конечный автомат.

Автомат «вообще» (от греческого $\alpha\upsilon\tau o\mu\alpha\tau o\zeta$ – самодействующий) – *управляющая система*, являющаяся *конечным автоматом* или некоторой его модификацией, полученной путем изменения его компонент или функционирования. Основное понятие – конечный автомат – возникло в середине 20 века в связи с попытками описать на математическом языке функционирование нервных систем, универсальных вычислительных машин и других реальных автоматов. Характерной особенностью такого описания является *дискретность* соответствующих математических моделей и конечность областей значений их параметров, что приводит к понятию конечного автомата.

Наряду с понятием конечного автомата рассматриваются различные его обобщения и модификации, отражающие те или иные особенности реальных устройств. Для конечного автомата (A, S, B, φ, ϕ) существующие модификации можно разбить на следующие три основные группы. К первой группе относятся автоматы, у которых некоторые из алфавитов A (входной), S (состояний) или B (выходной) бесконечны, в связи с чем такие автоматы называются бесконечными. Ко второй группе относятся автоматы, у которых вместо выходной и переходной функций φ и ϕ допускаются произвольные отношения или случайные функции. Таковы частичные, недетерминированные, вероятностные и другие автоматы. К третьей группе относятся автоматы со специфическими множествами входных объектов. Таковы, например, автоматы с переменной структурой. Существуют автоматы, принадлежащие одновременно разным группам. Наряду с этим большую роль играют специальные подклассы конечных автоматов, например, автоматы без памяти. Кроме того, использование понятий и методов из других разделов математики также приводит к появлению специфических классов автоматов и связанных с ними задач. Например, при применении алгебраических средств возникают понятия автоматов над термами, линейного, группового, свободного и других; вопросы теории кодирования порождают понятия самонастраивающихся, обратимых автоматов и другие.

Теория автоматов – это раздел теории *управляющих систем*, изучающий математические модели преобразователей дискретной информации, называемые *автоматами*. С определенной точки зрения такими преобразователями являются как реальные устройства (вычислительные машины, автоматы, живые организмы и т.д.), так и абстрактные системы (например, формальная система, аксиоматические теории и т.д.). Наиболее тесно теория автоматов связана с *теорией алгоритмов*.

Большинство задач теории автоматов – общие для основных видов управляющих систем. К ним относятся задачи анализа и синтеза автоматов, задачи полноты, минимизации, эквивалентных преобразований автоматов и другие. *Задача анализа* состоит в том, чтобы по заданному автомату описать его поведение или по неполным данным об автомате и его функционированию установить те или иные его свойства. *Задача синтеза* автоматов состоит в построении автомата с наперед заданным поведением или функционированием. *Задача полноты* состоит в выяснении, обладает ли множество $M' \subseteq M$

автоматов свойством полноты, т.е. совпадает ли с *M* множество всех автоматов, которые получаются путем конечного числа применений некоторых операций к автоматам из заданного подмножества автоматов *M'*. Задача эквивалентных преобразований в общем виде состоит в том, чтобы найти систему правил преобразований (так называемую полную систему правил) автоматов, которые удовлетворяют определенным условиям и позволяют преобразовать произвольный автомат в любой эквивалентный ему автомат (два автомата эквивалентны, если они имеют одинаковое поведение автомата. Поведение автомата – математическое понятие, описывающее взаимодействие автомата с внешней средой. Примером внешней среды конечного автомата является множество входных слов, а поведением – словарная функция, реализуемая автоматом, или событие, представимое автоматом.)

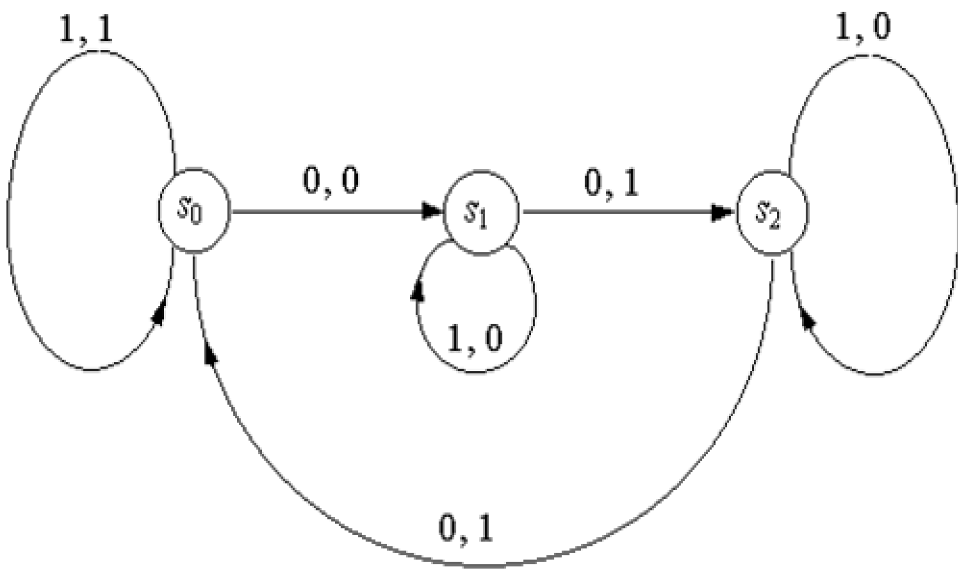
Помимо перечисленных, в теории автоматов имеются специфические проблемы, характерные для автоматов. Так, в зависимости от условий задачи поведение автомата удобно задавать на разных языках, в связи с чем важными задачами являются выбор достаточно удобного адекватного языка и перевод с одного языка на другой. В тесной связи с задачами синтеза и эквивалентных преобразований находится *задача минимизации* числа состояний автомата, а также получение соответствующих оценок. Близкий круг вопросов возникает в связи с моделированием поведения автоматов одного класса автоматами другого класса. Здесь также представляют интерес вопросы минимизации моделирующих автоматов и оценки их сложности. Специальный раздел теории автоматов связан с так *называемыми экспериментами с автоматами* (т.е. способами получения информации о внутренней структуре автоматов по их поведению). Основная задача здесь состоит в том, чтобы получить определенные сведения о строении автомата путем наблюдения его реакции на те или иные внешние воздействия. При этом возникает большой круг задач, связанный с классификацией экспериментов и с вопросами разрешимости задач определенными видами экспериментов, а также с оценками длин минимальных экспериментов, достаточных для решения тех или иных задач. Понятие эксперимента с автоматами используется также в задачах надежности и контроля управляющих систем, в частности контроля автоматов. Многие из перечисленных выше задач могут рассматриваться как алгоритмические проблемы. Для конечных автоматов большинство из них имеют положительное решение.

Теория автоматов находит применение как и в различных областях математики, так и в решении практических задач. Например, средствами теории автоматов доказывается разрешимость некоторых формальных исчислений. Применение методов и понятий теории автоматов к изучению формальных и естественных языков привело к возникновению математической лингвистики (математическая лингвистика – математическая дисциплина, предметом которой является разработка формального аппарата для описания строения естественных и некоторых искусственных языков.) Понятие автомата может служить модельным объектом в самых разнообразных задачах, благодаря чему возможно применение теории автоматов в различных научных и прикладных исследованиях.

Пример 1. Рассмотрим следующий конкретный конечный автомат $M = [A, S, B, \varphi, \phi]$. Входной алфавит $A = \{0, 1\}$; выходной алфавит $B = \{0, 1\}$; три внутренних состояния $S = \{s_0, s_1, s_2\}$; функции выхода и перехода задаются предписаниями

φ : $(s_0, 0)$ a s_1 ϕ: $(s_0, 0)$ a 0
$(s_0, 1)$ a s_0 $(s_0, 1)$ a 1
$(s_1, 0)$ a s_2 $(s_1, 0)$ a 1
$(s_1, 1)$ a s_1 $(s_1, 1)$ a 0
$(s_2, 0)$ a s_0 $(s_2, 0)$ a 1
$(s_2, 1)$ a s_2 $(s_2, 1)$ a 0

Подадим на вход последовательность 0, 1, 0, 1. Если автомат находился сначала в состоянии s_0, то, считав первый символ 0, он перейдет в состояние s_1 и выпечатает 0. Считав затем 1, он останется в состоянии s_1 и выпечатает 0. Считав следующий 0, он перейдет в состояние s_2 и выпечатает 1. Наконец, считав последний символ 1, автомат закончит работу в состоянии s_2, имея на выходной ленте последовательность 0, 0, 1, 0.Таким образом, автомат преобразовал вход 0, 1, 0, 1 (или, короче, 0101) в 0, 0, 1, 0 (или 0010).



Есть два удобных способа описать этот автомат. Прежде всего, можно построить помеченный ориентированный граф, называемый диаграммой состояний.

Вершины этого орграфа помечены символами, обозначающими внутренние состояния. Каждая дуга помечена парой символов a, b, где a – входной символ, вызывающий переход в следующее состояние, отвечающее этому ребру; а b – выходной символ, который автомат выпечатывает.

Второй способ описания – таблица состояний, табличное представление функций φ и ϕ.

Текущее состояние	Следующее состояние			Выход		
	Вход			Вход		
	φ	0	1	ϕ	0	1
s_0		s_1	s_0		0	1
s_1		s_2	s_1		1	0
s_2		s_0	s_2		1	0

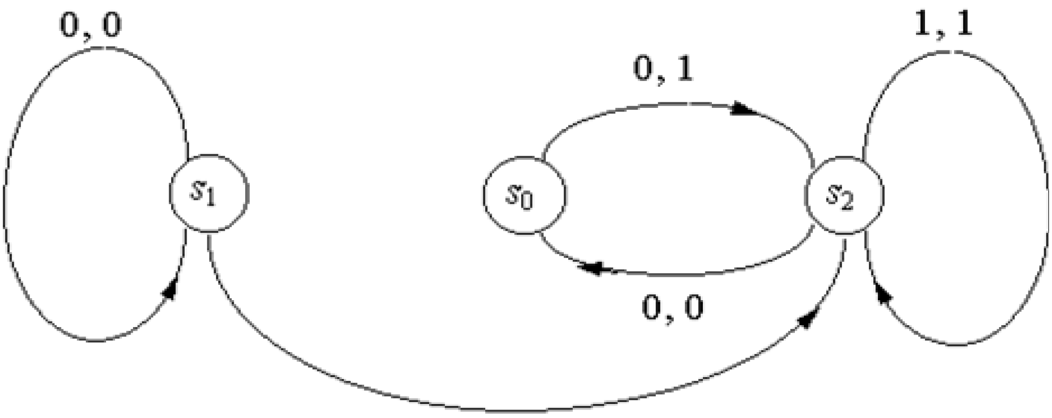
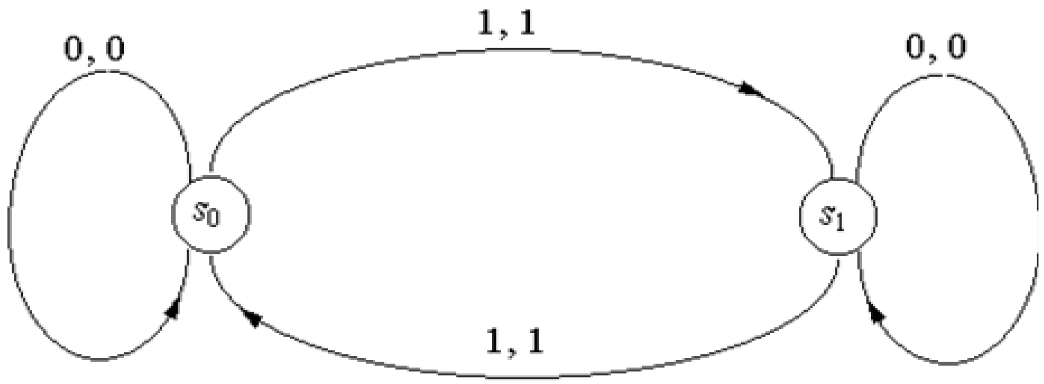


Диаграмма с недостижимыми состояниями

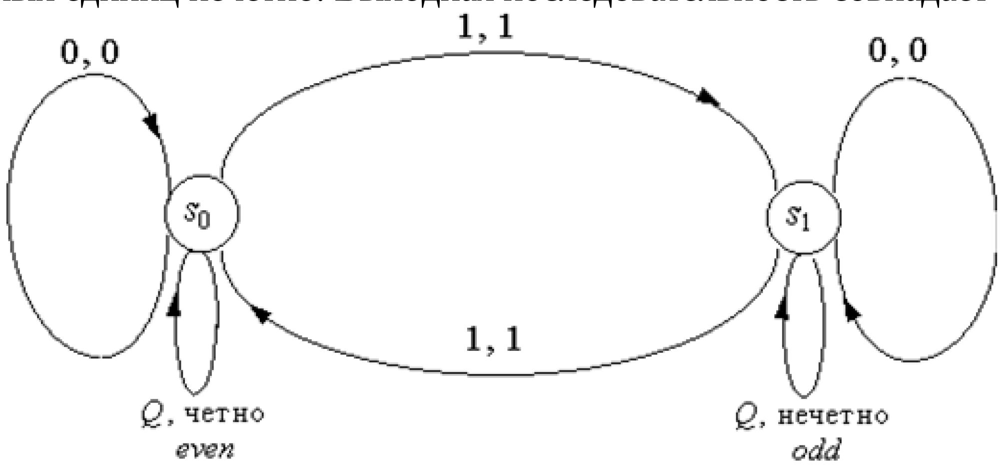
Оба способа имеют свои преимущества и недостатки. Таблица обычно удобнее при вычислениях, диаграмма нагляднее. Например, по диаграмме легче обнаружить состояния, не достижимые из других состояний. На следующем рисунке показана диаграмма состояний автомата, у которого состояние s_1 недостижимо, если автомат начинает работу из состояния s_0 или s_2.



Автомат для проверки четности

Пример 2. Автомат с двумя состояниями, изображенный на следующем рисунке – автомат для проверки четности.

Автомат считывает входную последовательность из нулей и единиц, и его состояние в любой момент времени совпадает с начальным, скажем, s_0, если число считанных к этому моменту единиц четно, и равно s_1, если число считанных единиц нечетно. Выходная последовательность совпадает с входной.



Автомат, выпечатывающий информацию о четности

Пример 3. Автомат, изображенный ниже, проверяет четность и выпечатывает EVEN (четный) или ODD (нечетный) в ответ на запрос, который соответствует входному символу Q.

Считав Q, автомат выпечатывает EVEN, если число ранее считанных единиц было четно, и ODD – если нечетно. Например, входная последовательность 0110Q1110Q будет переработана в 0110 EVEN1110 ODD.

Машины Тьюринга

Исторически понятие конечного автомата развилось из близкого понятия, введенного в 1936 г. логиком Тьюрингом. Тьюринг рассматривал гипотетическую «машину», имеющую конечное множество S внутренних *состояний* и одну бесконечно длинную ленту, разделенную на ячейки, которую машина могла передвигать на одну ячейку вправо или влево за такт. В каждой ячейке машина может записывать символ из конечного алфавита *A*. Первоначально лента должна быть пустой, кроме конечного числа ячеек заполненных заранее. (Эти заранее заполненные ячейки можно представлять себе как программу запуска машины.)

Основная разница между машиной Тьюринга и конечным автоматом состоит в том, что:

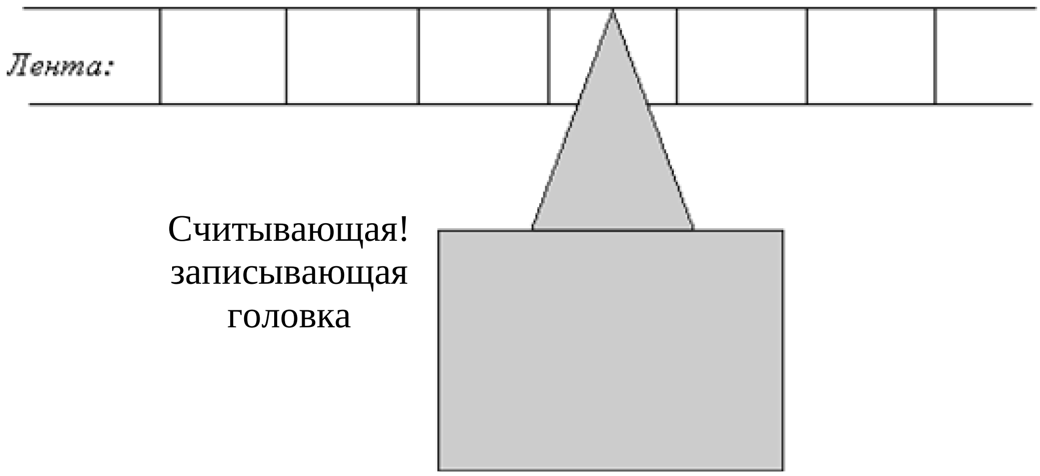
1) лента машины Тьюринга бесконечна;
2) машина Тьюринга может двигаться вдоль ленты (или смещать ленту) в любом направлении.

Это придает машине бесконечную память, которую можно использовать в ходе вычислений. Сверх того, каждую ячейку можно просматривать многократно. Формальное определение машины Тьюринга существует в нескольких вариантах. Вот один из них.

Определение. Машиной Тьюринга называется пятерка объектов $[A, S, \nu, \zeta, \delta]$ следующего типа. *A* есть конечный алфавит символов, которые могут быть записаны в ячейках и являются одновременно входными и выходными: $A = \{a_0, a_1, \ldots, a_n\}$. *S* есть конечное множество внутренних состояний $S = \{s_0, s_1, \ldots, s_r\}$; ν – функция из S β A в S; ζ – функция из S β A в A; δ – функция из S β A в {П, Л, ОСТАНОВ}, интуитивный смысл которого станет ясен из дальнейшего.

Машина Тьюринга работает следующим образом. Она начинает работу, находясь в начальном состоянии s_0. После считывания первого символа она переходит в новое внутреннее состояние, определяемое функцией ν, записывает в ячейке символ, являющийся значением функции ζ, перемещает ленту направо (П), налево (Л), или остается на месте и прекращает работу (ОСТАНОВ) в зависимости от значения функции δ.

На рисунке схематически изображена лента машины Тьюринга и считывающая-записывающая головка.



Таким образом, работа машины состоит в повторении следующего цикла: считывание символа из ячейки, впечатывание нового символа, выбор которого определяет функция ζ, в эту ячейку (может оказаться, что это тот же символ), сдвиг ленты направо или налево либо остановка. Лента бесконечна в обоих направлениях, однако вначале (и, значит, после любого такта) заполнено лишь конечное число ячеек.

Пример. Машина Тьюринга, описанная ниже, считывает входную последовательность нулей и единиц, выпечатывает Ч, если число единиц четное, и Н, если нечетно. Строке из нулей и единиц предшествуют и последуют пустые ячейки, обозначаемые #. Символы Ч или Н печатаются в первой пустой ячейке вслед за входной строкой. *Алфавит*, таким образом, имеет вид

$$A = \{\#, 0, 1, Ч, Н\}.$$

Внутренние состояния: $S = \{s_0, s_1, s_2\}$; s_0 – начальное состояние. Машина останавливается по сигналу ОСТАНОВ.

ν : $(s_0, 0)$ a s_1 ζ: $(s_0, 0)$ a 0 δ : $(s_0, 0)$ a Л
$(s_0, 1)$ a s_2 $(s_0, 1)$ a 1 $(s_0, 1)$ a Л
$(s_1, 0)$ a s_1 $(s_1, 0)$ a 0 $(s_1, 0)$ a Л
$(s_1, 1)$ a s_2 $(s_1, 1)$ a 1 $(s_1, 1)$ a Л
$(s_2, 0)$ a s_2 $(s_2, 0)$ a 0 $(s_2, 0)$ a Л
$(s_2, 1)$ a s_1 $(s_2, 1)$ a 1 $(s_2, 1)$ a Л
$(s_0,\#)$ a s_0 $(s_0, \#)$ a # $(s_0, \#)$ a Л
$(s_1,\#)$ a s_1 $(s_1, \#)$ a Ч $(s_1, \#)$ a ОСТАНОВ
$(s_2,\#)$ a s_2 $(s_2, \#)$ a Н $(s_2, \#)$ a ОСТАНОВ.

Удобнее задавать функции ν, ζ, δ, пользуясь обозначениями Тьюринга. В этом варианте машина Тьюринга задается конечным множеством пятерок $[s_i, a_j, s_r, z_l, t_n]$. В каждой такой пятерке

s_i – текущее состояние машины;
a_j – символ, считываемый из ячейки;
s_r – следующее состояние машины, $s_r = \nu\,(s_i, a_j)$;
z_l – символ, печатаемый в ячейке, $z_l = \zeta\,(s_i, a_j)$;
t_n – одна из команд П, Л, ОСТАНОВ.

В этих обозначениях описанная выше машина задается так:

s_0	#	s_0	#	Л
s_0	0	s_1	0	Л
s_0	1	s_2	1	Л
s_1	0	s_1	0	Л
s_1	1	s_2	1	Л
s_2	0	s_2	0	Л
s_2	1	s_1	1	Л
s_1	#	s_1	Ч	ОСТАНОВ
s_2	#	s_2	Н	ОСТАНОВ

Теорема. Пусть M = [A, S, Z, ν, ζ] – некоторый конечный автомат. Положим

$$\overline{A} = A \cup Z \cup \{\Lambda\}$$

(Λ – символ пустой ячейки) и

$$\overline{\nu}(s_i, a_k) = \nu(s_i, a_k)\overline{\nu}(s_i\Lambda) = s_i$$

$$\overline{\varsigma}(s_i, a_k) = \varsigma(s_i, a_k)\overline{\varsigma}(s_i, \Lambda) = \Lambda$$

$$\delta(s_i, a_k) = \Pi\delta(s_i, \Lambda) = \text{ОСТАНОВ}$$

для всех $(s_i, a_k) \in S \times A$. Тогда машина Тьюринга $T = \left[\overline{A}, S, \overline{\nu}, \overline{\varsigma}, \delta\right]$ ставит в соответствие входным строкам те же выходные строки, что и *M*.

12. ТЕОРИЯ ФОРМАЛЬНЫХ ГРАММАТИК

Теория формальных грамматик – раздел дискретной математики, изучающий способы описания закономерностей, характеризующих всю совокупность правильных текстов того или иного языка. *Формальные грамматики* – это абстрактные системы, позволяющие с помощью единообразных процедур получать правильные тексты данного языка вместе с описанием их структуры. Теория формальных грамматик занимает центральное место в математической лингвистике, так как именно она позволяет моделировать наиболее существенный аспект функционирования языка – переработку смыслов в тексты и обратно. Вместе с тем она выделяется среди других разделов математической лингвистики большей сложностью математического аппарата (сходного с аппаратом теории алгоритмов и общей теории автоматов) и возникающих в ней математических задач. Формальные грамматики наиболее разработанных типов представляют собой системы («устройства»), которые позволяют порождать или распознавать множества конечных последовательностей (цепочек), интерпретируемые обычно как множества правильных предложений, а также сопоставлять входящим в эти множества цепочкам описания их синтаксической структуры в терминах систем составляющих или деревьев подчинения.

Порождающая грамматика **состоит из четырех компонент: Г = (*V*, *W*,** *J*, *R*), где *V* и *W* – непересекающиеся конечные множества, называющиеся *основным* и *вспомогательным алфавитами (или словарями); элементы этих множеств называются соответственно основными (или терминальными) и вспомогательными (или нетерминальными) символами; J* – выделенный вспомогательный символ, называемый *начальным символом*; *R* – конечный набор *правил вывода*, имеющих вид φ ρ ψ, где φ и ψ – цепочки, состоящие из основных и вспомогательных символов.

Основные символы интерпретируются как слова языка, вспомогательные – как имена классов слов и словосочетаний, начальный символ – как символ предложения (т.е. имя класса словосочетаний, являющихся предложениями). Правила вывода описывают связи между частями предложения. Применение правила φ (ρ) ψ к цепочке, имеющей вид α φ β, означает преобразование ее в цепочку αψ β (здесь α и β – цепочки, одна из которых или обе могут быть пустыми). *Вывод* в порождающей грамматике есть конечная последовательность цепочек, в которой каждая следующая цепочка получается из предыдущей использованием каких-либо правил вывода. Последняя цепочка вывода называется выводимой из *первой*. Цепочка основных символов, выводимая из начального символа, называется *предложением*, а множество всех предложений - *языком*, *порожденным* данной грамматикой.

Пример. В грамматике с основным алфавитом {*a*, *b*, *c*}, вспомогательным алфавитом {*a*, *b*}, начальным символом *a* и набором правил {*a*(ρ) *aab*, *a*(ρ) *bc*, *b* (ρ) *b*} одним из выводов будет последовательность *a*, *aab*, *abcb*, *abcb*, *abcb*; цепочка *abcb* – предложение.

В зависимости от вида правил выделяется ряд основных классов порождающих грамматик. Наиболее важны для лингвистических приложений грамматики составляющих (грамматики непосредственно составляющих – НС-грамматики, контекстные грамматики), правила вывода которых имеют вид χ $a\omega$(ρ) χ ψω , где *a* – вспомогательный символ, ψ – непустая цепочка, χ и ω – произвольные цепочки. Важнейший подкласс грамматик составляющих – бесконтекстные грамматики (контекстно-свободные грамматики – КС-грамматики), у которых правила имеют вид *a* (ρ) ψ . Частный случай бесконтекстной грамматики – автоматная грамматика(грамматика с конечным числом состояний) с правилами вывода *a*(ρ) *ab* и *a*(ρ) *a*, где *a* – основной символ, *a* и *b* – вспомогательные символы. Языки, порождаемые грамматиками перечисленных классов, называются соответственно НС-языками (контекстными языками) и КС-языками (бесконтекстными, автоматными).

В *грамматиках составляющих* на каждом шаге вывода заменяется только один символ, поэтому в них с каждым выводом ассоциируется так называемое дерево вывода. Корень дерева отвечает начальному символу. Каждому символу цепочки, на которую заменяется начальный символ на первом шаге вывода, ставится в соответствие узел дерева, и к нему проводится дуга из корня. Для тех из полученных узлов, которые помечены вспомогательными символами, делается аналогичное построение и т.д. Дерево вывода, рассматриваемое как дерево составляющих предложения, задает на нем систему составляющих. Это делает грамматики составляющих хорошим инструментом для описания естественных и искусственных языков.

Еще один важный класс формальных грамматик – *доминационые грамматики*, позволяющие порождать цепочки, одновременно строя для них деревья подчинения.

Для описания естественных языков наиболее адекватны *трансформационные грамматики*, в которых преобразуются не цепочки, а непосредственно синтаксические структуры – «деревья» составляющих или деревья подчинения.

***Теория алгоритмов* – раздел математики, изучающий общие свойства алгоритмов.**

***Алгоритм*, алгорифм** – точное предписание, которое задает вычислительный процесс (называемый в этом случае алгоритмическим), начинающийся с некоторого *исходного данного* (из некоторой совокупности возможных для данного алгоритма исходных данных) и направленный на получение полностью определяемого этим исходным алгоритмом *результата*.

«Уточнение» понятия алгоритма. Как правило, для данного алгоритма можно выделить семь характеризующих его параметров:

1) совокупность возможных исходных данных;
2) совокупность возможных результатов;
3) совокупность возможных промежуточных результатов;
4) правило начала;
5) правило непосредственной переработки;
6) правило окончания;
7) правило извлечения результата.

Понятие алгоритма в общем его виде принадлежит к числу основных первоначальных понятий математики, не допускающих определения в терминах более простых понятий. Возможные уточнения понятия алгоритма приводят, строго говоря, к известному сужению этого понятия. Каждое такое уточнение состоит в том, что для каждого из указанных семи параметров алгоритма точно описывается некоторый класс, в пределах которого этот параметр может меняться. Выбор таких классов и отличает одно уточнение от другого. Поскольку семь параметров однозначно определяют некоторый алгоритм, то выбор семи классов изменения этих параметров определяет некоторый класс алгоритмов. Однако такой выбор может претендовать на название «уточнения», лишь если имеется убеждение, что для произвольного алгоритма, имеющего допускаемые данным выбором совокупность возможных исходных данных и совокупность возможных результатов, может быть указан равносильный ему алгоритм из определенного данным выбором класса алгоритмов.

Это убеждение формулируется для каждого уточнения в виде *основной гипотезы*, которая (при современном уровне наших представлений) не может быть предметом математического доказательства.

Алгоритмы прослеживаются в математике в течение всего времени ее существования. Общее понятие алгоритма сформировалось, однако, лишь в 20 веке и впервые появилось в трудах Э. Бореля (1912) и Г. Вейля (1921). Началом систематической разработки теории алгоритмов можно считать 1936 год, когда А.Черч опубликовал первое понятие вычислимой функции и привел первый пример функции, не являющейся вычислимой, а А. Тьюринг и Э. Пост дали первые уточнения понятия алгоритма (в терминах идеализированных вычислительных машин, машин Тьюринга), их конструкции во многом предвосхитили идеи, заложенные в основу современных цифровых вычислительных машин. В дальнейшем теория алгоритмов получила развитие в трудах С. Клини, Э. Поста, А.А. Маркова, А.Н. Колмогорова и других. В частности, А.А. Марков предложил уточнять понятие алгоритма с помощью введенного им понятия нормального алгорифма. Наиболее общий подход к уточнению понятия алгоритма предложил А.Н. Колмогоров: трактовать конструктивные объекты как топологические комплексы определенного вида, что дало возможность уточнить свойство «локальности» преобразования.

Основные понятия теории алгоритмов. Областью применимости алгоритма называется совокупность тех объектов, к которым он применим, т.е. в применении к которым дает результат.

Про алгоритм $\aleph$ говорят, что он

1) «вычисляет функцию f», коль скоро его область применимости совпадает с областью определения f, и $\aleph$ перерабатывает всякий x из своей области применимости в $f(x)$;

2) «разрешает множество A относительно множества X», коль скоро он применим ко всякому x из X, и перерабатывает всякий x из $X \cap A$ в слово «да», а всякий x из $X \setminus A$ - в слово «нет»;

3) «перечисляет множество B», коль скоро его область применимости есть натуральный ряд, а совокупность результатов есть B.

Функция называется *вычислимой*, если существует вычисляющий ее алгоритм. Множество называется *разрешимым* относительно X, если существует разрешающий его относительно X алгоритм. Множество называется *перечислимым*, если либо оно пусто, либо существует перечисляющий его алгоритм.

Детальный анализ понятия «алгоритм» обнаруживает, что

I) область возможных исходных данных и область применимости любого алгоритма суть перечислимые множества. В свою очередь,

II) для любой пары вложенных одно в другое перечислимых множеств можно подобрать алгоритм, у которого большее множество служит областью возможных исходных данных, а меньшее – областью применимости.

Имеют место следующие основные теоремы:

III) функция f вычислима тогда и только тогда, когда перечислим ее график, т.е. множество всех пар вида $\langle x, f(x)\rangle$.

IV) Подмножество A перечислимого множества X тогда и только тогда разрешимо относительно X, когда A и $X \setminus A$ перечислимы.

V) Если A и B перечислимы, то $A\cup B$ и $A\cap B$ также перечислимы.

VI) В каждом бесконечном перечислимом множестве X существует перечислимое подмножество с неперечислимым дополнением (в силу (IV) это перечислимое подмножество будет неразрешимым относительно X).

VII) Для каждого бесконечного перечислимого множества X существует вычислимая функция, определенная на подмножестве этого множества и не продолжаемая до вычислимой функции, определенной на всем X.

Утверждения (VI) и (II) в совокупности дают пример алгоритма с неразрешимой областью применимости.

Разрешимые и перечислимые множества составляют простейшие (и наиболее важные) примеры множеств, строение которых задается с помощью тех или иных алгоритмических процедур. Систематическое изучение множеств конструктивных объектов с точки зрения таких свойств этих множеств, которые связаны с наличием тех или иных алгоритмов, образует т.н. *алгоритмическую теорию множеств*.

Теорию алгоритмов можно разделить на *дескриптивную* (качественную) и *метрическую* (количественную). Первая исследует алгоритмы с точки зрения устанавливаемого ими соответствия между исходными данными и результатами; к ней относятся, в частности, проблемы построения алгоритма, обладающего теми или иными свойствами, – *алгоритмические проблемы*. Вторая исследует алгоритмы с точки зрения сложности как самих алгоритмов, так и задаваемых ими вычислений, т.е. процессов последовательного преобразования конструктивных объектов. Важно подчеркнуть, что как сложность алгоритмов, так и сложность вычислений могут определяться различными способами. Разработка методов оценки сложности алгоритмов и вычислений имеет важное теоретическое и практическое значение.

Приложения теории алгоритмов имеются во всех областях математики, в которых встречаются алгоритмические проблемы. Такие проблемы возникают практически во всех разделах математики. В математической логике для каждой теории формулируется проблема разрешения множества всех истинных или доказуемых предложений этой теории относительно множества всех ее предложений (теории подразделяются на разрешимые и неразрешимые – в зависимости от разрешимости или неразрешимости указанной проблемы); в 1936 году Черч установил неразрешимость проблемы разрешения для множества всех истинных предложений логики предикатов, дальнейшие важные результаты в этом направлении принадлежат А. Тарскому, А.И Мальцеву и др.

Неразрешимые алгоритмические проблемы встречаются:

– в алгебре (проблема тождества для полугрупп и, в частности, групп; первые примеры полугрупп с неразрешимой проблемой тождества были найдены в 1947 г. независимо А.А. Марковым и Э. Постом, а пример группы с неразрешимой проблемой тождества – в 1952 г. П.С. Новиковым);
– в топологии (проблема гомеоморфии, неразрешимость которой для важного класса случаев была доказана в 1958 г. А.А. Марковым);
– в теории чисел (проблема разрешимости диофантовых уравнений, неразрешимость которой была установлена в 1970 Ю.В. Матиясевичем);
– и в других разделах математики.

Теория алгоритмов тесно связана:

1) с математической логикой, поскольку в терминах теории алгоритмов может быть изложено одно из центральных понятий математической логики - понятие исчисления (и потому, например, теорема Геделя о неполноте формальных систем может быть получена как следствие теорем теории алгоритмов);
2) с основаниями математики, в которых одно из центральных мест занимает проблема соотношения конструктивного и неконструктивного (в частности, теория алгоритмов дает аппарат, необходимый для разработки конструктивного направления в математике); в 1965 г. А.Н. Колмогоров предложил использовать теорию алгоритмов для обоснования теории информации;
3) с кибернетикой, в которой важное место занимает изучение алгоритмов управления;
4) теория алгоритмов образует теоретический фундамент для ряда вопросов вычислительной математики.

КОНТРОЛЬНЫЕ ЗАДАНИЯ ПО КУРСУ

Задание 1. Тождества теории множеств

---

(1)

Доказать следующее тождество:

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

---

(2)

Доказать следующее тождество:

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

---

(3)

Доказать следующее тождество:

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

---

(4)

Доказать следующее тождество:

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

---

(5)

Доказать следующее тождество:

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

---

(6)

Доказать следующее тождество:

$$(A \setminus B) \setminus C)= (A \setminus C) \setminus (B \setminus C)$$

---

(7)

Доказать следующее тождество:

$$(A \cap B) \cup [\, A \cap (-B)] = (A \cup B) \cap [\, A \cup (-B)] = A$$

---

(8)

Доказать следующее тождество:

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

---

(9)

Доказать следующее тождество:

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

---

(10)

Доказать следующее тождество:

$$A \cup B = A — B — (A \cap B)$$

---

(11)

Доказать следующее тождество:

$$A \cup B = (A — B) \cup (A \cap B)$$

---

(12)

Доказать следующее тождество:

$$(A — B) \setminus C = (A \setminus C) — (B \cup C)$$

---

(13)

Доказать следующее тождество:

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

---

(14)

Доказать следующее тождество:

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

Задание 2. Отношения

Определите, являются ли указанные отображения морфизмами и, если «да», то какими:

1. Англо-русский словарь (отображение множества русских слов на множество английских)
2. Отображение множества фигур шахматной комбинации на множество занятых ими полей шахматной доски.
3. Отображение телефонов в городе на множество 5-значных номеров.
4. Нумерация произвольного счетного множества (его отображение на натуральный ряд)
5. Отображение алгебраических систем (Q_N, +) и (Q_{2N}, +): $n \to 2n$. (Q_N— множество всех целых чисел, Q_{2N} — множество всех четных чисел)
6. Отображение алгебраических систем (Q_N, +) и (Q_N, +): $n \to -n$. (Q_N— множество всех целых чисел)
7. Отображение алгебраических систем (Q_N, *) и (Q_N, *): $n \to -n$. (Q_N— множество всех целых чисел)
8. Отображение алгебраических систем (Q_N, *) и (Q_{2N}, *): $n \to 2n$. (Q_N— множество всех целых чисел, Q_{2N} — множество всех четных чисел)

9. Отображение алгебраических систем (N; +, *) и (N_7; ⊕, ⊗): $n \to n$ *mod 7*. (⊕ — сложение по модулю 7, ⊗ — умножение по модулю 7, N — множество натуральных чисел, N_7 — {0, 1, 2, 3, 4, 5, 6}).
10. Отображение множества студентов на множество их имён.
11. Отображение алгебраических систем (β({1,2,3,4,5,6,7,8,9}),∪,∩,–) и (β({*a*,*b*,*c*,*d*,*e*,*f*,*g*,*h*,*i*}),∪,∩,–), где β(*A*) — булеан (множество всех подмножеств множества *A*)

I.

1. Какие свойства имеет отношение “жить в одном городе”?
2. Какие свойства имеет отношение “быть знакомым”?
3. Какие свойства имеет отношение “быть моложе”?
4. Какие свойства имеет отношение “быть делителем” на множестве натуральных чисел?
5. Какие свойства имеет отношение “иметь общий делитель, отличный от единицы” на множестве натуральных чисел?
6. Какие свойства имеет отношение “быть потомком”?
7. Какие свойства имеет отношение “иметь общий остаток от деления на 5” на множестве натуральных чисел?
8. Какие свойства имеет отношение конгруэнтности на множестве треугольников?
9. Какие свойства имеет отношение подобия на множестве треугольников?

II.

1. Приведите пример рефлексивного отношения.
2. Приведите пример антирефлексивного отношения.
3. Приведите пример симметричного отношения.
4. Приведите пример антисимметричного отношения.
5. Приведите пример транзитивного отношения.
6. Приведите пример отношения эквивалентности.
7. Приведите пример отношения порядка.
8. Приведите пример отношения предпорядка.

III. Некоторое множество *N* имеет мощность *n*. Определите мощность множества *M*, представляющего из себя множество всех подмножеств множества *N*.

Задание 3. Гомоморфизмы. Бинарные операции. Группы

1

1. Выяснить, образуют ли группу целые неотрицательные числа относительно арифметического сложения.
2. Выяснить, образует ли группу множество всех ортогональных матриц n-го порядка относительно умножения матриц.
3. Перечислить все группы мощности $n \le 4$.

2

1. Выяснить, образуют ли группу целые числа относительно арифметического сложения.
2. Выяснить, образует ли группу множество чисел вида $a + b\sqrt{5}$, где a, b – рациональные числа, не равные одновременно нулю, относительно арифметического умножения.
3. Перечислить все группы мощности $n \le 4$.

3

1. Выяснить, образуют ли группу целые числа относительно арифметического вычитания.
2. Выяснить, образует ли группу множество всех квадратных матриц n-го порядка относительно умножения матриц.
3. Перечислить все группы мощности $n \le 4$.

4

1. Выяснить, образуют ли группу целые числа, кратные некоторому целому k, относительно арифметического сложения.
2. Выяснить, образует ли группу множество чисел вида $a + b\sqrt{5}$, где a, b – рациональные числа, не равные одновременно нулю, относительно арифметического умножения.
3. Перечислить все группы мощности $n \le 4$.

5

1. Выяснить, образует ли группу множество чисел вида $a + b\sqrt{5}$, где a, b – произвольные рациональные числа, относительно арифметического сложения.
2. Выяснить, образует ли группу множество положительных действительных чисел относительно арифметического умножения.
3. Перечислить все группы мощности $n \le 4$.

6

1. Выяснить, образует ли группу множество чисел вида $a+b\sqrt{5}$, где a, b – произвольные рациональные числа, относительно арифметического умножения.
2. Выяснить, образует ли группу множество всех квадратных невырожденных матриц n-го порядка относительно умножения матриц.
3. Перечислить все группы мощности $n \le 4$.

7

1. Выяснить, образует ли группу множество рациональных чисел относительно арифметического сложения.
2. Выяснить, образует ли группу множество всех квадратных матриц n-го порядка относительно умножения матриц.
3. Перечислить все группы мощности $n \le 4$.

8

1. Выяснить, образует ли группу множество рациональных чисел относительно арифметического умножения.
2. Выяснить, образует ли группу множество чисел вида $a+b\sqrt{5}$, где a, b – рациональные числа, не равные одновременно нулю, относительно арифметического умножения.
3. Перечислить все группы мощности $n \le 4$.

9

1. Выяснить, образует ли группу множество отличных от нуля рациональных чисел относительно арифметического умножения.
2. Выяснить, образует ли группу множество всех ортогональных матриц n-го порядка относительно умножения матриц.
3. Перечислить все группы мощности $n \le 4$.

10

1. Выяснить, образует ли группу множество положительных действительных чисел относительно арифметического умножения.
2. Выяснить, образует ли группу множество всех квадратных невырожденных матриц n-го порядка относительно умножения матриц.
3. Перечислить все группы мощности $n \le 4$.

Задание 4. Математическая логика. Кольца тела, поля

Вариант 1

1) Построить таблицу истинности для следующей формулы:
   $((P \supset Q) \vee (P \supset (Q \,\&\, P)))$
2) Доказать тождественную истинность формулы:
   $((P \supset Q) \vee (Q \supset P))$
3) Доказать эквивалентность:
   $((A \,\&\, B) \vee ((A \vee B) \,\&\, (\neg A \vee \neg B))) \sim (A \vee B)$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 2

1) Построить таблицу истинности для следующей формулы:
   $(\neg (P \supset \neg (Q \,\&\, P)) \supset (P \vee R))$
2) Доказать тождественную истинность формулы:
   $((P \supset Q) \vee (P \supset \neg Q))$
3) Доказать эквивалентность:
   $((A \vee B) \,\&\, (A \vee \neg B)) \sim A$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 3

1) Построить таблицу истинности для следующей формулы:
   $((P \,\&\, (Q \supset P)) \supset \neg P)$
2) Доказать тождественную истинность формулы:
   $(P \supset (Q \supset (P \,\&\, Q)))$
3) Доказать эквивалентность:
   $(A \vee (\neg A \,\&\, B)) \sim (A \vee B)$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 4

1) Построить таблицу истинности для следующей формулы:
   $(((P \,\&\, \neg Q) \supset Q) \supset (P \supset Q))$
2) Доказать тождественную истинность формулы:
   $((P \supset Q) \supset ((Q \supset R) \supset (P \supset R)))$
3) Доказать эквивалентность:
   $(A \vee (B \,\&\, \neg B)) \sim A$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 5

1) Построить таблицы истинности для следующей формулы:
$((P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R)))$
2) Доказать тождественную истинность формулы:
$((\neg P \supset \neg Q) \supset (Q \supset P))$
3) Доказать эквивалентность:
$(A \;\&\; (B \vee \neg B)) \sim A$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 6

1) Построить таблицы истинности для следующей формулы:
$((P \;\&\; (Q \vee \neg P)) \;\&\; ((\neg Q \supset P) \vee Q))$
2) Доказать тождественную истинность формулы:
$((P \supset Q) \supset ((P \supset \neg Q) \supset \neg P))$
3) Доказать эквивалентность:
$((A \vee B) \;\&\; (B \vee C) \;\&\; (C \vee A)) \sim ((A \;\&\; B) \vee (B \;\&\; C) \vee (C \;\&\; A))$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 7

1) Построить таблицу истинности для следующей формулы:
$((P \supset Q) \vee (P \supset (Q \;\&\; P)))$
2) Доказать тождественную истинность формулы:
$((\neg Q \supset \neg P) \supset (\neg Q \supset P) \supset Q)$
3) Доказать эквивалентность:
$\neg(A \vee B) \sim (\neg A \;\&\; \neg B)$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 8

1) Построить таблицу истинности для следующей формулы:
$(\neg (P \supset \neg (Q \;\&\; P)) \supset (P \vee R))$
2) Доказать тождественную истинность формулы:
$((Q \supset R) \supset ((P \vee Q) \supset (P \vee R))$
3) Доказать эквивалентность:
$\neg(A \supset B) \sim (A \;\&\; \neg B)$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 9

1) Построить таблицу истинности для следующей формулы:
   $((P \,\&\, (Q \supset P)) \supset \neg P\ )$
2) Доказать тождественную истинность формулы:
   $((P \supset Q) \supset ((P \supset (Q \supset R)) \supset (P \supset R)))$
3) Доказать эквивалентность:
   $\neg(A \,\&\, B) \sim (\neg A \vee \neg B)$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 10

1) Построить таблицу истинности для следующей формулы:
   $(((P \,\&\, \neg Q)) \supset Q) \supset (P \supset Q)$
2) Доказать тождественную истинность формулы:
   $(((P \supset Q) \supset P) \ \supset P)$
3) Доказать эквивалентность:
   $(A \vee (B \,\&\neg\, B)) \sim A$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 11

1) Построить таблицу истинности для следующей формулы:
   $((P \,\&\, (Q \supset P)) \supset \neg P\ )$
2) Доказать тождественную истинность формулы:
   $(P \supset (P \vee Q))$
3) Доказать эквивалентность:
   $((A \vee B) \,\&\, (B \vee C) \,\&\, (C \vee A)) \sim ((A \,\&\, B) \vee (B \,\&\, C) \vee (C \,\&\, A))$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

Вариант 12

1) Построить таблицу истинности для следующей формулы:
   $(((P \,\&\, \neg Q)) \supset Q) \supset (P \supset Q)$
2) Доказать тождественную истинность формулы:
   $((P \,\&\, Q) \supset Q)$
3) Доказать эквивалентность:
   $((A \,\&\, B) \vee ((A \vee B) \,\&\, (\neg A \vee \neg\, B))) \sim (A \vee B)$
4) Перечислить пары булевских операций, образующие кольца, тела, поля.

1. Образует ли кольцо, тело, поле

1) множество всех целых чисел;
2) множество всех целых чисел, кратных данному целому числу *m*;
3) множество всех рациональных чисел;
4) множество всех действительных чисел;
5) множество всех комплексных чисел;
6) множество всех квадратных матриц порядка *n*;
7) множество всех векторов 3-мерного пространства относительно обычного сложения и векторного умножения?

2. Перечислить (привести примеры) колец, тел и полей мощности $n \le 5$.

Задание 5. Графы. Задание графа, связность

Вариант 1

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	0	1	0	0	1
B	0	1	1	1	1	1
C	1	0	0	1	0	0
D	0	0	1	0	0	1
E	0	0	1	0	0	1
F	1	0	0	1	1	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 2

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	1	1	1	0	1	1
B	1	1	0	1	1	0
C	1	1	0	1	0	0
D	1	1	0	1	0	0
E	0	0	0	1	1	1
F	1	0	1	1	0	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 3

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	1	0	1	0	1
B	0	0	1	1	0	1
C	1	1	0	0	1	0
D	0	1	0	0	0	1
E	1	1	0	0	1	1
F	0	0	0	0	0	1

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 4

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	1	1	1	0	1	1
B	1	0	0	1	1	0
C	0	1	0	1	1	0
D	1	0	1	1	1	1
E	0	1	1	0	1	1
F	1	1	1	0	0	1

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 5

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	1	1	0	0	0
B	1	1	0	0	0	1
C	0	1	1	0	0	0
D	0	1	1	0	1	0
E	0	1	0	1	0	0
F	0	1	0	0	0	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 6

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	1	0	0	1	0	0
B	1	1	0	0	1	0
C	0	0	0	0	1	0
D	0	0	1	0	0	0
E	0	1	0	1	0	0
F	0	1	0	0	0	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 7

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	1	0	1	0	1
B	1	1	1	1	0	0
C	1	1	1	0	1	1
D	1	1	1	1	0	0
E	0	0	1	0	0	1
F	1	1	1	1	1	1

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 8

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0	1	1
C	1	0	0	1	1	1
D	0	0	1	1	0	0
E	1	0	0	1	0	1
F	0	0	1	0	0	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 9

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	1	1	1	1	1
B	1	1	0	0	1	0
C	0	1	0	1	1	0
D	0	1	0	0	1	0
E	0	0	1	0	1	0
F	0	1	0	0	0	1

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 10

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	1	0	1	1	1
B	1	0	1	0	1	1
C	1	0	0	1	0	1
D	1	0	1	0	0	1
E	0	1	0	0	0	0
F	1	1	0	0	0	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 11

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	0	1	0	1	1
B	0	1	1	0	0	0
C	1	1	0	1	0	0
D	0	1	1	1	0	0
E	1	0	0	0	0	0
F	1	1	0	1	0	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 12

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	1	0	0	0	1	1
B	0	0	0	0	0	1
C	0	0	1	0	1	0
D	0	0	1	0	1	0
E	1	0	1	1	0	1
F	1	1	1	1	1	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Вариант 13

1. Нарисовать граф, заданный следующей матрицей смежности:

	A	B	C	D	E	F
A	0	1	1	1	0	1
B	1	0	1	0	1	1
C	1	0	1	1	1	0
D	0	0	0	1	1	1
E	1	0	0	0	1	0
F	0	1	1	1	0	0

2. Является ли данный граф сильно связным, односторонне связным, слабо связным, несвязным.
3. Описать данный граф при помощи матрицы инциденций, списка ребер, списка инциденций.

Задание 6. Графы. Обходы (Эйлер, Гамильтон)

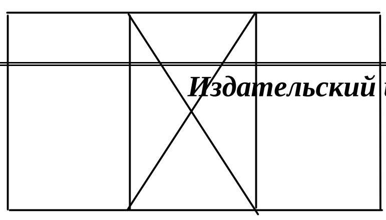
Для каждого из четырех графов ответить на следующие вопросы:

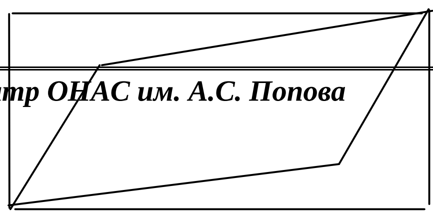
а) есть ли в нем циклы Эйлера? Если есть, то привести пример, если нет, то сказать, почему;

б) есть ли в нем пути Эйлера? Если есть, то привести пример, если нет, то сказать, почему;

в) есть ли в нем циклы Гамильтона? Если есть, то привести пример;

г) есть ли в нем пути Гамильтона? Если есть, то привести пример;





Задание 9. Алгоритмы в графах

1. Максимальный связный подграф графа G называется *компонентой связности*, или просто *компонентой* графа G.

Представить алгоритм нахождения количества компонент связности графа.

---

2. Максимальный связный подграф графа G называется *компонентой связности*, или просто *компонентой* графа G. *Мостом* графа G называется каждое ребро, удаление которого приводит к увеличению числа *компонент связности*.

Представить алгоритм нахождения всех мостов графа.

---

3. Максимальный связный подграф графа G называется *компонентой связности*, или просто *компонентой* графа G. *Точкой сочленения* графа G называется каждая вершина, удаление которой (вместе с инцидентными ей ребрами) приводит к увеличению числа *компонент связности*.

Представить алгоритм нахождения всех точек сочленения графа.

---

4. *Кликой* называется произвольное подмножество вершин, в котором каждая пара различных вершин соединена ребром.

Представить алгоритм нахождения клики наибольшей мощности в неориентированным графе.

---

5. *Двудольный граф* (или *биграф*) G – это граф, множество вершин V которого можно разбить на два подмножества V_1 и V_2 таким образом, что каждое ребро графа G соединяет вершины из разных подмножеств.

Представить алгоритм, проверяющий, будет ли данный граф двудольным.

---

6. *Связностью* к графа G называется наименьшее число вершин, удаление которых приводит к несвязному или тривиальному (одна вершина, ноль ребер) графу.

Представить алгоритм, определяющий связность к графа.

---

7. *Реберной связностью* λ графа G называется наименьшее количество ребер, удаление которых приводит к несвязному или тривиальному (одна вершина, ноль ребер) графу.

Представить алгоритм, определяющий реберную связность λ графа.

---

8. *Длина маршрута* $v_0v_1\ldots v_n$ равна n, т. е. количеству ребер в нем. *Обхват* графа G – это длина кратчайшего простого цикла графа G (если он есть). Это понятие не определено в случае, когда в G нет циклов.

Представить алгоритм, определяющий обхват графа.

---

9. *Длина маршрута* $v_0v_1\ldots v_n$ равна n, т. е. количеству ребер в нем. *Окружение* графа G — это длина самого длинного простого цикла графа G (если он есть). Это понятие не определено в случае, когда в G нет циклов.

Представить алгоритм, определяющий окружение графа.

---

10. *Длина цепи* $v_0v_1\ldots v_n$ равна n, т. е. количеству ребер в ней. *Расстоянием* $d(u, v)$ между двумя вершинами u и v графа G называется длина кратчайшей простой цепи, соединяющей их; если u и v не соединены, то полагаем $d(u, v) = \infty$.

Кратчайшая простая (u-v)-цепь часто называется *геодезической*. *Диаметр* $d(G)$ связного графа G — это длина самой длинной геодезической.

Представить алгоритм, определяющий диаметр графа.

---

11. Максимальный связный подграф графа G называется *компонентой связности*, или просто *компонентой* графа G. *Точкой сочленения* графа G называется каждая вершина, удаление которой (вместе с инцидентными ей ребрами) приводит к увеличению числа *компонент связности*. *Неразделимым* графом называется связный, непустой, не имеющий точек сочленения граф. *Блок* графа — это его максимальный неразделимый подграф.

Представить алгоритм нахождения блока графа.

---

12. *Дерево* — это связный ациклический (не имеющий циклов) граф. Каждый граф, не содержащий циклов, называется лесом.

Представить алгоритм, определяющий, является ли граф деревом, лесом.

Задание 10. Конечные автоматы

1. Придумать автомат для поиска во входной строке слова-фрагмента «какаду». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

2. Придумать автомат для поиска во входной строке слова-фрагмента «мамаша». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

3. Придумать автомат для поиска во входной строке слова-фрагмента «папаша». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

4. Придумать автомат для поиска во входной строке слова-фрагмента «ананас». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

5. Придумать автомат для поиска во входной строке слова-фрагмента «макака». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

6. Придумать автомат для поиска во входной строке слова-фрагмента «кокосы». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

7. Придумать автомат для поиска во входной строке слова-фрагмента «гагауз». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

8. Придумать автомат для поиска во входной строке слова-фрагмента «лелеять». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

9. Придумать автомат для поиска во входной строке слова-фрагмента «кукушка». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

10. Придумать автомат для поиска во входной строке слова-фрагмента «рококо». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

11. Придумать автомат для поиска во входной строке слова-фрагмента «гагара». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

12. Придумать автомат для поиска во входной строке слова-фрагмента «татарин». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

13. Придумать автомат для поиска во входной строке слова-фрагмента «кокотка». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

14. Придумать автомат для поиска во входной строке слова-фрагмента «кокошник». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

15. Придумать автомат для поиска во входной строке слова-фрагмента «мимика». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

16. Придумать автомат для поиска во входной строке слова-фрагмента «папаха». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

17. Придумать автомат для поиска во входной строке слова-фрагмента «тетерев». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

18. Придумать автомат для поиска во входной строке слова-фрагмента «кикиморки». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

19. Придумать автомат для поиска во входной строке слова-фрагмента «чечевица». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

20. Придумать автомат для поиска во входной строке слова-фрагмента «попович». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

---

21. Придумать автомат для поиска во входной строке слова-фрагмента «лилиана». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

22. Придумать автомат для поиска во входной строке слова-фрагмента «хохотало». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

23. Придумать автомат для поиска во входной строке слова-фрагмента «тотоша». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

24. Придумать автомат для поиска во входной строке слова-фрагмента «бабахнуть». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

25. Придумать автомат для поиска во входной строке слова-фрагмента «комком». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

26. Придумать автомат для поиска во входной строке слова-фрагмента «тамтам». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

27. Придумать автомат для поиска во входной строке слова-фрагмента «канкан». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

28. Придумать автомат для поиска во входной строке слова-фрагмента «какао». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

29. Придумать автомат для поиска во входной строке слова-фрагмента «гогот». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

30. Придумать автомат для поиска во входной строке слова-фрагмента «кукурузку». Описать автомат двумя способами: а) таблицей состояний; б) диаграммой состояний.

13. ВОПРОСЫ К ЗАЧЕТУ ПО ОСНОВАМ ДИСКРЕТНОЙ МАТЕМАТИКИ

1. Абелева группа
2. Абелевость
3. Автоморфизм
4. Аддитивная и мультипликативная формы записи
5. Алгебра Вебба
6. Алгебра Поста
7. Алгебра Россера-Тьюкетта
8. Антиномия Рассела
9. Антисимметричность
10. Ассоциативность
11. Биекция
12. Биморфизм
13. Бинарные отношения
14. Булева алгебра
15. Вектор ошибок
16. Вложение
17. Выполнимая формула
18. Гомеоморфизм
19. Гомоморфизм
20. Группа
21. Групповые коды
22. Группоид
23. Декартово произведение множеств
24. Делитель нуля
25. Диаграмма Хассе
26. Дизъюнкция
27. ДНФ
28. Единичное множество
29. Законы и правила исчисления высказываний
30. Законы и правила исчисления предикатов
31. И
32. Идеал
33. Идемпотентность
34. Изоморфизм
35. Или

36 Импликация
37 Инъекция
38 Исключающее или
39 Исчисление высказываний
40 Исчисление предикатов
41 Квазигруппа
42 Кванторы существования и общности
43 Класс
44 Коды, исправляющие ошибки
45 Код Хэмминга
46 Кольцо
47 Кольцо вычетов Zm
48 Коммутативность
49 Композиция
50 Конгруэнция
51 Конъюнкция
52 Круги Эйлера
53 Латинский квадрат
54 Матричное кодирование
55 Метрические свойства кодов
56 Множество
57 Множество подмножеств
58 Мономорфизм
59 Морфизмы
60 Мощность множества
61 Неравнозначность
62 Несовместность
63 Нормальная (инвариантная) подгруппа
64 Образ отображения
65 Образующий элемент
66 Общезначимое высказывание
67 Общезначимая формула
68 Операции и алгебры
69 Операции над множествами
70 Операции над отношениями

71 Отношение как предикат и как множество кортежей
72 Отношение предпорядка
73 Отношение строгой упорядоченности
74 Отношение упорядоченности
75 Отношение. Способы задания отношений
76 Отношение частичного порядка
77 Отношение эквивалентности
78 Отображения (морфизмы) и их примеры
79 Отрицание
80 Парадокс Рассела
81 Подгруппа
82 Поле
83 Порядок группы
84 Порядок элемента
85 Предикат
86 Применение соответствия Галуа в проективном распознавании образов
87 Примитивность
88 Принцип двойственности
89 Прообраз отображения
90 Пустое множество
91 Расстояние Хэмминга
92 Рефлексивность
93 Решетка
94 Свободная группа
95 Связь отображения и эквивалентности
96 Связь бинарной операции с отображением множеств
97 Смежный класс
98 Соответствие Галуа
99 Способы задания множеств
100 Способы задания отношений
101 Сравнимость целых чисел по фиксированному модулю
102 Структура
103 Сюръекция
104 Таблица Кэли
105 Таблица умножения
106 Тавтология
107 Теоремы о гомоморфизмах
108 Теорема о единственности конечного поля данного порядка
109 Транзитивность
110 Универсальное множество
111 Универсум
112 Фактогруппа по нормальному делителю
113 Факторкольцо
114 Центр кольца
115 Циклическая группа
116 Штрих Шеффера
117 Эквивалентность
118 Эндоморфизм
119 Эпиморфизм
120 Ядро гомоморфизма

14. ЗАДАНИЯ ДЛЯ ДОПУСКА К ЭКЗАМЕНУ ПО ОДМ

1. Ввод и визуализация неориентированного графа и орграфа.
2. Проверка связности неориентированного графа, сильной, односторонней и слабой связности для орграфа.
3. Нахождение цикла Эйлера в неориентированном графе и орграфе.
   3*. Нахождение всех (без циклических сдвигов) циклов Эйлера в неориентированном графе и орграфе.
   3**. Нахождение путей Эйлера в неориентированном графе и орграфе.
4. Нахождение всех циклов Гамильтона в неориентированном графе и орграфе.
5. Индивидуальное задание по графам.
6. Нахождение всех элементарных циклов (контуров) в неориентированном графе (орграфе).
7. Нахождение кратчайшего расстояния между двумя вершинами графа (орграфа). Веса – произвольные вещественные числа.
8. Нахождение кратчайших путей между двумя вершинами графа (орграфа).
9. Знаковые неориентированные графы: проверка сбалансированности и разбиение на классы.
10. Знаковые орграфы: перечисление контуров положительной и отрицательной обратной связи.
11. Придумать и запрограммировать автомат для поиска во входной строке заданного слова-фрагмента.
12. Описать грамматику арифметических выражений в обратной польской записи с операциями «+», «–», «*», «/» и операндами, образованными из одной буквы латинского алфавита (первый символ имени) и произвольного множества цифр. Реализовать в рамках данной грамматики: а) генератор конечных цепочек; б) проверку выводимости.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Обязательная

1. *Лавров И. А., Максимова Л. Л.* Задачи по теории множеств, математической логике и теории алгоритмов. М.: Физматлит, 1995.
2. *Нефедов В.Н., Осипова В.А.* Курс дискретной математики. – М.: Изд-во МАИ, 1992.

Дополнительная

1. *Амамия М., Танака Ю.* Архитектура ЭВМ и искусственный интеллект. – М.: Мир, 1993.
2. *Багавантам С., Венкатарайуду Т.* Теория групп и ее применение к физическим проблемам. – М.: Изд-во иностранной литературы, 1959.
3. *Бальдассарри М.* Алгебраические многообразия. – М.: Изд-во иностранной литературы, 1961.
4. *Берзтисс А. Т.* Структуры данных. – М.: Статистика, 1974.
5. *Биркгоф Г., Барти Т.* Современная прикладная алгебра. – М.: Мир, 1976.
6. *Бурбаки Н.* Теория множеств. – М.: Мир, 1965.
7. *Гинзбург С.* Математическая теория контекстно-свободных языков. – М.: Мир, 1970.
8. *Горбатов В.А.* Основы дискретной математики. – М.: Высшая школа,1986.
9. *Гроссман И., Магнус В.* Группы и их графы. – М.: Мир, 1971.
10. *Гротедник А.* О некоторых вопросах гомологической алгебры. – М.: Изд-во иностранной литературы, 1961.
11. *Джексон Н.* Строение колец. – М.: Изд-во иностранной литературы, 1961.
12. *Джордж Ф.* Мозг как вычислительная машина. – М.: Изд-во иностранной литературы, 1963.
13. *Евстигнеев В. А.* Применение теории графов в программировании. – М.: Наука, 1985.
14. *Картеси Ф.* Введение в конечные геометрии. – М.: Наука, 1980.
15. *Касинов В. Б.* О симметрии в биологии. – Л.: Наука, 1971.
16. *Кац М., Улам С.* Математика и логика. Ретроспектива и перспективы. – М.: Мир, 1971.
17. *Колмогоров А. Н., Драгалин А. Г.* Введение в математическую логику. – М.: Изд-во Московского университета, 1982.
18. *Комбинаторный анализ.* Задачи и упражнения / Под ред. К.А. Рыбникова – М.: Наука, 1982.
19. *Кон П.* Универсальная алгебра. – М.: Мир, 1968.
20. *Криницкий Н. А.* Алгоритмы вокруг нас. – М.: Наука, 1977.
21. *Криницкий Н. А.* Алгоритмы и роботы. – М.: Радио и связь, 1983.
22. *Кузичев А.С.* Диаграммы Венна. История и применения. – М.: Наука,1968.
23. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера. – М.: Энергоатомиздат, 1988.
24. *Курош А. Г.* Курс высшей алгебры. – М.: Наука, 1975.
25. *Курош А. Г.* Лекции по общей алгебре. – М.: Физматлит, 1962.

26. *Курош А. Г.* Теория групп. – М., – Л.: Огиз, Гостехиздат, 1944.
27. *Ленг С.* Алгебра. – М.: Мир, 1968.
28. *Липский В.* Комбинаторика для программистов. – М.: Мир, 1988.
29. *Ляпин Е. С.* Полугруппы. – М.: Физматлит, 1960.
30. *Ляпин Е. С., Айзенштат А. Я., Лесохин М. М.* Упражнения по теории групп. – М.: Наука, 1967.
31. *Миркин Б. Г., Родин С. Н.* Графы и гены. – М.: Наука, 1977.
32. *Нечеткие* множества и теория возможностей. Последние достижения / Под ред. Р. Ягера – М.: Радио и связь, 1986.
33. *Оре О.* Графы и их применение. – М.: Мир, 1965.
34. *Пиотровский Р.Г., Бектаев К.Б., Пиотровская А.А.* Математическая лингвистика. – М.: Высшая школа, 1977.
35. *Попов Ю.П., Пухначев Ю.В.* Математика в образах. – М.: Знание, 1989.
36. *Постников М. М.* Теория Галуа. – М.: Физматлит, 1963.
37. *Рейнгольд Э., Нивергельт Ю., Део Н.* Комбинаторные алгоритмы. Теория и практика. – М.: Мир, 1980.
38. *Робертс Ф. С.* Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. – М.: Наука, 1986.
39. *Сачков В. Н.* Введение в комбинаторные методы дискретной математики. – М.: Наука, 1982.
40. *Серпинский В.* О теории множеств. – М.: Просвещение, 1966.
41. *Скорняков Л. А.* Элементы теории структур. – М.: Наука, 1982.
42. *Судзуки М.* Строение группы и строение структуры ее подгрупп. – М.: Изд-во иностранной литературы, 1960.
43. *Супруненко Д. А.* Группы матриц. – М.: Наука, 1972.
44. *Теория графов.* Покрытия, укладки, турниры / Сборник переводов под ред. В. Б. Алексеева, Г. П. Гаврилова, А.А. Сапоженко – М.: Мир, 1974.
45. *Успенский В. А., Семенов А. Л.* Теория алгоритмов: основные открытия и приложения. – М.: Наука, 1987.
46. *Харари Ф.* Теория графов. – М.: Мир, 1973.
47. *Хейне В.* Теория групп в квантовой механике. – М.: Изд-во иностранной литературы, 1963.
48. *Холл М.* Теория групп. – М.: Изд-во иностранной литературы, 1962.
49. *Чеботарев Н. Г.* Введение в теорию алгебр. – М., – Л.: ОГИЗ, 1949.
50. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. – М.: Наука, 1983.
51. *Чунихин С. А.* Подгруппы конечных групп. – Минск: Наука и техника, 1964.
52. *Шапиро Г. М.* Высшая алгебра. – М.: Учпедгиз, 1935.
53. *Эндрюс Г.* Теория разбиений. – М.: Наука, 1982.
54. *Яглом И. М.* Булева структура и ее модели. – М.: Советское радио, 1980.
55. *Яглом И. М.* Конечная алгебра, конечная геометрия и коды. – М.: Знание, 1980.

УЧЕБНОЕ ПОСОБИЕ
ДЛЯ ВСЕХ СПЕЦИАЛЬНОСТЕЙ

ПЛОТНИКОВ
ВАЛЕРИЙ МИХАЙЛОВИЧ

КУРС ДИСКРЕТНОЙ МАТЕМАТИКИ

Рецензент	***Л.И. Соколов***
Редактор	***А.М. Харсун***
Компьютерная правка и макетирование	***Е.С. Корнейчук***