

**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное бюджетное государственное образовательное
учреждение высшего профессионального образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

Г.А. Аршинов, В.Г. Аршинов

**ДИСКРЕТНАЯ МАТЕМАТИКА
Курс лекций для бакалавров**

Краснодар – 2013

УДК 519(075.8)
ББК 32.973.3
А20

Рецензенты:

Доктор технических наук В.В. Степанов – профессор факультета компьютерных технологий и автоматизированных систем Кубанского государственного технологического университета.

Доктор экономических наук Е.В. Луценко – профессор кафедры компьютерных технологий и систем Кубанского государственного аграрного университета.

А20 Аршинов Г.А., Аршинов В.Г. Дискретная математика: Курс лекций для бакалавров. – Краснодар: ФБГОУ ВПО «Кубанский государственный аграрный университет», 2013. – 68 с.

В учебном пособии изложены элементы теории множеств, логики предикатов, комбинаторики, элементы теории графов, автоматов и алгоритмов. Учебное пособие подготовлено в соответствии с требованиями Государственного образовательного стандарта высшего профессионального образования для специальностей «Бизнес-информатика», «Прикладная информатика»

ISBN 978-5-94672-308-4

© Аршинов Г. А., Аршинов В.Г. 2013.

© Федеральное бюджетное государственное образовательное учреждение высшего профессионального образования «Кубанский государственный аграрный университет», 2013.

СОДЕРЖАНИЕ

Введение	5
Лекция 1. Элементы теории множеств.....	6
1.1.Основные понятия и определения.....	6
1.2.Операции над множествами и свойства операций.....	6
1.3.Соответствия, отображения, функции и отношения.....	9
Лекция 2. Элементы логики предикатов.....	14
2.1.Понятие предиката.....	14
2.2. Операции над предикатами.....	17
Лекция 3. Применение логики предикатов.....	19
3.1.Применение логики предикатов для описания отношений.....	19
3.2. Применение логики предикатов для описания функциональных зависимостей и их свойств.....	20
Лекция 4. Формулы логики предикатов.....	23
4.1. Кванторные операции над многоместными предикатами.....	23
4.2.Формулы логики предикатов. Преобразование формул.....	26
Лекция 5. Правила комбинаторики. Размещения.....	29
5.1. Правило суммы и произведения.....	29
5.2. Размещения с повторениями.....	30
5.3. Размещения без повторений.....	31
Лекция 6. Перестановки и сочетания без повторений.....	32
6.1. Перестановки без повторений.....	32
6.2. Сочетания без повторений.....	33
Лекция 7. Графы.....	35
7.1. Основные понятия и определения.....	35
7.2.Элементы графов.....	37
7.3. Представление графов в ЭВМ.....	38
Лекция 8. Задачи оптимизации на графах.....	42
8.1. Кратчайший путь на графе. Метод приписывания индексов.....	42
8.2. Граф наименьшей длины.....	44
Лекция 9. Основные понятия теории автоматов.....	46
9.1. Конечные автоматы	46
9.2. Способы задания автоматов.....	48
9.3. Другие модели конечных автоматов.....	52
Лекция 10. Минимизация конечных автоматов	54
10.1. Эквивалентность конечных автоматов.....	54
10.2. Построение минимального конечного автомата.....	57
Лекция 11. Основные понятия теории алгоритмов.....	60

11.1. Неформальное понятие алгоритма. Свойства алгоритмов.....	61
11.2.Способы описания алгоритмов.....	62
11.3. Виды алгоритмических процессов.....	64
Литература.....	68

Введение

К дискретной математике обычно относят такие области математического знания, как комбинаторика, теория чисел, математическая логика, теория алгебраических систем, теория графов и сетей, теория конечных автоматов. Наиболее значимой областью применения методов дискретной математики является область компьютерных технологий. Это вызвано необходимостью создания и эксплуатации электронных вычислительных машин, средств передачи и обработки информации, автоматизированных систем управления и проектирования, разработки методов решения прикладных дискретных задач оптимизации. Дискретная математика является одной из основных дисциплин при подготовке специалистов в области прикладной информатики, программирования, а также по экономическим, техническим и гуманитарным направлениям.

Цель настоящего учебного пособия – дать студентам основы дискретной математики, показать некоторые области ее приложения, вооружить учащихся методами, применяемыми для решения широкого круга задач, содействовать формированию конструктивного стиля мышления. Пособие рассчитано на семестровый курс. Оно написано в доступной форме с определенной степенью строгости, однако изучение материала не требует специальной математической подготовки.

Учебное пособие предназначено прежде всего для студентов специальности «Прикладная информатика», а также студентов других специальностей, изучающих дискретную математику.

Лекция 1. Элементы теории множеств.

Учебные вопросы.

1.1.Основные понятия и определения.

1.2.Операции над множествами и свойства операций.

1.3.Соответствия, отображения, функции и отношения.

1.1.Основные понятия и определения.

Определение. Множеством называют совокупность элементов, объединенных по некоторому общему признаку.

Примеры: множество студентов университета, множество натуральных чисел.

Множества обозначаются большими буквами латинского алфавита, а элементы множеств - соответствующими малыми буквами с нижними индексами, нумерующими сами элементы.

Пример. Множество $A = \{a_1, a_2, \dots, a_n, \dots\}$ содержит элементы $a_1, a_2, \dots, a_n, \dots$. Если элемент a_i принадлежит множеству A , то пишется $a_i \in A$.

Определение. Множество B называют подмножеством множества A , если каждый элемент B является также и элементом множества A . Этот факт обозначается $A \subset B$. (Читается: «Множество A включает множество B », или « B является подмножеством множества A ».)

Определение. Множество, которое не содержит ни одного элемента, называют пустым множеством.

Пустое множество является подмножеством любого множества и обозначается символом \emptyset .

Определение. Мощностью конечного множества называется число элементов этого множества.

Множества могут содержать конечное или бесконечное число элементов. В первом случае их называют конечными, во втором – бесконечными. Например, множество студентов в группе – конечное, а множество целых положительных чисел, кратных 3 – бесконечное.

1.2.Операции над множествами и свойства операций.

Определение. Множества A, B называются равными, если они состоят из одинаковых элементов.

Имеются множества $A = \{a_1, a_2, \dots, a_n \dots\}$ и $B = \{b_1, b_2, \dots, b_n \dots\}$.

Определение. Объединением или суммой двух множеств A и B называют третье множество C , которое состоит из элементов, принадлежащих хотя бы одному из множеств A, B .

Объединение двух множеств A и B обозначается $A \cup B$ или $A + B$. Пишется $C = A \cup B$ или $C = A + B$. На рис. 1 геометрически показано объединение двух множеств A и B , элементами которых являются соответственно точки, лежащие внутри левого и правого контуров.



$A+B.$

Рис.1 Объединение множеств A, B .

Пример. $A = \{1, 3, 5\}$, $B = \{4, 7, 1, 3\}$, тогда $A+B = \{1, 3, 5, 4, 7\}$

Определение. Пересечением (или произведением) двух множеств A и B называют третье множество C , которое состоит из элементов, принадлежащих одновременно и множествам A и B (рис. 2).

Пересечение двух множеств A и B обозначается $A \cap B$ или $A \cdot B$. Пишется $C = A \cap B$ или $C = A \cdot B$. На рис.2 геометрически показано пересечение двух множеств A и B , элементами которых являются соответственно точки, лежащие внутри левого и правого контуров.

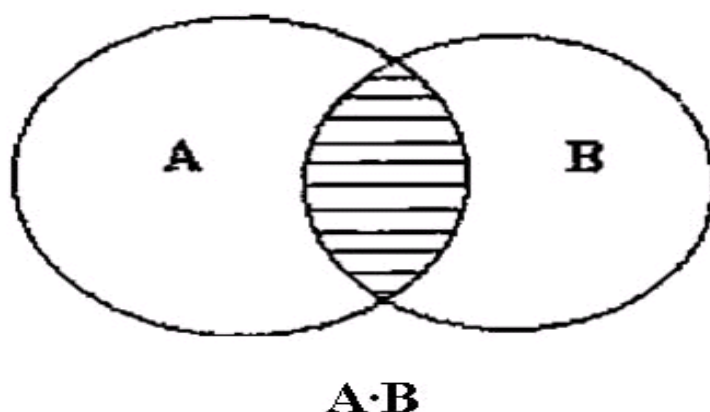


Рис.2. Пересечение множеств A, B.

Пример. $A=\{1,3,5\}$, $B=\{4,7,1,3\}$, тогда $A \cdot B=\{1,3\}$

Определение. Разностью множеств B и A называют множество, состоящие из тех элементов множества B, которые не принадлежат A. Разность множеств B и A обозначается $B \setminus A$ (рис.3).

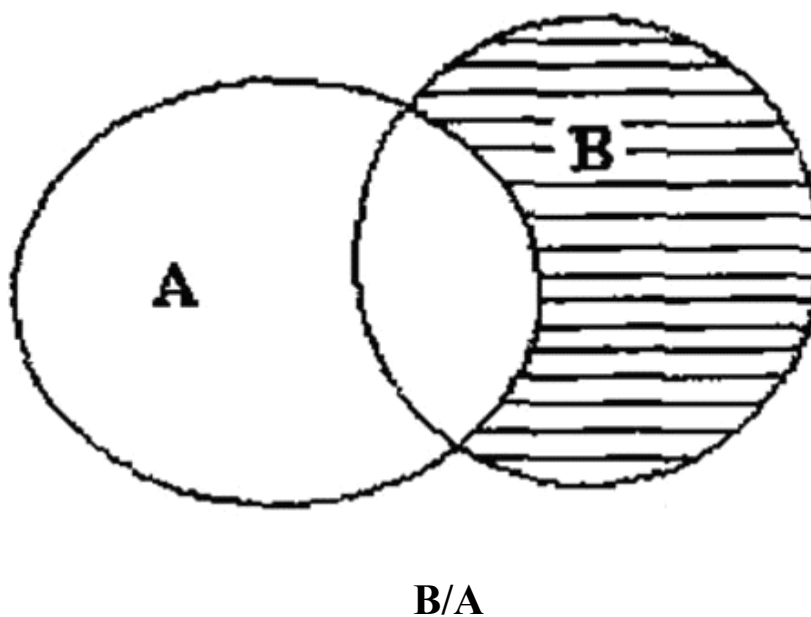


Рис. 3. Разность множеств B, A.

Пример. $A=\{1,3,5\}$, $B=\{4,7,1,3\}$, тогда $B/A=\{4,7\}$.

Определение. Универсальным называется множество всех мыслимых множеств. Будем обозначать его буквой U. Геометрически элементы универсального множества изображаются точками бесконечной плоскости.

Определение. Дополнением множества A до универсального множества U называется множество, обозначаемое символом \bar{A} и равное

разности $U \setminus A$.

Некоторые свойства операций над множествами:

1. $A+B = B+A$ – коммутативность объединения множеств.
2. $A \cdot B = B \cdot A$ – коммутативность пересечения множеств.
3. $A+(B+C)=(A+B)+C$ – ассоциативность объединения множеств.
4. $A \cdot (B \cdot C)=(A \cdot B) \cdot C$ – ассоциативность пересечения множеств.
5. $A \cdot (B+C)=A \cdot B+A \cdot C$ – дистрибутивность пересечения относительно объединения.
6. $A+(B \cdot C)=(A+B) \cdot (A+C)$ – дистрибутивность объединения относительно пересечения.
7. $A+A \cdot C=A$ – закон поглощения.
8. $A \cdot (A+C) = A$ – закон поглощения.

Пусть имеются множества: $A = \{a_1, a_2, \dots, a_n, \dots\}$ и $B = \{b_1, b_2, \dots, b_n, \dots\}$.

Определение. Декартовым или прямым произведением множеств A и B называют множество C , состоящее из всех упорядоченных пар, в которых первый элемент выбран из множества A , а второй – из множества B : $C = \{(a_1, b_1), (a_1, b_2), (a_1, b_n), \dots, (a_2, b_1), (a_2, b_2), \dots, (a_2, b_n), \dots\}$.

Декартово произведение множеств X и Y обозначается $A \times B$, то есть $C = A \times B$. Пара является упорядоченной, то есть порядок элементов в ней имеет существенное значение, иными словами, пара (a, b) не равна паре (b, a) .

Примеры декартового произведения.

1. Пусть имеются множества: $A = \{1, 3, 6\}$ и $B = \{3, 7\}$, тогда $A \times B = \{(1, 3), (1, 7), (3, 3), (3, 7), (6, 3), (6, 7)\}$

1.3. Соответствия, отображения, функции и отношения.

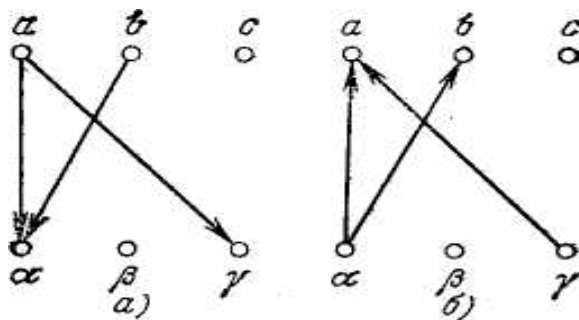
Определение. Будем говорить, что между множествами X , Y установлено соответствие, если по определенному правилу каждому элементу подмножества $A \subset X$ сопоставляется элемент подмножества $B \subset Y$. При этом совершенно необязательно, чтобы в сопоставлении участвовали все элементы множеств X и Y . Соответствие между двумя элементами множеств A, B называются бинарным отношением.

Обозначив соответствие q , его можно представить тройкой множеств $q = (X, Y, Q)$. X называется областью отправления соответствия, а Y – областью прибытия. Множество A называется областью определе-

ния соответствия, V —областью значения соответствия, а множество Q является подмножеством прямого произведения $X \times Y$.

Пример 1. Пусть $X = \{1, 2\}$, $Y = \{3, 5\}$, тогда $X \times Y = \{(1, 3), (1, 5), (2, 3), (2, 5)\}$. Это множество дает возможность получить 16 различных соответствий. Некоторые из них: $Q_1 = \{(1, 3)\}$; $Q_2 = \{(1, 3), (1, 5)\}$.

Пример 2. На предприятии есть две грузовые автомашины α , β , работающие в две смены, и автобус γ , используемый редко. Машина β находится в ремонте. В штате имеются три шофера a , b , c , из которых c находится в отпуске. Любое распределение шоферов по машинам представляет собой соответствие. Одним из возможных соответствий будет следующее: $q = (\{a, b, c\}, \{\alpha, \beta, \gamma\}, \{(a, \alpha), (a, \gamma), (b, \alpha)\})$, в котором область определения соответствия $A = \{a, b\}$, область значений $V = \{\alpha, \gamma\}$, $Q = \{(a, \alpha), (a, \gamma), (b, \alpha)\}$ (рис.1).



**Рис.1.Соответствие между водителями и машинами:
а- прямое соответствие, б-обратное соответствие.**

Элемент a соответствует элементам α и γ , а элемент b — элементу α .

Определение. Композицией соответствий называют последовательное применение двух соответствий.

Композиция соответствий есть операция с тремя множествами X , Y и Z , на которых определены два соответствия $q = (X, Y, Q)$, и $p = (Y, Z, P)$, причем область прибытия первого соответствия совпадает с областью отправления второго.

Композицию соответствий q и p будем обозначать $q(p)$.

Пример. Если q — соответствие, определяющее распределение шоферов по автомашинам, p — соответствие, определяющее распределение автомашин по маршрутам, то соответствие $q(p)$ — соответствие, определяющее распределение водителей по маршрутам.

Операцию композиции можно распространить и на большее число соответствий.

Определение. Пусть X и Y — некоторые множества. Соответствие

$q=(X, Y, Q)$ называется отображением множества X во множество Y , если область определения соответствия совпадает с областью отправления X . Таким образом, каждому элементу x отображение Q ставит в соответствие некоторый элемент подмножества множества Y , называемый образом элемента x .

Пример. Если в примере 2 исключить из рассмотрения шофера c , то получим отображение $Q: X \rightarrow Y$, в котором $X=\{a,b\}$ – множество шоферов; $Y=\{\alpha, \beta, \gamma\}$ – множество автомашин; $Q = \{(a, \alpha), (a, \gamma), (b, \alpha)\}$ – распределение шоферов по автомашинам (рис.2)

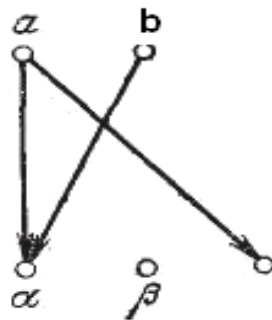


Рис. 2. Геометрическое представление отображения.

Определение. Функцией называется отображение $F: X \rightarrow Y$, если оно однозначно, т. е. если для любого x из X существует единственный элемент y из Y .

Пример. Из одного города в другой можно проехать по железной дороге (ж.д.), автобусом (авт.) или самолетом (сам.). Стоимость билета будет соответственно руб. Стоимость билета можно представить как функцию от вида транспорта. Для этого рассмотрим множества $X=\{\text{ж.д.}, \text{авт.}, \text{сам.}\}$; $Y=\{300, 290, 1000\}$. Функция $F: X \rightarrow Y$ представляется прямым произведением $X \times Y$, которое записывается в виде $f=\{(\text{ж.д.}, 7), (\text{авт.}, 9), (\text{сам.}, 12)\}$.

Определение. Бинарным отношением на множестве X называется отображение множества X во множество Y , если множества X и Y совпадают.

Пример. Пусть X – множество людей. Обозначим через Γ отношение быть ребенком человека x , то есть $y=\Gamma x$ является ребенком человека x . Тогда композиция отношений $\Gamma(\Gamma x)=\Gamma^2 x$ – множество внуков человека x ; $\Gamma^3 x$ – множество правнуков человека x ; обратное отображение $\Gamma^{-1} x$ – множество родителей человека x и т.д.

Изображая людей точками и рисуя стрелки, идущие из x в Γx , по-

лучаем родословное или генеалогическое

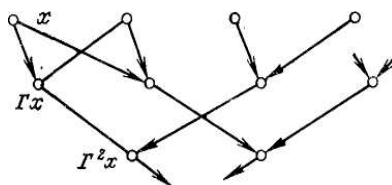


Рис. 6. Генеалогическое дерево.

Пусть задано отношение $\Gamma: X \rightarrow X$. Если элемент x находится в отношении Γ к элементу y , то это удобно записывать в виде $y\Gamma x$.

Рассмотрим некоторые свойства отношений.

Определение. Отношение (X, Γ) называется **рефлексивным**, если $x\Gamma x$ истинно и **антирефлексивным**, если $x\Gamma x$ ложно.

Определение. Отношение (X, Γ) называется **симметричным**, если из $x\Gamma y$ следует $y\Gamma x$, в противном случае - **несимметричным**.

Определение. Отношение (X, Γ) называется **антисимметричным**, если из $x\Gamma y$ и $y\Gamma x$ следует $x=y$.

Определение. Отношение (X, Γ) называется **транзитивным**, если из $x\Gamma y$ и $y\Gamma z$ следует $x\Gamma z$.

Определение. Отношение (X, Γ) называется **отношением эквивалентности**, если оно **рефлексивно, симметрично и транзитивно**, т.е. для любых элементов x, y, z , принадлежащих множеству X , имеет место:

1. $x\Gamma x$;
2. Если $x\Gamma y$, то $y\Gamma x$;
3. Если $x\Gamma y$ и $y\Gamma z$, то $x\Gamma z$.

Примеры.

1. Отношение «быть на одном курсе», определенное на множестве студентов X какого-либо факультета есть отношение эквивалентности;

2. Отношение параллельности на множестве прямых плоскости – отношение эквивалентности;

3. Отношение подобия на множестве треугольников – отношение эквивалентности.

Определение. Подмножество элементов множества X , на котором определено отношение эквивалентности, эквивалентных некоторому элементу, будем называть **классом эквивалентности**.

Отношение эквивалентности разбивает множество X на непересе-

кающиеся подмножества, называемые классами эквивалентности. Объединение всех классов эквивалентности дает множество X .

Пример. Пусть X — множество студентов второго курса, а отношением эквивалентности является отношение «быть в одной группе», тогда группа, в которой учится студент Иванов, будет классом эквивалентности, эквивалентным студенту Иванову.

Отношение эквивалентности разбило исходное множество студентов X на непересекающиеся классы эквивалентности (группы), объединение которых дает полное множество X студентов второго курса.

Рассмотрим отношения порядка, задающие некоторый порядок расположения элементов множества на основе понятий «раньше» и «позже», «больше» и «меньше». Различают отношение нестрогого порядка, для обозначения которого используется символ \leq , и отношение строгого порядка, для которого используется символ $<$.

Определение. Отношением нестрогого порядка на множестве X называют отношение, обладающее следующими свойствами:

1. $x \leq x$ истинно (рефлексивность);
2. Если $x \leq y$ и $y \leq x$ истинно, то $x=y$ (антисимметричность);
3. Если $x \leq y$ и $y \leq z$ истинно, то $x \leq z$ (транзитивность).

Определение. Отношением строгого порядка на множестве X называют отношение, обладающее следующими свойствами:

1. $x < x$ ложно (антирефлексивность);
2. Если $x < y$ истинно, то и $y < x$ ложно (несимметричность);
3. Если $x < y$ и $y < z$ истинно, то $x < z$ (транзитивность).

Определение. Пусть X — множество людей. Если x в чем-то превосходит y , то на множестве X определено **отношение доминирования**.

В этом случае говорят, что x доминирует над y , и пишут $x \gg y$.

Отношение доминирования является:

- антирефлексивным** (нельзя доминировать над самим собой);
- несимметричным** (в каждой паре только один доминирует над другим);
- не является транзитивным**.

В отношении доминирования свойство транзитивности не имеет места. Например, если в соревнованиях команда x победила команду y , а команда y победила команду z , то отсюда еще не следует, что команда x обязательно победит команду z .

Лекция 2. Элементы логики предикатов.

Учебные вопросы.

2.1.Понятие предиката.

2.2.Операции над предикатами.

2.1.Понятие предиката.

В алгебре логики высказывания рассматриваются только с точки зрения их истинности и ложности. Структура и смысл высказываний не учитываются. Однако, в рассуждениях часто используются заключения, существенно зависящие от структуры и смысла используемых в них высказываний.

Например, из истинных высказываний "3 меньше 5" и "5 меньше 7" заключаем, что "3 меньше 7".

Из высказываний "Петр сын Ивана" и "Павел сын Петра" заключаем: "Павел - внук Ивана" и т.д.

Истинность заключения зависит не только от истинности посылок, но и от их смысла. Если изменить смысл посылок, то может оказаться, что заключение будет неверным, несмотря на то, что истинность посылок сохранится.

Так из истинных высказываний "3 меньше 5" и "5 не равно 7" не следует заключение, что "3 меньше 7" (хотя оно само по себе и истинно).

Необходимо обобщение логики высказываний, в рамках которого можно исследовать структуру и смысл высказываний. Таким обобщением является логика предикатов.

Рассмотрим утверждение « x – простое число». Оно не является высказыванием и называется высказывательной формой, которая и является предикатом. В зависимости от x он может иметь логические значения истина или ложь, т.е. является логической функцией переменной x .

Определение. Одноместным предикатом $P(x)$ называется функция переменной x , определенная на некотором множестве X любой природы и принимающая значения из множества $\{И, Л\}$ или $\{1, 0\}$, причем 1 означает истину, 0 – ложь.

Множество X , на котором предикат определен, называется его областью определения.

Определение. Множество всех элементов x , принадлежащих X , на которых $P(x)$ принимает значение истина, называется множеством истинности предиката $P(x)$. Будем обозначать его I_P .

Предикат $P(x)$ еще называется предикатом – свойством, поскольку в нем утверждается, что элемент x обладает свойством $P(x)$.

Пример 1. На множестве натуральных чисел X задан предикат $P(x) = \langle x - \text{четное число} \rangle$. Множество натуральных чисел X – его область определения. Его множество истинности I_P есть множество натуральных чисел, кратных двум. $P(x)$ утверждает, что число x обладает свойством четности.

Пример 2. Предикат $P(x) = \langle x - \text{простое число} \rangle$. Область определения этой функции (предиката) – множество натуральных чисел, а область значений функции – множество $\{И, Л\}$. Предикат $P(x) = \langle x \text{ есть простое число} \rangle$ можно задать в виде следующей таблицы, которую называют **матрицей предиката** или **таблицей истинности предиката**

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
$P(x)$	Л	И	И	Л	И	Л	И	Л	Л	Л	И	Л	И	Л	Л	...

Пример 3. Предиката $G(x) = \langle x - \text{четное натуральное число} \rangle$. Его область определения предиката – множество натуральных чисел, а область значений – множество $\{И, Л\}$. Предикат $G(x)$ можно задать в виде следующей **матрицы**

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
$G(x)$	Л	И	Л	И	Л	И	Л	И	Л	И	Л	И	Л	И	Л	...

Область определения X предиката $P(x)$ может иметь произвольную природу: это может быть числовое множество, множество людей, слонов, цветов, и т.д., т.е. предикат, как функция, не обязательно определен на числовых множествах.

Два предиката $P(x)$ и $Q(x)$, определенные на одном множестве X и имеющие одно и то же множество истинности, считаются **одинаковыми** и в логике предикатов не различаются.

Например, предикаты " x – четное натуральное число" и " x – целое положительное число, делящееся на число 2 без остатка" одинаковы с точки зрения логики предикатов, поскольку имеют одно и то же множество истинности.

Различных предикатов-свойств, определенных на множестве X ,

столько, сколько различных подмножеств имеет множество X . Поэтому если множество X конечно и содержит n элементов, то число его подмножеств равно 2^n и столько же имеется предикатов, определенных на X .

Определение. Предикат называется тождественно-истинным, если его множество истинности совпадает с множеством определения X , и тождественно-ложным, если его множество истинности пусто.

Пример. Предикат $P(x) = "x > 0"$ на множество натуральных чисел- тождественно-истинный.

При той же области определения предикат $R(x) = "x < -2"$ является тождественно-ложным.

Определение. Предикат с областью определения X , множество истинности которого не пусто, но и не совпадает с множеством X , называют выполнимым. Выполнимый предикат на одних элементах множества X принимает значение И, а на других - Л.

Обобщением одноместного предиката является понятие двухместного предиката, выражающие отношения между объектами некоторых множеств.

Определение. Двухместным предикатом $P(x,y)$ называется функция двух переменных $x \in X$ и $y \in Y$, определенная на прямом произведении $X \times Y$. Элементы множеств X, Y называются предметными константами.

Примеры. Двухместные предикаты:

1. $P(x,y) = "x \text{ есть брат } y"$ с областью определения $X \times X$, где X – множество людей;

2. $T(x,y) = "x \parallel y"$, область определения $X \times X$, X – множество прямых на плоскости.

Предикат становится высказыванием, если в нем переменные заменить конкретными значениями из соответствующих множеств.

Пример. Предикат $Q(x,y) = "x > y"$ при $x=3$ и $y=2$ становится истинным высказыванием $Q(3,2) = "3 > 2"$.

Множество истинности двухместного предиката представляет собой подмножество $R_{\text{ист}}$ соответствующего декартового произведения $X \times Y$, причем для всякой пары $(x, y) \in R_{\text{ист}}$ имеет место $P(x, y) = \text{И}$.

Если $R_{\text{ист}} = X \times Y$, то предикат $P(x, y)$ называют тождественно истинным на множествах X и Y . Если $R_{\text{ист}} = \emptyset$, то $P(x, y)$ называют тождественно ложным на множествах X и Y .

Определение. n - местным предикатом называется функция от n

переменных $Q(x_1, x_2, \dots, x_n)$, определенная на декартовом произведении множеств: $X_1 \times X_2 \times \dots \times X_n$ и принимающую значения из множества $\{И, Л\}$.

Булевская функция $f(x_1, x_2, \dots, x_n)$ n переменных - частный случай предиката, когда множества $X_1 = X_2 = \dots = X_n = \{0, 1\}$. Ее область определения прямое произведение $X_1 \times X_2 \times \dots \times X_n$.

2.2. Операции над предикатами.

Логические операции на предикатами.

Определение. Отрицанием предиката $Q(x, y)$, определенного на множествах X, Y , называется предикат, определенный на тех же множествах и обозначаемый $\neg Q(x, y)$, такой, что для любой пары $(x, y) \in X \times Y$ принимает значение ложь, если $Q(x, y)$ принимает значение истина, т.е. если $Q(x, y) = И$, то $\neg Q(x, y) = Л$ и обратно, если $Q(x, y) = Л$, то $\neg Q(x, y) = И$.

Определение. Конъюнкцией предикатов $Q(x, y)$ и $G(x, y)$ называется предикат, обозначаемый $Q(x, y) \& G(x, y)$ и определенный на множествах X, Y , такой, что для любой пары $(x, y) \in X \times Y$ $Q(x, y) \& G(x, y) = И$, если одновременно $Q(x, y) = И$ и $G(x, y) = И$, в остальных случаях $Q(x, y) \& G(x, y) = Л$.

Определение. Дизъюнкцией предикатов $Q(x, y)$ и $G(x, y)$ называется предикат, обозначаемый $Q(x, y) \vee G(x, y)$ и определенный на множествах X, Y , такой, что для любой пары $(x, y) \in X \times Y$ $Q(x, y) \vee G(x, y) = Л$, если одновременно $Q(x, y) = Л$ и $G(x, y) = Л$, в остальных случаях $Q(x, y) \vee G(x, y) = И$.

Определение. Импликацией предикатов $Q(x, y)$ и $G(x, y)$ называется предикат, обозначаемый $Q(x, y) \rightarrow G(x, y)$ и определенный на множествах X, Y , такой, что для любой пары $(x, y) \in X \times Y$ $Q(x, y) \rightarrow G(x, y) = Л$, если одновременно $Q(x, y) = И$ и $G(x, y) = Л$, в остальных случаях $Q(x, y) \rightarrow G(x, y) = И$.

Определение. Эквиваленцией предикатов $Q(x, y)$ и $G(x, y)$ называется предикат, обозначаемый $Q(x, y) \sim G(x, y)$ и определенный на множествах X, Y , такой, что для любой пары $(x, y) \in X \times Y$ $Q(x, y) \sim G(x, y) = И$, если одновременно $Q(x, y)$ и $G(x, y)$ имеют одинаковый логический смысл, в остальных случаях $Q(x, y) \sim G(x, y) = Л$.

Кванторные операции над предикатами.

Рассмотрим две кванторные операции над предикатами, которые одноместный предикат $P(x)$ превращают в высказывание, двухместный предикат – в одноместный, n -местный – в $(n-1)$ -местный предикат, т.е. применение этих операций уменьшает количества свободных переменных в заданном предикате на единицу.

Квантор всеобщности. Пусть $P(x)$ – предикат, причем $x \in X$. Под выражением $\forall x P(x)$ понимается высказывание «Для всех x $P(x)=И$ ». Символ \forall называется квантор всеобщности.

В соответствии со смыслом слова "все" высказывание $\forall x P(x)$ будет истинным тогда и только тогда, когда $P(x)$ – тождественно истинный предикат в области определения X , в остальных случаях это высказывание ложно. Значение предиката $P(x)$ зависит от значения x , а высказывание $\forall x P(x)$ не зависит от переменной x , поэтому ее называют связанной квантором общности \forall .

Если множество X состоит из конечного числа объектов, то квантор общности выражается через конъюнкцию единичных высказываний.

Например, пусть $X = \{x_1, x_2, x_3, x_4\}$, тогда высказывание $\forall x P(x) = P(x_1) \& P(x_2) \& P(x_3) \& P(x_4)$.

Квантор существования. Пусть $P(x)$ – предикат, $x \in X$. Под выражением $\exists x P(x)$ понимается высказывание «Существует x , для которого $P(x)=И$ ». Символ \exists называется квантор существования.

Высказывание $\exists x P(x)$ – ложно, если $P(x)$ – тождественно ложный предикат, в остальных случаях это высказывание истинно.

Если множество X состоит из конечного числа объектов, то квантор существования выражается через дизъюнкцию единичных высказываний.

Например, пусть $X = \{x_1, x_2, x_3, x_4\}$, тогда высказывание $\exists x P(x) = P(x_1) \vee P(x_2) \vee P(x_3) \vee P(x_4)$.

Использование кванторов значительно обогащает выразительные возможности языка логики предикатов, по сравнению с логикой высказываний.

Например, пусть имеются предикаты $P(x) = "x - \text{простое число}"$ и $Q(x) = "x - \text{нечетное число}"$. Тогда запись $\exists x (P(x) \& Q(x))$ означает высказывание "Существуют такие числа, которые являются нечетными и простыми", или "Некоторые нечетные числа простые".

Лекция 3. Применение логики предикатов.

Учебные вопросы.

3.1. Применение логики предикатов для описания отношений.

3.2. Применение логики предикатов для описания функциональных зависимостей и их свойств.

3.1. Применение логики предикатов для описания отношений.

Язык логики предикатов применим как для описания математических понятий и утверждений: аксиом, теорем, доказательств, так и вне математики. С помощью его просто выражаются самые общие понятия. Рассмотрим отношения эквивалентности, порядка и понятие функции.

Отношение эквивалентности или просто эквивалентность является одним из простейших типов бинарных отношений.

Пример. Отношение эквивалентности задает бинарный предикат $K(x,y)$ = «Прямая x параллельна прямой y », определенный на множестве прямых на плоскости.

Это отношение параллельности обладает следующими свойствами:

1. Каждая прямая параллельна сама себе (рефлексивность). Для всех прямых x $K(x,x)$ имеет значение И, что можно записать выражением $\forall x K(x,x)$.

2. Если прямая x параллельна прямой y , то прямая y параллельна прямой x (симметричность) или $\forall x \forall y (K(x,y) \rightarrow K(y,x))$.

3. Если прямая x параллельна прямой y , а прямая y параллельна прямой z , то прямая x параллельна прямой z (транзитивность), что можно записать формулой логики предикатов в виде $\forall x \forall y \forall z ((K(x,y) \& K(y,z)) \rightarrow K(x,z))$.

Всякий предикат, для которого выполняются свойства рефлексивности, симметричности, транзитивности, задает отношение эквивалентности. Условие того, что некоторый предикат $P(x,y)$ задает отношение эквивалентности, записывается формулой логики предикатов:

$\forall x P(x,x) \& \forall x \forall y (P(x,y) \rightarrow P(y,x)) \& \forall x \forall y \forall z ((P(x,y) \& P(y,z)) \rightarrow P(x,z))$.

Отношение порядка (упорядочение) также является бинарным отношением. Они возникают при сравнении чисел по величине, людей по возрасту, событий во времени и т.п. Соответствующие предикаты

описываются словами "больше", "старше", "раньше" и т.п.

Рассмотрим предикат $R(x,y)=\langle x \geq y \rangle$, определенный на множестве натуральных чисел, дополненном нулем.

Отметим некоторые свойства этого предиката.

1. $\forall x R(x,x)$ (рефлексивность).
2. $\forall x \forall y (R(x,y) \& R(y,x) \rightarrow (x=y))$ (антисимметричность).
3. $\forall x \forall y \forall z (R(x,y) \& R(y,z) \rightarrow R(x,z))$ (транзитивность).
4. $\forall x \forall y (R(x,y) \vee R(y,x))$ (свойство дихотомии).

Произвольный предикат $Q(x,y)$ определенный на декартовом квадрате $M \times M$ и обладающий свойствами 1)-4), называется **отношением порядка**, определенным на множестве M .

Рассмотрим предикат $G(x,y)=\langle x > y \rangle$, определенный на множестве действительных чисел.

Отметим некоторые свойства этого предиката.

1. $\forall x \neg G(x,x)$ (антирефлексивность).
2. $\forall x \forall y (G(x,y) \rightarrow \neg G(y,x))$ (антисимметричность).
3. $\forall x \forall y \forall z (G(x,y) \& G(y,z) \rightarrow G(x,z))$ (транзитивность).
4. $\forall x \forall y (G(x,y) \vee G(y,x) \vee (x=y))$ (свойство трихотомии).

3.2. Применение логики предикатов для описания функциональных зависимостей и их свойств.

Функциональные отношения. Функцией называется правило, задающее однозначное отображение некоторого множества M в некоторое множество N . Функцию принято записывать в форме $y = f(x)$.

Рассмотрим случай, когда множества M и N совпадают, и будем говорить об отображении множества M в себя.

Всякой функции $f(x)$, определенной на M и со значениями в M , поставим в соответствие предикат $F(x, y)$, определенный на $M \times M$, считая $F(m, n) = И$, если $n = f(m)$, и $F(m, n) = Л$, если $n \neq f(m)$. Задание предиката $F(x, y)$ однозначно определяет функцию $y = f(x)$. Эти предикаты называют **функциональными** и должны обладать свойствами:

1. $\forall x \exists y P(x,y)$, т.е. каждому x обязательно соответствует элемент y ;
2. $\forall x \forall y \forall z ((P(x,y) \& P(x,z)) \rightarrow (y=z))$, т.е. каждому x соответствует только один элемент y .

Приведем несколько примеров выражения математических предложений на языке логики предикатов. При этом мы будем использовать обозначения множеств, функций и предикатов, широко принятые в ма-

тематике:

N - множество натуральных чисел;

R - множество действительных чисел;

$|x|$ - функция "абсолютная величина числа **x**";

x<**y** - предикат "x меньше y"

и т.д.

Пример 1. Утверждение "Всякое натуральное число больше нуля" можно записать в виде импликации с квантором общности в виде следующей формулы: $\forall x((x \in N) \rightarrow (x > 0))$.

Пример 2. Определение предела числовой последовательности. Число **A** называется пределом бесконечной числовой последовательности $\{a_n\} = a_1, a_2, \dots, a_n, \dots$, если для всякого $\varepsilon > 0$ существует такое натуральное n_ε , что для всякого номера **n**, если $n > n_\varepsilon$, то $|a_n - A| < \varepsilon$. На языке логики предикатов это определение может быть записано в виде формулы:

$$\forall \varepsilon_{>0} \exists n_\varepsilon \in N \forall n \in N ((n > n_\varepsilon) \rightarrow (|a_n - A| < \varepsilon))$$

Построим отрицание этой формулы, то есть утверждение "Число **A** не является пределом бесконечной числовой последовательности $\{a_n\}$ ":

$$\begin{aligned} & \neg(\forall \varepsilon_{>0} \exists n_\varepsilon \in N \forall n \in N ((n > n_\varepsilon) \rightarrow (|a_n - A| < \varepsilon))) \equiv \\ & \equiv \exists \varepsilon_{>0} \forall n_\varepsilon \in N \exists n \in N (\neg((n > n_\varepsilon) \rightarrow (|a_n - A| < \varepsilon))) \equiv \\ & \equiv \exists \varepsilon_{>0} \forall n_\varepsilon \in N \exists n \in N ((n > n_\varepsilon) \& \neg(|a_n - A| < \varepsilon)) \equiv \\ & \equiv \exists \varepsilon_{>0} \forall n_\varepsilon \in N \exists n \in N ((n > n_\varepsilon) \& (|a_n - A| \geq \varepsilon)). \end{aligned}$$

Эта цепочка равносильностей построена на основе возможности переноса отрицания через квантор с заменой квантора на двойственный, преобразования формулы логики высказываний: выражение отрицания импликации через конъюнкцию ($\neg(X \rightarrow Y) \equiv (X \& \neg Y)$), и на замене отношения "<" его отрицанием – "≥".

Таким образом, утверждение "Число **A** не является пределом бесконечной числовой последовательности $\{a_n\}$ " раскрывается так: "Существует положительное число ε такое, что для всякого числа n_ε найдется такое **n**, большее n_ε , что $|a_n - A| \geq \varepsilon$ ".

Если последнее утверждение справедливо для любого числа **A**, то

последовательность $\{a_n\}$ не имеет предела.

Пример 3. Запишем на языке логики предикатов определение непрерывности функции в точке.

Пусть область определения M_f функции f - отрезок, содержащий точку x_0 . В этом случае определение непрерывности функции f в точке x_0 может быть сформулировано следующим образом:

Функция f непрерывна в точке x_0 , если для всякого положительного числа ε существует положительное число δ такое, что для всякого x из области определения M_f функции f из неравенства $|x - x_0| < \delta$ следует неравенство $|f(x) - f(x_0)| < \varepsilon$.

На языке логики предикатов это определение запишется:

$$\forall \varepsilon > 0 \exists \delta > 0 \forall x \in M_f ((|x - x_0| < \delta) \rightarrow (|f(x) - f(x_0)| < \varepsilon))$$

Построим отрицание этого определения, то есть предложение "функция f не является непрерывной в точке x_0 ". Возьмем отрицание последней формулы и выполним ряд эквивалентных преобразований:

$$\begin{aligned} & \neg(\forall \varepsilon > 0 \exists \delta > 0 \forall x \in M_f ((|x - x_0| < \delta) \rightarrow (|f(x) - f(x_0)| < \varepsilon))) \equiv \\ & \equiv \exists \varepsilon > 0 \forall \delta > 0 \exists x \in M_f (\neg((|x - x_0| < \delta) \rightarrow (|f(x) - f(x_0)| < \varepsilon))) \equiv \\ & \equiv \exists \varepsilon > 0 \forall \delta > 0 \exists x \in M_f ((|x - x_0| < \delta) \& \neg(|f(x) - f(x_0)| < \varepsilon)) \equiv \\ & \equiv \exists \varepsilon > 0 \forall \delta > 0 \exists x \in M_f ((|x - x_0| < \delta) \& (|f(x) - f(x_0)| \geq \varepsilon)). \end{aligned}$$

Эта цепочка равносильностей построена на основе переносе отрицания через квантор с заменой квантора на двойственный, применения тождества $\neg(X \rightarrow Y) \equiv (X \& \neg Y)$ и на замене отношения "<" его отрицанием – "≥".

Таким образом, утверждение "Функция f не является непрерывной в точке x_0 " раскрывается так: "Существует положительное число ε такое, что для всякого положительного δ найдется x из области определения функции такое, что $|x - x_0| < \delta$ и (вместе с тем) $|f(x) - f(x_0)| \geq \varepsilon$ ".

Лекция 4. Формулы логики предикатов.

Учебные вопросы.

4.1. Кванторные операции над многоместными предикатами.

4.2. Формулы логики предикатов. Преобразование формул.

4.1. Кванторные операции над многоместными предикатами.

Пусть дан двухместный предикат $Q(x,y)$ - определенный на множествах X, Y . Квантор общности и квантор существования можно применять к нему по любой из переменных как по x , так и по y : $\forall xQ(x,y)$, $\exists xQ(x,y)$, $\forall yQ(x,y)$, $\exists yQ(x,y)$.

Переменная, по которой применяется квантор, называется **связанной**, а переменная, не находящаяся под действием предиката, – **свободной**. Все четыре приведенные выражения являются одноместными предикатами от соответствующей свободной переменной.

Пример 1. Пусть предикат $Q(x,y)$ определен на конечных множествах: $X=\{a_1, a_2, a_3, a_4, a_5\}$, $Y=\{b_1, b_2, b_3, b_4, b_5, b_6\}$ и имеет таблицу истинности

X	Y					
	b_1	b_2	b_3	b_4	b_5	b_6
a_1	И	И	Л	Л	И	Л
a_2	Л	И	Л	Л	И	Л
a_3	И	И	Л	Л	И	И
a_4	Л	И	Л	Л	И	Л
a_5	И	И	Л	Л	И	Л

Рассмотрим предикат $F_1(y) = \forall xQ(x,y)$. Он истинен для тех b_i из Y , для которых заданный предикат $Q(x,b_i)$ истинен для всех $x \in X$, то есть таблица истинности этого предиката имеет вид:

$\forall xQ(x,y)$	Y					
	b_1	b_2	b_3	b_4	b_5	b_6
	Л	И	Л	Л	И	Л

Рассмотрим предикат $F_2(y) = \exists x Q(x, y)$. Он истинен для тех b_i из Y , для которых предикат $Q(x, b_i)$ истинен хотя бы для одного $x \in X$, то есть таблица истинности этого предиката имеет вид:

$\exists x Q(x, y)$	Y					
	b_1	b_2	b_3	b_4	b_5	b_6
	И	И	Л	Л	И	И

Предикат $\Phi_1(x) = \forall y Q(x, y)$ истинен для тех a_i из X , для которых предикат $Q(x, b_i)$ истинен для всех $y \in Y$, т.е. таблица истинности этого предиката имеет вид:

X	$\forall y Q(x, y)$
a_1	Л
a_2	Л
a_3	Л
a_4	Л
a_5	Л

Предикат $\Phi_2(x) = \exists y Q(x, y)$ истинен для тех a_i из X , для которых предикат $Q(x, b_i)$ истинен хотя бы для одного $y \in Y$, то есть таблица истинности этого предиката имеет вид:

X	$\exists y Q(x, y)$
a_1	И
a_2	И
a_3	И
a_4	И
a_5	И

Во всех случаях применение квантора по одной из переменных двухместного предиката превращает его в одноместный. В общем же случае n -местного предиката применение квантора по одной из переменных превращает его в $(n-1)$ -местный предикат. Так, если к свободным переменным x и y предикатов $\forall x Q(x, y)$, $\exists x Q(x, y)$, $\forall y Q(x, y)$,

$\exists yQ(x,y)$ применить кванторы общности и существования, то получим восемь высказываний, представленных в таблице:

Высказывание	Значение истинности
$\forall y\forall xQ(x,y)$	Л
$\exists y\forall xQ(x,y)$	И
$\forall y\exists xQ(x,y)$	Л
$\exists y\exists xQ(x,y)$	И
$\forall x\forall yQ(x,y)$	Л
$\exists x\forall yQ(x,y)$	Л
$\forall x\exists yQ(x,y)$	И
$\exists x\exists yQ(x,y)$	И

Имеют место тождества : $\forall x\forall yQ(x,y) \equiv \forall y\forall x Q(x,y)$; $\exists x\exists yQ(x,y) \equiv \exists y\exists xQ(x,y)$, т.е. кванторы одного вида можно менять местами.

Но порядок следования разноименных кванторов существенен. Высказывания $\forall x\exists yQ(x,y)$ и $\exists y\forall xQ(x,y)$, вообще говоря, не эквивалентны.

Для предиката $Q(x,y)$ высказывание $\forall x\exists yQ(x,y)=И$ означает, что во всякой строке таблицы, задающей $Q(x,y)$, встречается по меньшей мере один раз значение И.

Высказывание $\exists y\forall xQ(x,y)=И$ тогда и только тогда, когда имеется по меньшей мере один столбец, содержащий только значение И. Как видно, если выполняется второе требование, то автоматически выполняется и первое.

Таким образом, если $\exists y\forall xQ(x,y)=И$, то и $\forall x\exists yQ(x,y)=И$ (каков бы ни был предикат $Q(x,y)$). Следовательно, имеет место импликация: $\exists y\forall xQ(x,y) \rightarrow \forall x\exists yQ(x,y)$, но обратная импликация, вообще говоря, не имеет места.

Пример 2. Рассмотрим предикат натуральных чисел: $D(x,y) = \text{«}x \text{ делит } y \text{ без остатка»}$, в котором x,y – натуральные числа. Тогда утверждение $\forall x\exists yD(x,y)$ означает: "Для всякого числа x существует такое число y , что x делит y без остатка", что очевидно истинно. В то же время утверждение $\exists y\forall xQ(x,y)$, которое звучит так: "Существует такое число y , что любое x делит y ", является ложным.

4.2. Формулы логики предикатов. Преобразование формул.

Дадим определение формулы логики предикатов. Будем использовать следующие символы:

1. $\neg, \&, \oplus, \vee, \rightarrow, \sim, \forall, \exists$ – символы операций в логике предикатов;
2. $x_1, x_2, x_3, \dots, x_n, \dots, y_1, y_2, y_3, \dots, y_m, \dots$ - предметные переменные;
3. $a_1, a_2, a_3, \dots, a_n, \dots, b_1, b_2, b_3, \dots, b_m, \dots$ - предметные постоянные;
4. $P(x_1, x_2, \dots, x_i, \dots, x_n)$ – n -местные предикаты ($n = 0, 1, 2, \dots$);
5. $()$ - скобки.

Определение.

1. Всякая формула логики высказываний есть формула логики предикатов.
2. Всякий n -местный ($n = 0, 1, 2, \dots$) предикат $P(x_1, x_2, \dots, x_i, \dots, x_n)$ - есть формула логики предикатов.
3. Результат подстановки в предикат $P(x_1, x_2, \dots, x_i, \dots, x_n)$ вместо всех вхождений предметной переменной x_i предметной постоянной a_i $P(x_1, x_2, \dots, a_i, \dots, x_n)$ есть формула логики предикатов.
4. Если A - формула логики предикатов, то результат применения кванторов общности \forall и существования \exists к формуле A по любой переменной x_i , входящей в формулу A , приводит к формулам логики предикатов $\forall x_i A$ или $\exists x_i A$.
5. Если A и B - формулы логики предикатов, то следующие выражения также будут формулами логики предикатов: $\neg(A)$, $(A \& B)$, $(A \oplus B)$, $(A \vee B)$, $(A \supset B)$, $(A \sim B)$.
6. Формулами логики предикатов являются только те выражения, которые следуют из пп. 1 - 5.

В основе логики предикатов лежит ряд законов для преобразования формул, содержащих предикаты, операции логики высказываний и кванторы.

В логике предикатов имеют место равносильности:

$$\forall x_i P(x_1, \dots, x_i, \dots, x_n) \& \forall x_i Q(x_1, \dots, x_i, \dots, x_n) \equiv \forall x_i (P(x_1, \dots, x_i, \dots, x_n) \& Q(x_1, \dots, x_i, \dots, x_n))$$

;

$$\exists x_i P(x_1, \dots, x_i, \dots, x_n) \vee \exists x_i Q(x_1, \dots, x_i, \dots, x_n) \equiv \exists x_i (P(x_1, \dots, x_i, \dots, x_n) \vee Q(x_1, \dots, x_i, \dots, x_n)).$$

Следующие равносильности позволяют сводить квантор существования к квантору общности, и наоборот:

$$\neg \exists x_i P(x_1, \dots, x_i, \dots, x_n) \equiv \forall x_i \neg P(x_1, \dots, x_i, \dots, x_n);$$
$$\neg \forall x_i P(x_1, \dots, x_i, \dots, x_n) \equiv \exists x_i \neg P(x_1, \dots, x_i, \dots, x_n).$$

Эти равносильности являются обобщениями законов Де Моргана, к которым они просто сводятся в случае конечности множеств, на которых определен предикат $P(x_1, \dots, x_i, \dots, x_n)$.

Применяя отрицание к обеим частям предыдущих эквивалентностей и учитывая, что двойное отрицание высказывания равносильно ему самому, имеем:

$$\begin{aligned}\exists x_i P(x_1, \dots, x_i, \dots, x_n) &\equiv \neg \forall x_i \neg P(x_1, \dots, x_i, \dots, x_n); \\ \forall x_i P(x_1, \dots, x_i, \dots, x_n) &\equiv \neg x_i \neg P(x_1, \dots, x_i, \dots, x_n).\end{aligned}$$

Следует заметить, что квантор общности нельзя выносить за скобки, если выражения объединены знаком дизъюнкции, то есть формулы

$$\begin{aligned}\forall x_i (P(x_1, \dots, x_i, \dots, x_n) \vee Q(x_1, \dots, x_i, \dots, x_n)) \text{ и} \\ \forall x_i P(x_1, \dots, x_i, \dots, x_n) \vee \forall x_i Q(x_1, \dots, x_i, \dots, x_n)\end{aligned}$$

- не равносильны.

Но в то же время, если первая формула истинна, то, очевидно, истинна и вторая. Поэтому, вторая формула является логическим следствием первой:

$$\forall x_i (P(x_1, \dots, x_i, \dots, x_n) \vee Q(x_1, \dots, x_i, \dots, x_n)) \rightarrow \forall x_i P(x_1, \dots, x_i, \dots, x_n) \vee \forall x_i Q(x_1, \dots, x_i, \dots, x_n)$$

Аналогично имеет место следствие:

$$\exists x_i (P(x_1, \dots, x_i, \dots, x_n) \& Q(x_1, \dots, x_i, \dots, x_n)) \rightarrow \exists x_i P(x_1, \dots, x_i, \dots, x_n) \& \exists x_i Q(x_1, \dots, x_i, \dots, x_n),$$

но знак импликации опять-таки нельзя заменить знаком равносильности.

Имеется, однако, возможность выносить квантор общности и квантор существования за скобки и в этих случаях. А именно, пусть B обозначает формулу, не зависящую от x_i (или не содержащую x_i свободно). Тогда имеют место следующие равносильности:

$$\begin{aligned}\forall x_i P(x_1, \dots, x_i, \dots, x_n) \vee B &\equiv \forall x_i (P(x_1, \dots, x_i, \dots, x_n) \vee B), \\ \exists x_i P(x_1, \dots, x_i, \dots, x_n) \& B &\equiv \exists x_i (P(x_1, \dots, x_i, \dots, x_n) \& B).\end{aligned}$$

Следует отметить, что обозначение связанной переменной в предикате произвольно и что формулы $\forall t P(x_1, \dots, t, \dots, x_n)$ и $\forall s P(x_1, \dots, s, \dots, x_n)$ – равносильны. Поэтому можно переименовать связанные переменные в формуле так, чтобы все связанные переменные были различными и. выносить кванторы за скобки:

$$\begin{aligned}\forall x_i P(x_1, \dots, x_i, \dots, x_n) \vee \forall x_i Q(x_1, \dots, x_i, \dots, x_m) &\equiv \\ \forall x_i P(x_1, \dots, x_i, \dots, x_n) \vee \forall t Q(x_1, \dots, t, \dots, x_m) &\equiv \forall x_i \forall t (P(x_1, \dots, x_i, \dots, x_n) \vee \\ Q(x_1, \dots, t, \dots, x_m)),\end{aligned}$$

то есть

$$\forall x_i P(x_1, \dots, x_i, \dots, x_n) \vee \forall x_i Q(x_1, \dots, x_i, \dots, x_m) \equiv \forall x_i \forall t (P(x_1, \dots, x_i, \dots, x_n) \vee Q(x_1, \dots, t, \dots, x_m))$$

$\vee Q(x_1, \dots, t, \dots, x_m)),$

Аналогично:

$$\begin{aligned} & \exists x_i P(x_1, \dots, x_i, \dots, x_n) \& \exists x_i Q(x_1, \dots, x_i, \dots, x_m) \equiv \\ & \equiv \exists x_i \exists t (P(x_1, \dots, x_i, \dots, x_n) \& Q(x_1, \dots, t, \dots, x_m)). \end{aligned}$$

Приведенные равносильности позволяют любые формулы логики предикатов сводить к равносильным им формулам специального вида, называемым предваренными нормальными формами. Это формулы, у которых все кванторы вынесены вперед. Например, формула $\exists x \exists y (P(x) \& Q(y))$ записана в предваренной нормальной форме.

Общий вид предваренной нормальной формы $K_1 x_1 K_2 x_2 \dots K_n x_n A(x_1, x_2, \dots, x_n)$, где K_1, K_2, \dots, K_n – кванторы общности или существования в любом порядке, $A(x_1, x_2, \dots, x_n)$ – формула логики предикатов, содержащая предикаты от переменных x_1, x_2, \dots, x_n (может быть еще и от других переменных), связанные только операциями логики высказываний.

Рассмотрим пример приведения формулы логики предикатов к предваренной нормальной форме. Пусть дана формула $\exists x \forall y P(x, y) \vee \neg \forall x \exists y Q(x, y)$.

Сначала с помощью законов Де Моргана можно вынести кванторы за знак отрицания $\exists x \forall y P(x, y) \vee \neg \forall x \exists y Q(x, y) \equiv \exists x \forall y P(x, y) \vee \exists x \forall y \neg Q(x, y)$.

Далее можно вынести квантор существования за скобки:

$$\exists x \forall y P(x, y) \vee \exists x \forall y \neg Q(x, y) \equiv \exists x (\forall y P(x, y) \vee \forall y \neg Q(x, y)).$$

Переименовывая связанную переменную y в формуле $\forall y \neg Q(x, y)$, получим:

$$\exists x (\forall y P(x, y) \vee \forall y \neg Q(x, y)) \equiv \exists x (\forall y P(x, y) \vee \forall z \neg Q(x, z)).$$

Выносим последовательно кванторы общности за скобки:

$$\exists x (\forall y P(x, y) \vee \forall z \neg Q(x, z)) \equiv \exists x \forall y (P(x, y) \vee \forall z \neg Q(x, z)) \equiv \exists x \forall y \forall z (P(x, y) \vee \neg Q(x, z)).$$

Последняя формула представлена в предваренной нормальной форме.

Лекция 5. Правила комбинаторики. Размещения.

Учебные вопросы.

5.1.Правило суммы и произведения.

5.2.Размещения с повторениями.

5.3.Размещения без повторений.

5.1. Правило суммы и произведения.

Правило суммы. Если A и B – конечные непересекающиеся множества, т.е. $A \cap B = \emptyset$, число элементов множества A $|A|=m$, а множества B – $|B|=n$, тогда число элементов множества $A \cup B$ $|A \cup B|=m+n$.

Доказательство. Множества A , B не имеют общих элементов, поэтому их объединение, являющееся множеством $A \cup B$, содержит $m+n$ элементов.

Следствие. Пусть X_1, X_2, \dots, X_n – попарно непересекающиеся множества, т.е. $X_i \cap X_j = \emptyset$, где $i \neq j$, тогда

$$\left| \bigcup_{i=1}^k X_i \right| = \sum_{i=1}^k |X_i|,$$

т.е. количество элементов объединения множеств X_1, X_2, \dots, X_n равно сумме элементов этих множеств.

Правило произведения. Если A и B – конечные множества, число элементов множества A $|A|=m$, а число элементов множества B $|B|=n$, тогда число элементов множества $A \times B$ $|A \times B|=m \cdot n$.

Доказательство. Элемент $a \in A$ можно выбрать m способами и для каждого такого выбора элемент $b \in B$ можно выбрать n способами, тогда выбор пары $(a, b) \in A \times B$ в указанном порядке можно осуществить $|A \times B|=m \cdot n$ способами. В этом случае говорят, что выбор элементов множества A не зависит от способа выбора элементов множества B .

Следствие. Пусть теперь X_1, X_2, \dots, X_k – произвольные множества, $|X_i|=n_i$, $i=1, k$. Тогда $|X_1 \times X_2 \times \dots \times X_k|=n_1 \cdot n_2 \cdot \dots \cdot n_k$. Элементами множества $X_1 \times X_2 \times \dots \times X_k$ являются комбинации (x_1, x_2, \dots, x_k) , в которых $x_i \in X_i$, $i=1, 2, \dots, k$.

Задача. Найти число маршрутов из пункта М в пункт N через пункт К. Из М в К введут 5 дорог, из К в N — 3 дороги.



Решение. Введем два множества: $S = \{s_1, s_2, s_3, s_4, s_5\}$ — дороги из М в К, $T = \{t_1, t_2, t_3\}$ — дороги из К в N. Тогда дорогу из М в N можно представить парой (s_i, s_j) , где $i = 1, 2, 3, 4, 5$; $j = 1, 2, 3$. Значит $S \times T$ — это множество всех дорог из М в N, количество которых равно $|S \times T| = 5 \cdot 3 = 15$.

5.2. Размещения с повторениями.

Определение. Имеются предметы n различных видов a_1, a_2, \dots, a_n . Из них составляют всевозможные расстановки длины k . Две расстановки считаются различными, если они отличаются друг от друга или видом входящих в них предметов, или порядком этих предметов. Такие расстановки называются размещениями с повторениями из n элементов по k (элементы одного вида могут повторяться).

Пример. $a_3 a_1 a_1 a_2 a_1 a_3$ - расстановка длины 6.

Теорема. Число различных размещений с повторениями из n элементов по k $N = n^k$.

Доказательство. При составлении указанных расстановок длины k из n предметов на каждое место можно поставить предмет любого вида. Введем множества X_1, X_2, \dots, X_k такие, что $X_1 = X_2 = \dots = X_k = \{a_1, a_2, \dots, a_n\}$. Тогда все размещения с повторениями составят множество $X_1 \times X_2 \times \dots \times X_k$. По правилу произведения общее число размещений с повторениями из n по k равно $|X_1 \times X_2 \times \dots \times X_k| = n \cdot n \cdot \dots \cdot n = n^k$.

Задача. Найти количество всех пятизначных телефонных номеров.

Решение. Введем пять множеств: $X_1 = \{1, \dots, 9\}$, $X_2 = X_3 = X_4 = X_5 = \{0, 1, \dots, 9\}$. Тогда все пятизначные телефонные номера составят прямое произведение указанных множеств $X_1 \times X_2 \times X_3 \times X_4 \times X_5$. Согласно правилу произведения, количество элементов в множестве $X_1 \times X_2 \times X_3 \times X_4 \times X_5$ равно $9 \cdot 10 \cdot 10 \cdot 10 \cdot 10 = 90000$.

5.3. Размещения без повторений.

Определение. Имеются предметы n различных видов a_1, a_2, \dots, a_n . Из них составляют всевозможные расстановки длины k . Две расстановки считаются различными, если они отличаются друг от друга видом входящих в них предметов или порядком этих предметов. Такие расстановки называются размещениями без повторений из n элементов по k . Их количество обозначается символом A_n^k .

Теорема. Число различных размещений без повторений из n элементов по k

$$A_n^k = n(n-1)\dots(n-k+1) = n!/(n-k)!.$$

Доказательство. При составлении данных расстановок на первое место можно поставить любой из имеющихся n предметов. На второе место теперь можно поставить только любой из $(n - 1)$ оставшихся. И, наконец, на k -е место – любой из $(n - k + 1)$ оставшихся предметов. По правилу произведения общее число размещений без повторений из n по k равно $A_n^k = n(n-1)\dots(n-k+1) = n!/(n-k)!$. По определению $n! = n(n-1)(n-2)\dots 1$, $1! = 1$ и $0! = 1$.

Задача. В хоккейном турнире участвуют 17 команд. Разыгрываются золотые, серебряные и бронзовые медали. Сколькими способами могут быть распределены медали?

Решение. 17 команд претендуют на 3 места. Тогда тройку призеров можно выбрать способами $A_{17}^3 = 17 \cdot 16 \cdot 15 = 4080$.

Лекция 6. Перестановки и сочетания без повторений.

Учебные вопросы.

6.1. Перестановки без повторений.

6.2. Сочетания без повторений.

6.1. Перестановки без повторений.

Определение. Имеются предметы n различных видов a_1, a_2, \dots, a_n . Из них составляют всевозможные расстановки длины n . Две расстановки считаются различными, если они отличаются друг от друга порядком этих предметов. Такие расстановки называются перестановками из n элементов по n . Их количество обозначается символом P_n .

Теорема. Число различных перестановок из n элементов $P_n = n!$.

Доказательство. Число перестановок из n элементов равно числу размещений без повторений из n элементов по n элементов, поэтому

$$P_n = A_n^n = \frac{n!}{(n-n)!} = \frac{n!}{0!} = n!$$

Перестановки $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ элементов $1, 2, \dots, n$ записываются и в матричной форме

$$\pi = \begin{pmatrix} 1 & 2 & \dots & n \\ \pi_1 & \pi_2 & \dots & \pi_n \end{pmatrix},$$

где первая строка – порядковые номера $1, 2, \dots, n$ позиций элементов в перестановке; нижняя строка – тот же набор чисел $1, 2, \dots, n$, взятых в каком-либо порядке; π_i – номер элемента на i -ом месте перестановки.

Порядок столбцов в перестановках, записанных в матричной форме, не является существенным, так как в этом случае номер позиции каждого элемента в перестановке указывается явно в верхней строке. Например, перестановка $(3, 2, 4, 1)$ может быть записана в одном из видов:

$$\begin{pmatrix} 1234 \\ 3241 \end{pmatrix}, \begin{pmatrix} 3142 \\ 4312 \end{pmatrix}, \begin{pmatrix} 2143 \\ 2314 \end{pmatrix}.$$

Задача о ладьях. Сколькими способами можно расположить на шахматной доске 8 ладей, чтобы они «не били» друг друга?

Решение. Условие «не могли бить» означает, что на каждой горизонтали и вертикали доски может стоять лишь одна ладья. Ввиду этого, каждому расположению ладей на доске соответствует перестановка

$$\pi = \begin{pmatrix} 1 & 2 & \dots & 8 \\ \pi_1 & \pi_2 & \dots & \pi_8 \end{pmatrix}$$

Верхняя строка – номер горизонталей, нижняя – номер вертикалей, пересечение которых определяет положение ладей на доске. Следовательно, число расстановок ладей равно числу перестановок $P_8=8! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 = 40320$.

6.2. Сочетания без повторений.

В тех случаях, когда нас не интересует порядок элементов в расстановке, а интересует лишь ее состав, то говорят о сочетаниях.

Определение. Имеются предметы n различных видов a_1, a_2, \dots, a_n . Из них составляют всевозможные расстановки длины k . Две расстановки считаются различными, если они отличаются друг от друга видом входящих в них предметов, но не порядком этих предметов. Такие расстановки называются сочетаниями без повторений из n элементов по k . Их количество обозначается символом C_n^k .

Теорема. Число различных сочетаний из n элементов по k

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Доказательство. Составим все сочетания из n по k . Дополним составленные сочетания возможными перестановками каждого сочетания. В результате получатся все расстановки, отличающиеся либо составом, либо порядком, т.е. размещения без повторений из n по k . Каждое сочетание дает $k!$ перестановок из k элементов, тогда по правилу произведения можно записать

$$C_n^k \cdot k! = A_n^k.$$

Отсюда следует, что

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Задача о прямоугольниках. Сколько различных прямоугольников с целыми размерами можно вырезать из клеток доски, размер которой $m \times n$?

1	2	3	...	n
2				
3				
\vdots				
m				

Решение. Прямоугольник однозначно определяется положением его сторон. Горизонтальные стороны могут занимать любое из $(m + 1)$ положений, а вертикальные - любое из $(n + 1)$ положений. Тогда число способов выбора горизонтальных сторон равно C_{m+1}^2 . Вертикальные стороны можно выбрать C_{n+1}^2 способами. По правилу произведения количество прямоугольников

$$C_{m+1}^2 \cdot C_{n+1}^2$$

Лекция 7. Графы.

Учебные вопросы:

7.1. Основные понятия и определения.

7.2. Элементы графов.

7.3. Представление графов в ЭВМ.

7.1. Основные понятия и определения.

Определение. Графом $G(V, E)$ называется совокупность двух множеств – непустого множества $V = \{v_1, v_2, \dots, v_n\}$, называемого множеством вершин и множества $E = \{e_1, e_2, \dots, e_k\}$ неупорядоченных и упорядоченных пар (v_i, v_j) вершин множества V . Неупорядоченные пары вершин называются ребрами, а упорядоченные – дугами.

Определение. Граф, содержащий одни ребра, называется неориентированным, а одни дуги – ориентированным или орграфом.

Определение. Ребро (дуга), начинающееся и заканчивающееся в одной и той же вершине, называется петлей.

Определение. Два и более ребра (две и более дуги одного направления), соединяющие две вершины графа, называются кратными.

Обычно граф изображают диаграммой: вершины – точками (или кружками), ребра – линиями.

Пример.

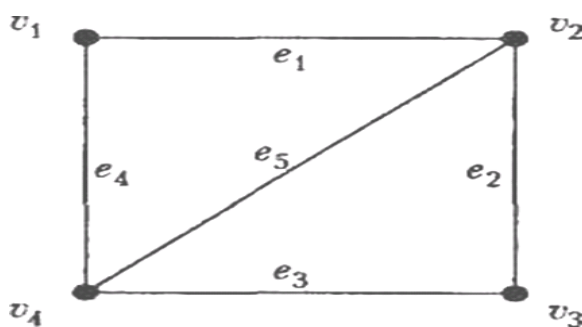


Рис.1. Диаграмма неориентированного графа.

Определение. Вершины v_1, v_2 называются инцидентными соединяющему их ребру $e = (v_1, v_2)$, т.е. вершина v_1 и ребро e инцидентны, вершина v_2 и ребро e также инцидентны.

Определение. Два ребра, инцидентные одной вершине, называ-

ются смежными. Две вершины, инцидентные одному ребру, также называются смежными. Множество вершин, смежных с вершиной v , называется множеством смежности вершины v и обозначается $\Gamma(v)$.

Определение. Граф называется псевдографом, если в нем допускаются петли и кратные ребра, т. е. две вершины могут быть соединены более чем одним ребром. Псевдограф без петель называется мультиграфом.

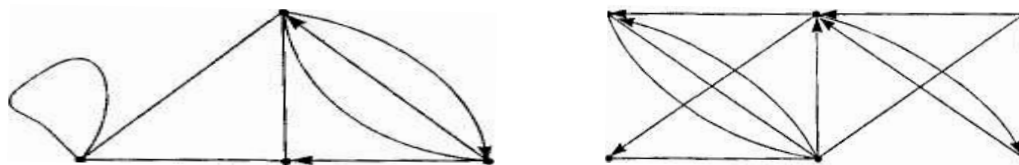


Рис. 2. Псевдограф (слева) и мультиграф (справа).

Определение. Подграфом графа $G(V, E)$ называется такой граф $\Gamma = (V_1, E_1)$, что $V_1 \subset V$ и $E_1 \subset E$.

Определение. Неориентированный граф называется простым, если он не имеет петель и любая пара вершин соединена не более чем одним ребром.

Определение. Простой граф называется полным, если каждая пара вершин соединена ребром.

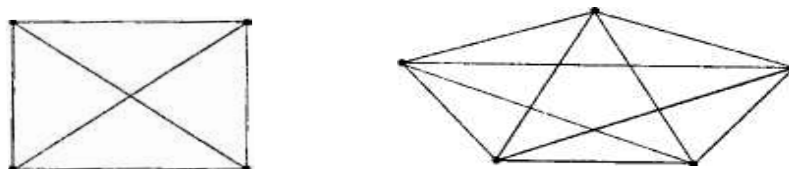


Рис. 3. Полные неориентированные графы.

Определение. Дополнением простого G графа называется граф G_1 , имеющий те же вершины, а его ребра являются дополнением G до полного графа (рис. 4).



Рис. 4. Исходный граф слева и его дополнение справа.

Определение. Степенью (валентностью) вершины графа называется количество ребер, инцидентных данной вершине. Вершина графа,

имеющая степень 0, называется изолированной, а если степень ее равна 1, то такая вершина называется висячей.

Теорема Эйлера. Сумма степеней вершин графа равна удвоенному количеству ребер.

Доказательство. При подсчете суммы степеней вершин каждое ребро учитывается дважды.

Определение. Граф $G(V,E)$ называется связным, если для любой пары его вершин существует соединяющий их путь.

Определение. Связный орграф называется сильно связным. Орграф называется слабо связным, если соответствующий ему неориентированный граф (игнорируется ориентация ребер) связный (рис. 5)



Рис. 5. Слева – слабо связный, справа – сильно связный орграфы.

7.2.Элементы графов.

Определение. Путем или маршрутом на графе $G(V,E)$ называется последовательность его вершин и ребер $v_1e_1v_2e_2v_3...v_ne_nv_{n+1}$, в которой любые два соседних элемента инцидентны.

- Маршрут называется цепью, если все его ребра различные.
- Маршрут называется простой цепью, если все его вершины, а значит и ребра, различные.
- Маршрут называется замкнутым, если исходная и конечная вершины совпадают.
- Замкнутая цепь называется циклом. Простая замкнутая цепь называется простым циклом.
- Гамильтоновой цепью называется простая цепь, содержащая все вершины графа.

• Гамильтоновым циклом называется простой цикл, содержащий все вершины графа.

На рис.6:

1. v_1, v_3, v_1, v_4 – путь, но не цепь.
2. $v_1, v_3, v_5, v_2, v_3, v_4$ – цепь, но не простая цепь.
3. v_1, v_4, v_3, v_2, v_5 – простая цепь.
4. $v_1, v_3, v_5, v_2, v_3, v_4, v_1$ – цикл, но не простой цикл.
5. v_1, v_3, v_4 – простой цикл.

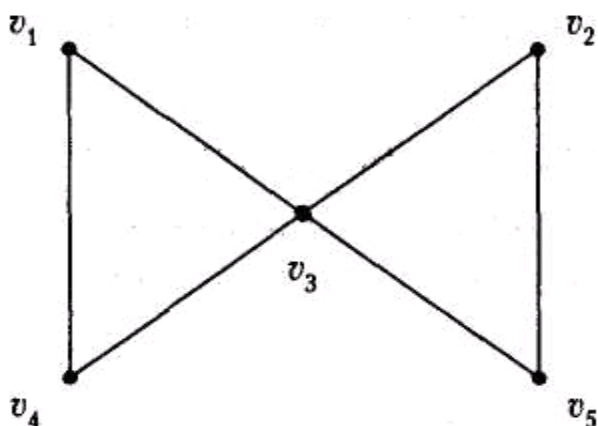


Рис.6. Элементы графа.

Определение. Длиной маршрута называется количество ребер в нем (с повторениями). Если маршрут $M = v_1 e_1 v_2 e_2 v_3 \dots v_n e_n v_{n+1}$, то длина M равна n и обозначается $|M|=n$.

Определение. Расстоянием между вершинами u и v (обозначается $d(u,v)$) называется длина кратчайшей цепи, соединяющей эти вершины, а сама кратчайшая цепь называется геодезической.

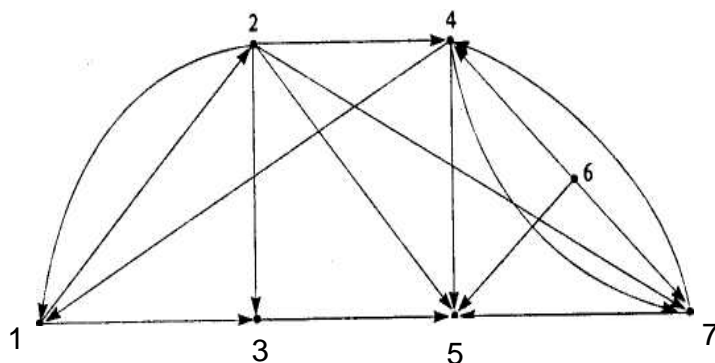
Определение. Диаметром графа G (обозначается $D(G)$) называется длина длиннейшей геодезической цепи.

Определение. Множество вершин, находящихся на одинаковом расстоянии n от вершины v (обозначение $D(v,n)$), называется ярусом.

7.3. Представление графов в ЭВМ.

Определение. Матрицей смежности неориентированного (ориентированного) помеченного графа с n вершинами $V = \{v_1, v_2, \dots, v_n\}$ называется матрица $A = [a_{ij}]$, $i, j = 1, 2, \dots, n$, в которой $a_{ij} = k$, где k – число ре-

бер, соединяющих вершины v_i, v_j (соответственно дуг, идущих из вершины v_i в вершину v_j).



$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

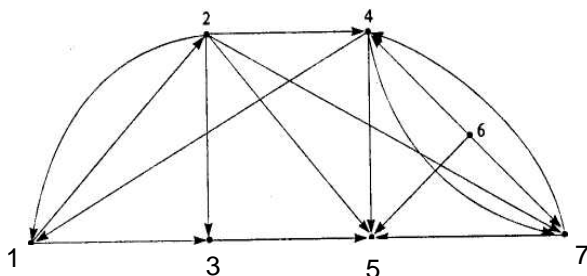
Рис. 7. Ориентированный граф и его матрица смежности A .

Определение. Матрицей инцидентности для неориентированного графа с n вершинами и m ребрами называется матрица $B = [b_{ij}]$, $i=1, 2, \dots, n$, $j = 1, 2, \dots, m$, строки которой соответствуют вершинам, а столбцы — ребрам. Элемент $b_{ij}=1$, если вершина v_i инцидентна ребру e_j и $b_{ij}=-1$, если вершина v_i не инцидентна ребру e_j .

Определение. Матрицей инцидентности для ориентированного графа с n вершинами и m ребрами называется матрица $B = [b_{ij}]$, $i=1, 2, \dots, n$, $j = 1, 2, \dots, m$, строки которой соответствуют вершинам, а столбцы — ребрам.

Ее элемент:

1. $b_{ij}=1$, если ребро e_i выходит из вершины v_j ;
2. $b_{ij}=-1$, если ребро e_i входит в вершины v_j ;
3. $b_{ij}=0$, если вершина v_j не инцидентна ребру e_i .



$$B = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 1 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix}$$

Рис. 8. Орграф и его матрица инцидентности B .

Определение. Граф называется взвешенным, если каждому его

ребру сопоставлено число. Простой взвешенный граф может быть представлен своей матрицей весов $W = [w_{ij}]$, где w_{ij} — вес ребра, соединяющего вершины $i, j = 1, 2, \dots, n$. Веса несуществующих ребер полагают равными ∞ или 0 в зависимости от приложений. Заметим, что матрица весов является простым обобщением матрицы смежности.

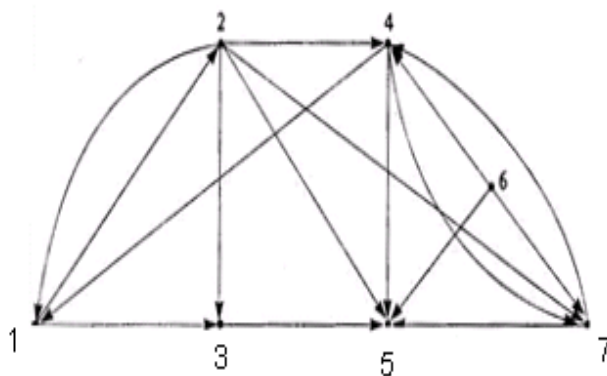
Список ребер графа. При описании графа списком его ребер каждое ребро представляется парой инцидентных ему вершин. Это представление можно реализовать двумя массивами $r = (r_1, r_2, \dots, r_m)$ и $t = (t_1, t_2, \dots, t_m)$, где m — количество ребер в графе. Каждый элемент в массиве есть метка вершины, а i -е ребро графа выходит из вершины r_i и входит в вершину t_i . Данное представление позволяет легко описать петли и кратные ребра.

Например, соответствующие массивы списка ребер для представления графа на рис. 6 будут иметь вид:

$r = (1, 1, 2, 2, 2, 2, 2, 3, 4, 4, 4, 6, 6, 6, 7, 7),$

$t = (2, 3, 1, 3, 4, 5, 7, 5, 1, 5, 7, 4, 5, 7, 4, 5).$

Структура смежности графа. Ориентированный или неориентированный граф может быть однозначно представлен структурой смежности своих вершин. Структура смежности состоит из списков $Adj[v]$ вершин графа, смежных с вершиной v . Списки $Adj[v]$ составляются для каждой вершины графа.



v_i	$Adj[v_i]$
1	2,3
2	1,3,4,5,7
3	5
4	1,5,7
5	-
6	4,5,7
7	4,5

Рис. 9. Ориентированный граф и его структура смежности.

Структуры смежности удобно представляются массивом из n (число вершин в графе) линейно связанных списков. Каждый список содержит вершины, смежные с вершиной, для которой составляется список. Во многих задачах на графах выбор представления является решающим для эффективности алгоритмов.

Лекция 8. Задачи оптимизации на графах.

Учебные вопросы:

8.1. Кратчайший путь на графе. Метод приписывания индексов.

8.2. Граф наименьшей длины.

8.1. Кратчайший путь на графе. Метод приписывания индексов.

Задача о кратчайшем пути на взвешенном графе состоит в определении в определенном смысле кратчайшего пути, соединяющего две заданные вершины графа – начальную и конечную.

Задача о кратчайшем пути на графе решается с помощью алгоритма приписывания индексов вершинам графа:

1. Каждая вершина x_i помечается индексом c_i . Первоначально конечной вершине x_0 приписывается индекс $c_0=0$. Для остальных вершин предварительно полагаем $c_i = \infty$ ($i \neq 0$).

2. Ищем такую дугу (x_i, x_j) , для которой $c_i - c_j > l(x_i, x_j)$ и заменяем индекс c_j индексом $c_j = c_i + l(x_i, x_j) < c_j$. Продолжаем этот процесс замены индексов до тех пор, пока остается хотя бы одна дуга, для которой можно уменьшить c_j .

Приписанные вершинам индексы имеют два свойства:

1. Для произвольного ребра (x_k, x_s) выполняется условие $c_s - c_k \leq l(x_k, x_s)$ так как если бы оно не выполнялось, индекс c_s нужно было бы уменьшить.

2. Пусть x_p – произвольная вершина. При рассмотренном процессе приписывания индексов индекс c_p монотонно уменьшается. Пусть x_q — последняя вершина, послужившая для его уменьшения. Тогда $c_p = c_q + l(x_q, x_p)$.

Следовательно, для произвольной вершины x_p с индексом c_p найдется вершина x_q , соединенная ребром с x_p , такая, что $c_p - c_q = l(x_q, x_p)$.

Эти свойства позволяют сформулировать следующий алгоритм нахождения кратчайшего пути.

Пусть $x_n = a$ — начальная вершина с индексом c_n . Ищем вершину x_k такую, что $c_n - c_k = l(x_k, x_n)$. Далее ищем вершину x_s такую, что $c_k - c_s = l(x_s, x_k)$ и т. д., до тех пор, пока не дойдем до конечной вершины $x_0 = b$.

Теорема. Путь $m_0 = (x_n, x_k, x_s, \dots, x_r, x_0)$, длина которого равна c_n , является кратчайшим.

Доказательство. Для доказательства рассмотрим произвольный путь из a в b $m = \{x_n, x_t, x_q, \dots, x_d, x_0\}$. Его длина будет $l(m)$. Согласно правилу расстановки индексов будут выполняться следующие неравенства:

$$c_n - c_t \leq l(x_n, x_t);$$

$$\mathbf{c}_t - \mathbf{c}_q \leq \mathbf{l}(\mathbf{x}_t, \mathbf{x}_q);$$

$$c_d - 0 \leq l(x_d, x_0).$$

Складывая эти неравенства, находим, что для любого пути m_0 имеет место $c_n - 0 \leq l(m)$. Так как для пути m_0 выполняется условие $c_n = l(m_0)$, то путь m_0 является кратчайшим.

Метод нахождения кратчайшего пути проиллюстрирован на примере карты дорог, представленной в виде графа на рис. 1. Цифры у ребер указывают время проезда по каждой из дорог. Индексы вершин, помещенные в окружности, дают время проезда от данной вершины графа до его вершины b .

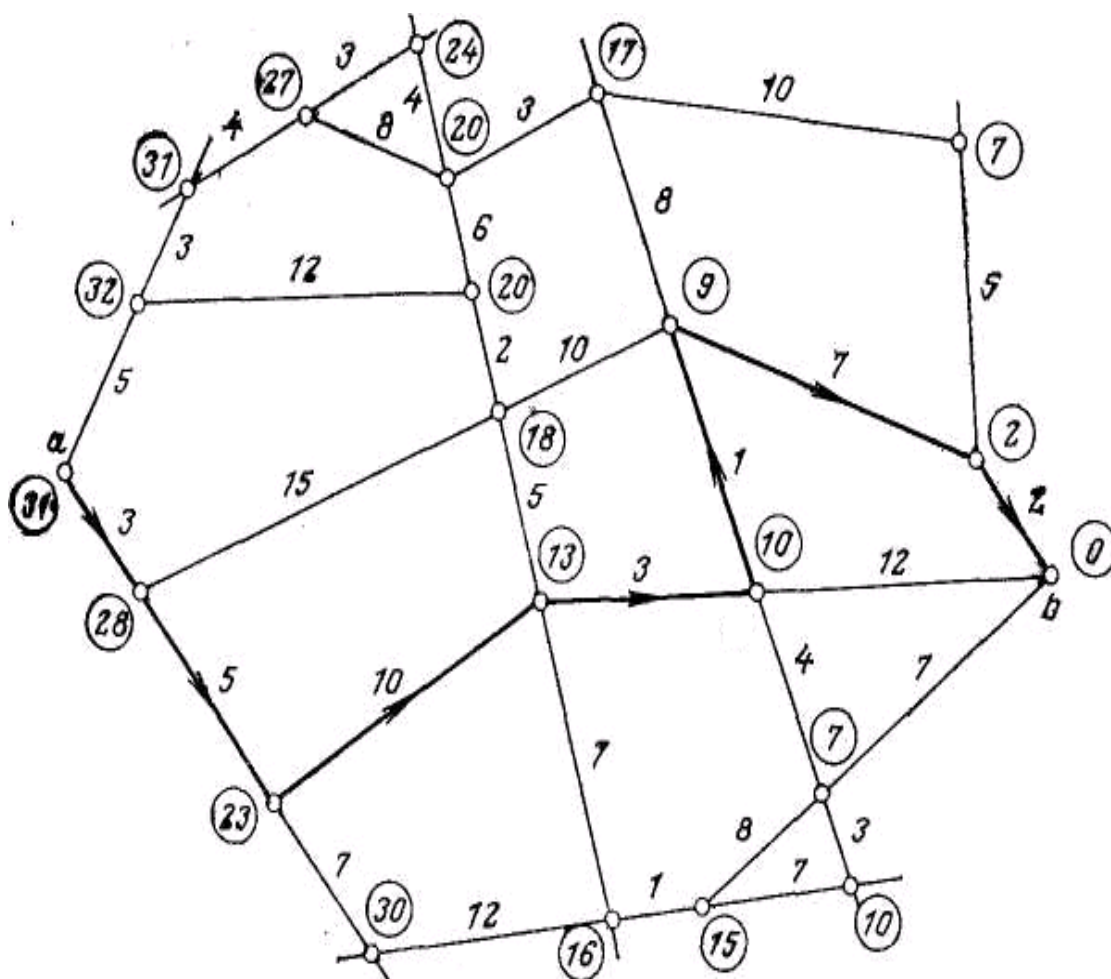


Рис. 1. Карта дорог

8.2. Граф наименьшей длины.

Большое практическое значение имеет задача о проведении дорог. Имеется несколько городов a, b, c, \dots , которые нужно соединить между собой сетью дорог. Для каждой пары городов (x, y) известна стоимость $l(x, y)$ строительства соединяющей их дороги. Задача состоит в том, чтобы построить самую дешевую из возможных сетей дорог. Вместо сети дорог можно рассматривать сеть линий электропередачи, сеть нефтепроводов, сеть ПК и т. п. Называя в графе, изображающем сеть дорог, величину $l(x, y)$ длиной ребра (x, y) , приходим к задаче о построении графа наименьшей длины. Поэтому далее в качестве стоимости дорог примем длину ребер графа.

Если имеются всего три вершины (a, b, c) , то достаточно построить одну из соединяющих цепей abc, acb, bac , причем если bc — самое длинное ребро, то именно его и надо исключить, построив цепь bac .

Граф наименьшей длины всегда является деревом, так как если бы он содержал цикл, можно было бы удалить одно из ребер этого цикла и вершины все еще остались бы соединенными. Следовательно, для соединения n вершин нужно построить $(n-1)$ ребро.

Граф наименьшей длины можно построить, пользуясь следующим правилом. Сначала соединяем две вершины с наиболее коротким ребром u_1 . На каждом из следующих шагов добавляем самое короткое из ребер u_i , при присоединении которого к уже имеющимся ребрам не образуется никакого цикла.

Если имеется несколько ребер одинаковой длины, то выбираем любое из них. Каждое дерево Q , построенное таким образом, будем называть экономическим деревом. Его длина равна сумме длин отдельных ребер:

$$L(Q) = l(u_1) + \dots + l(u_{n-1})$$

Покажем, что никакое другое дерево, соединяющее те же вершины, не может иметь длину, меньшую длины экономического дерева Q . Пусть P — дерево наименьшей длины, соединяющее рассматриваемые вершины, а Q — любое экономическое дерево.

Предположим, что ребра u_1, u_2, \dots, u_{n-1} занумерованы в том поряд-

ке, в котором они присоединялись при построении Q , т. е. удовлетворяют условию $l(u_k) \leq l(u_{k+1})$. Если дерево P не совпадает с Q , то Q имеет по меньшей мере одно ребро, не принадлежащее P . Пусть $u_i = (a, b)$ — первое такое ребро и пусть $L(a, b)$ — цепь графа P , соединяющая вершины a и b , как, например, на рис 2.

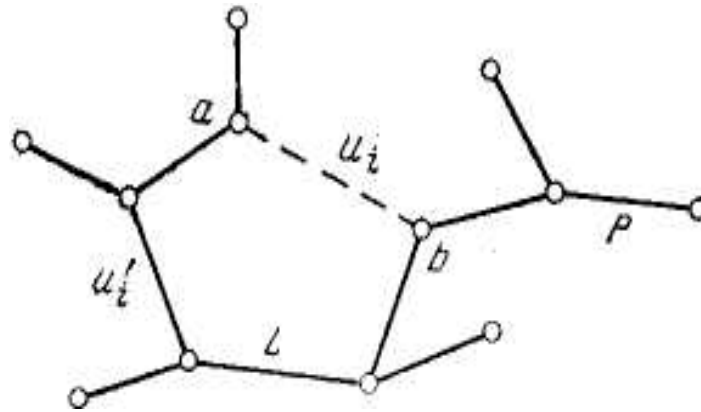


Рис. 2. К построению дерева наименьшей длины.

Если ребро u_i добавить к P , то получим цикл, а так как Q не имеет циклов, то в этот цикл должно входить по крайней мере одно ребро, не принадлежащее Q . Пусть это будет u'_i . Удалив его, получим дерево P' с тем же числом вершин, что и P , длина которого $l(P') = l(P) + l(u_i) - l(u'_i)$.

Так как граф P имеет наименьшую длину, то $l(u_i) \geq l(u'_i)$. Но u_i было ребром наименьшей длины, при добавлении которого к ребрам u_1, u_2, \dots, u_{i-1} не получается циклов. Так как при добавлении u'_i к этим ребрам также не получается никакого цикла, то $l(u_i) = l(u'_i)$ и, следовательно, P' имеет, так же как и P , наименьшую длину. Но P' имеет с экономическим деревом Q на одно общее ребро больше, чем P . Повторяя эту операцию несколько раз, получаем дерево наименьшей длины, совпадающее с Q . Следовательно, Q — дерево наименьшей длины.

Лекция 9. Основные понятия теории автоматов.

Учебные вопросы:

9.1. Конечные автоматы.

9.2. Способы задания конечных автоматов.

9.1. Конечные автоматы

Дискретные устройства без памяти, в которых значения выходов однозначно определяются входными сигналами в **тот же момент времени**, то есть без запоминания, называются **комбинационными схемами**.

В более общих случаях выходы устройств могут зависеть не только от входных сигналов, но и от внутренних состояний, в которых находятся эти устройства. Посредством внутренних состояний может быть уточнена необходимая информация о сигналах, поступивших на входы устройства раньше. **Если число этих состояний конечно, то говорят об устройствах с конечной памятью**. Математической моделью устройств с конечной памятью являются **конечные автоматы**.

Конечный автомат имеет **n** входов и **m** выходов (рис.1).

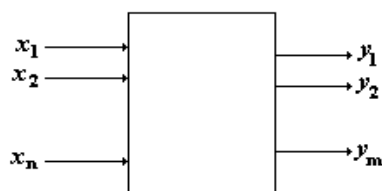


Рис.1. Конечный автомат с n входами и m выходами.

На входы x_1, x_2, \dots, x_n могут подаваться символы конечного алфавита **A** , выходы автомата y_1, y_2, \dots, y_m могут принимать значения из конечного алфавита **B** . Автомат имеет конечное число внутренних состояний **$Q = \{q_0, q_1, \dots, q_{r-1}\}$** .

Конечный автомат функционирует в дискретные моменты времени **t_0, t_1, t_2, \dots** . Будем обозначать эти моменты времени **$t = 0, 1, 2, \dots$** .

Если обозначить через **$x_i(t)$** , **$y_j(t)$** , **$q(t)$** значения входа x_i , выхода y_j и состояния **q** в момент времени **t** , то работа автомата описывается **$m+1$** уравнениями

$$y_j(t) = \Phi_j(x_1(t), x_2(t), \dots, x_n(t), q(t-1)), \quad (j = 1, 2, \dots, m)$$

$$q(t) = \Psi(x_1(t), x_2(t), \dots, x_n(t), q(t-1)),$$

которые называют **каноническими**. Функции Φ_j и Ψ называются соответственно **функцией j-того выхода** и **функцией переходов**.

Для того, чтобы задать поведение автомата, нужно дополнительно указать его начальное состояние $q(0)$. Зная $q(0)$ и входные символы в 1-ый момент времени, можно из канонических уравнений, определить значения выходов и состояние в 1-ый момент времени, по $q(1)$ и входным символам во 2-й момент времени можно определить значения выходов и состояние во 2-й момент времени и т.д.

Существует два типа автоматов: **инициальные** и **неинициальные**. В инициальных автоматах начальное состояние фиксировано (то есть они всегда начинают функционировать из одного и того состояния). В неинициальных автоматах в качестве начального состояния может быть взято любое. Выбор начального состояния определяет поведение автомата в последующие моменты.

Пример 1. Требуется построить автомат, описывающий работу двоичного сумматора **последовательного** действия.

Обозначим входы сумматора через x_1 и x_2 , а выход – через y . Состояния, соответствующие наличию и отсутствию переносов, обозначим соответственно символами **1** и **0**. Функции выхода и перехода, в чем можно убедиться с помощью таблиц истинности логики высказываний, будут иметь вид:

$$y(t) = x_1(t) \oplus x_2(t) \oplus q(t-1),$$

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

Последовательный сумматор начинает функционировать из состояния $q(0)=0$, соответствующего отсутствию переноса, поэтому ему отвечает инициальный автомат.

Обозначим через A' n -тую декартову степень множества A , то есть $A' = A \times A \times \dots \times A$ (n раз), а через B' m -тую декартову степень множества B , то есть $B' = B \times B \times \dots \times B$ (m раз). Элементами множеств A' и B' являются n -ки и m -ки символов из A и B соответственно. Тогда автомат с n входами и m выходами можно рассматривать как автомат с 1 входом и 1 выходом. Каждую входную n -ку $\langle a^{(1)}, a^{(2)}, \dots, a^{(n)} \rangle$ можно рассматривать как один входной символ, а каждую выходную m -ку $\langle b^{(1)}, b^{(2)}, \dots, b^{(m)} \rangle$ – как один выходной символ. Таким образом, изучение автоматов с произвольным числом входов и выходов сводится к изучению автоматов с 1 входом и 1 выходом.

Итак, в качестве основной модели мы будем рассматривать авто-

мат с 1 входом x и 1 выходом y .

Он описывается каноническими уравнениями:

$$\begin{aligned} y(t) &= \Phi(x(t), q(t-1)), \\ q(t) &= \Psi(x(t), q(t-1)). \end{aligned}$$

Входные сигналы принимают значения из конечного алфавита A , выходные – из конечного алфавита B , состояниями являются символы конечного алфавита Q . Автоматы этого вида называют **автоматами Миля**.

9.2. Способы задания автоматов

Помимо задания автоматов системами канонических уравнений для решения разных задач, связанных с анализом, синтезом и упрощением устройств, оказываются удобными различные способы задания автоматов.

Рассмотрим наиболее распространенные. Эти способы относятся к неинициальным автоматам (для инициальных дополнительно указывается начальное состояние).

Задание автомата с помощью таблицы. Функции Φ и Ψ задаются таблицей, строки которой соответствуют буквам входного алфавита, а столбцы – состояниям. В пересечении строки a_j и столбца q_i помещается состояние $\Psi(a_j, q_i)$, в которое переходит автомат из состояния q_i под воздействием входного символа a_j , и значение $\Phi(a_j, q_i)$, которое при этом появляется на выходе.

В таблице 1. приведен пример автомата с алфавитами $A=\{0, 1, 2\}$, $B=\{0, 1\}$ и $Q = \{q_0, q_1, q_2, q_3\}$.

Таблица 1.

Символы внешнего алфавита	С о с т о я н и я			
	q_0	q_1	q_2	q_3
0	$q_1, 1$	$q_2, 1$	$q_2, 0$	$q_1, 0$
1	$q_0, 1$	$q_0, 0$	$q_2, 0$	$q_0, 1$
2	$q_0, 0$	$q_0, 1$	$q_2, 1$	$q_3, 0$

Задание автомата в виде диаграммы. Состояния q_i изображаются на плоскости кружками. Из кружка q_i проводится стрелка в q_u , если автомат, находящийся в состоянии q_i , при подаче некоторого входного символа может быть переведен в состояние q_u . Пусть $a^{(1)}_j, a^{(2)}_j, \dots, a^{(s)}_j$ – все входные символы, переводящие состояние q_i в состояние q_u , а $b^{(1)}_v, b^{(2)}_v, \dots, b^{(s)}_v$ – сигналы на выходе, соответствующие этому переходу. Тогда стрелке диаграммы, ведущей из q_i в q_u , приписывается выражение $(a^{(1)}_j, b^{(1)}_v) \vee (a^{(2)}_j, b^{(2)}_v) \vee \dots \vee (a^{(s)}_j, b^{(s)}_v)$.

На рис.2. приведена диаграмма автомата, заданного таблицей 1.

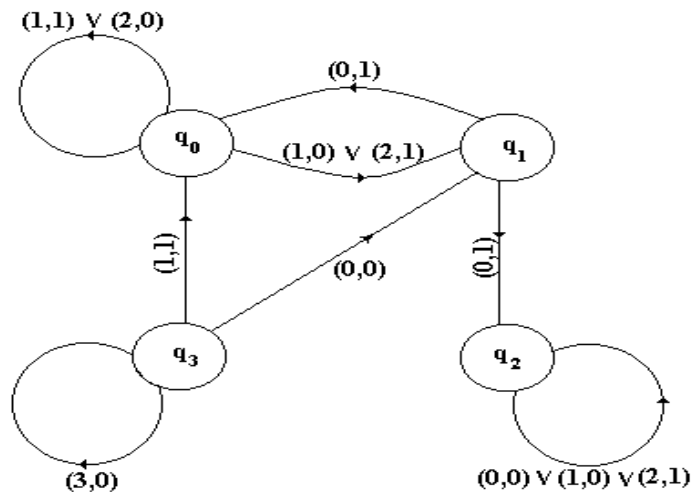


Рис. 2. Диаграмма автомата

Диаграммы обладают большей наглядностью, чем таблицы. Так, например, из приведенной диаграммы видно, что состояние q_2 является тупиковым. Оказавшись в нем, автомат навсегда остается в этом состоянии. Заметить подобные свойства по таблице сложно.

Задание автомата совокупностью четверок. Работа автомата может быть описана совокупностью четверок вида $q_i a_j \rightarrow q_l a_p$, которую можно интерпретировать следующим образом: если на вход автомата, находящегося в состоянии q_i , в момент времени t подается символ a_j , то на выходе автомата в этот же момент времени появляется символ a_p , и в следующий момент времени $t+1$ автомат будет находиться в состоянии q_l .

Пример 1. Пусть конечный автомат с алфавитами $A=\{0, 1, 2\}$, $B=\{0, 1\}$ и $Q = \{q_0, q_1, q_2, q_3\}$ задан таблицей

Символы внешнего алфавита	Состояния			
	q_0	q_1	q_2	q_3
0	$q_1, 1$	$q_2, 1$	$q_2, 0$	$q_1, 0$
1	$q_0, 1$	$q_0, 0$	$q_2, 0$	$q_0, 1$
2	$q_0, 0$	$q_0, 1$	$q_2, 1$	$q_3, 0$

Этот автомат может быть описан следующей совокупностью четверок:

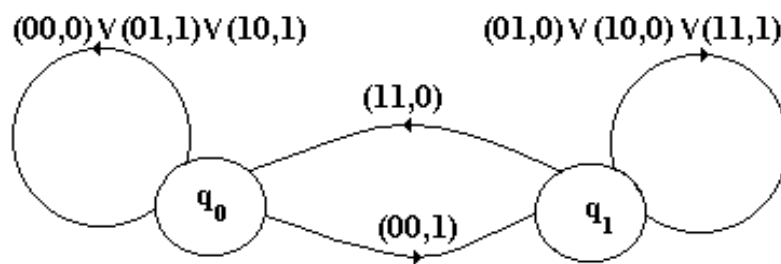
$$\begin{aligned}
 q_0 0 &\rightarrow q_1 1 & q_1 0 &\rightarrow q_2 1 & q_2 0 &\rightarrow q_2 0 & q_3 0 &\rightarrow q_1 0 \\
 q_0 1 &\rightarrow q_0 1 & q_1 1 &\rightarrow q_0 0 & q_2 1 &\rightarrow q_2 0 & q_3 1 &\rightarrow q_0 1 \\
 q_0 2 &\rightarrow q_0 0 & q_1 2 &\rightarrow q_0 1 & q_2 2 &\rightarrow q_2 1 & q_3 2 &\rightarrow q_3 0
 \end{aligned}$$

Пример 2. Требуется построить таблицу и диаграмму автомата, представляющего двоичный сумматор последовательного действия.

Обозначим через q_0 и q_1 его состояния, соответствующие отсутствию и наличию переноса. Таблица, задающая автомат, будет иметь вид:

Символы внешнего алфавита		Состояния	
x_1	x_2	q_0	q_1
0	0	$q_0, 0$	$q_0, 1$
0	1	$q_0, 1$	$q_1, 0$
1	0	$q_0, 1$	$q_1, 0$
1	1	$q_1, 0$	$q_1, 1$

Диаграмма этого автомата имеет вид:



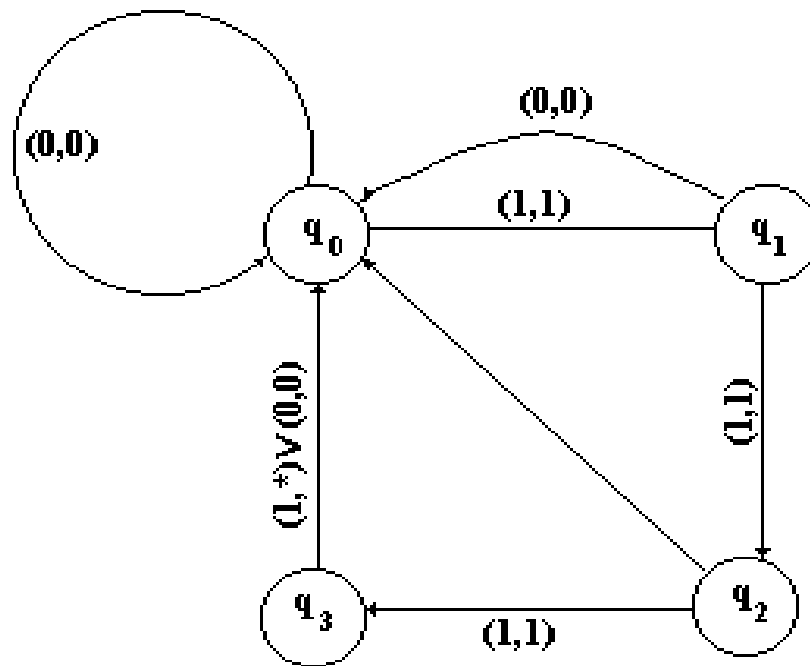
Пример 3. Построить конечный автомат, воспринимающий на входе двоичную последовательность и выдающий на выходе специальный символ (например, *) в том случае, если во входной последовательности подряд встретится 4 единицы. В остальных случаях автомат на выходе повторяет входной символ.

Программа работы такого автомата в виде четверок будет следующей:

$q_0 0 \rightarrow q_0 0$	$q_2 0 \rightarrow q_0 0$
$q_0 1 \rightarrow q_1 1$	$q_2 1 \rightarrow q_3 1$
$q_1 0 \rightarrow q_0 0$	$q_3 0 \rightarrow q_0 0$
$q_1 1 \rightarrow q_2 1$	$q_3 1 \rightarrow q_0 *$

Посредством состояния q_1 автомат «запоминает» первую единицу на входе, посредством состояния q_2 автомат «запоминает» две единицы подряд на входе (11), q_3 - три единицы подряд на входе (111), и если в этом состоянии на вход автомата поступает четвертая единица подряд (1111), то он выдает на выходе * и возвращается в состояние q_0 . В это же состояние автомат возвращается, если на вход автомата в состояниях q_1 , q_2 , q_3 поступает нуль, то есть автомат «забывает» всю предысторию.

Диаграмма этого автомата представлена на следующем рисунке:



9.3. Другие модели конечных автоматов

Автомат Мили не является единственной моделью дискретного устройства с конечной памятью. Широкое распространение получила и другая модель – **автомат Мура**. В нем значение выхода определяется состоянием **в тот же момент** времени, в связи с чем канонические уравнения имеют вид:

$$\mathbf{q}(t) = \Psi(\mathbf{x}(t), \mathbf{q}(t-1)),$$

$$\mathbf{y}(t) = \lambda(\mathbf{q}(t)).$$

Подставив во второе уравнение из первого значение $\mathbf{q}(t)$ и положив $\Phi = \lambda(\Psi)$, можно преобразовать второе уравнение к виду

$$\mathbf{y}(t) = \lambda(\Psi(\mathbf{x}(t)$$

$$\mathbf{q}(t-1))) = \Phi(\mathbf{x}(t), \mathbf{q}(t-1)).$$

Таким образом, автомат Мура можно рассматривать как частный случай автомата Мили при функции выхода специального вида $\Phi = \lambda(\Psi)$.

Автоматы Мили в некотором смысле также сводятся к автоматам Мура. Поэтому имеет место утверждение: **для всякого автомата Мили существует эквивалентный ему автомат Мура**, которое мы здесь принимаем без доказательства.¹⁾

¹⁾ Доказательство см., например, в книге Л.А. Шоломова «Основы теории дискретных логических и вычислительных устройств», -М.:Наука, 1980.

Описанные модели автоматов соответствуют так называемым **синхронным дискретным устройствам**, которые характеризуются тем, что в них имеется специальное устройство – генератор синхронизирующих импульсов, определяющее моменты времени, в которые происходит изменение состояний. Соседние моменты времени обычно оказываются разделенными равными временными промежутками.

Другой разновидностью дискретных устройств являются **асинхронные устройства**. В них моменты перехода из одного состояния в другое заранее не определены. При подаче входного сигнала устройство проходит через некоторую последовательность состояний (**неустойчивых**) и останавливается в **устойчивом** состоянии. После этого можно подавать следующие входные сигналы. **Устойчивое состояние характеризуется тем, что при повторной подаче на входы устройства того же входного сигнала состояние устройства не изменяется.** Значение выхода устройства определяется обычно устойчивым состоянием.

Если не учитывать переходный процесс (смену неустойчивых состояний) и принимать во внимание лишь устойчивые состояния, то можно ввести абстрактное время $t = 0, 1, 2, \dots$, определяемое моментами t_0, t_1, t_2, \dots переходов в устойчивые состояния под действием входной последовательности. При этом моделью асинхронного дискретного устройства будет автомат Мура, функция переходов которого удовлетворяет условию

$$\Psi(x, \Psi(x, q)) = \Psi(x, q)$$

при любых x и q . Это условие означает, что состояние полученное автоматом при повторной подаче одного и того же входного сигнала, совпадает с состоянием, в которое переходит автомат после однократной подачи этого сигнала.

Наиболее общей из рассмотренных моделей является модель автомата Мили, все остальные можно считать ее частными случаями.

Лекция 10. Минимизация конечных автоматов

Учебные вопросы:

10.1. Эквивалентность конечных автоматов.

10.2. Построение минимального конечного автомата.

10.1. Эквивалентность конечных автоматов.

Автоматы являются моделями устройств для переработки дискретной информации. Входной и выходной алфавиты определяются перерабатываемой информацией, в то время как на их внутренний алфавит Q никаких условий обычно не накладывается. Требуемое преобразование информации может быть осуществлено автоматами с разным числом состояний. Поэтому возникает задача построения автомата, для которого это число **минимально**.

Состояния q автомата M и q' автомата M' , будем называть **эквивалентными**, если оба автомата, получив любую одну и ту же входную последовательность в состояниях q и q' соответственно, перерабатывают ее в одинаковую выходную последовательность.

Если M и M' означают один и тот же автомат, то получаем определение эквивалентности состояний одного автомата.

Автоматы M и M' будем называть **эквивалентными**, если для каждого состояния автомата M существует эквивалентное состояние автомата M' , и наоборот.

Автомат, эквивалентный заданному и имеющий наименьшее возможное число состояний, называют **минимальным**.

Задачу построения минимального автомата называют задачей **минимизации автомата**.

Рассмотрим множество Q всех состояний некоторого автомата M . Отношение эквивалентности состояний обладает всеми обычными свойствами эквивалентности: **рефлексивностью, симметричностью, транзитивностью**. Поэтому согласно известному результату множество Q разбивается на **классы эквивалентности**.

Состояния, принадлежащие одному классу, эквивалентны между собой, а разным классам – неэквивалентны.

Построение минимального автомата будет осуществляться в два этапа. Вначале множества Q разбивается на классы эквивалентности, а затем на их основе строится минимальный автомат.

Разбиение множества состояний на классы эквивалентности.

Рассмотрим вначале метод нахождения всех пар $q_i q_j$ эквивалентных состояний на примере автомата, заданного таблицей

Символы входного алфавита	Состояния					
	q_1	q_2	q_3	q_4	q_5	q_6
0	$q_4, 1$	$q_2, 0$	$q_4, 1$	$q_4, 1$	$q_6, 0$	$q_5, 0$
1	$q_6, 0$	$q_2, 0$	$q_2, 0$	$q_3, 0$	$q_6, 0$	$q_6, 0$
2	$q_1, 0$	$q_6, 1$	$q_3, 0$	$q_3, 0$	$q_5, 1$	$q_2, 1$

Составим треугольную таблицу, клетки которой соответствуют всем различным неупорядоченным парам $q_i q_j$ ($i \neq j$), и заполним ее следующим образом. Если для состояний q_i и q_j существует входной символ x , приводящий к разным значениям выхода, то соответствующую клетку таблицы отметим серым цветом (или перечеркнем).

В нашем примере так можно поступить с клеткой сопоставленной, например, паре $q_1 q_2$, ибо при $x=0$ автомат в состояниях q_1 и q_2 выдает 1 и 0 соответственно.

Если же при каждом x выход автомата в состояниях q_i и q_j принимает одинаковые значения, то в клетке запишем все пары состояний $q_v q_w$ ($v \neq w$), отличные от $q_i q_j$, в которые автомат может перейти из q_i и q_j при подаче одного и того же входного символа.

Так, например, в клетке для пары $q_5 q_6$ следует записать пару $q_2 q_5$, возникающую при $x=2$, т.к. $x=0$ приводит к исходной паре $q_5 q_6$, а $x=1$ - к паре $q_6 q_6$ одинаковых состояний. В соответствии с этой процедурой рассматриваемому примеру соответствует следующая треугольная таб-

лица:

q_2					
q_3	q_2q_6				
q_4	q_3q_6 , q_1q_3		q_2q_3		
q_5		q_2q_6 , q_5q_6			
q_6		q_2q_5			q_2q_5
	q_1	q_2	q_3	q_4	q_5

Далее в таблице следует зачеркнуть клетки, в которых присутствуют пары, соответствующие вычеркнутым (серым) клеткам. Отметим их также серым цветом. В нашем примере это клетки для пары q_1q_4 (в ней содержится пара q_3q_6) и клетку для пары q_3q_4 (в ней содержится пара q_2q_3). Затем снова необходимо зачеркнуть (отметить серым цветом) все клетки, которые содержат пары, соответствующие вычеркнутым (отмеченным серым цветом) клеткам, и так далее, пока не образуется таблица, в которой больше нельзя вычеркнуть ни одной клетки. В рассматриваемом примере таким свойством обладает последняя таблица.

Таблица примет следующий вид:

q_2					
q_3	q_2q_6				
q_4					
q_5		q_2q_6 , q_5q_6			
q_6		q_2q_5			q_2q_5
	q_1	q_2	q_3	q_4	q_5

Невычеркнутые (белые) клетки результирующей таблицы соответствуют всем парам эквивалентных состояний.

Действительно, если состояния q' и q'' неэквивалентны, то найдется входная последовательность $\tilde{x} = x_1 x_2 \dots x_p$, на которой автомат, запущенный в состояниях q' и q'' , выдает разные выходные последовательности. Можно считать, что значения выходов различаются лишь на шаге p (то есть после подачи x_p), иначе последовательность $\tilde{x} = x_1 x_2 \dots x_p$ можно укоротить (до первого различия выходов).

Пусть автомат при подаче последовательности $\tilde{x} = x_1 x_2 \dots x_p$, начав функционировать в состоянии q' , проходит последовательность состояний q'_1, q'_2, \dots, q'_p , а из состояния q'' — $q''_1, q''_2, \dots, q''_p$. В состояниях q'_{p-1} и q''_{p-1} подача x_p приводит к разным значениям выхода, поэтому пара $q'_{p-1} q''_{p-1}$ будет вычеркнута в треугольной таблице на первом шаге. Поскольку сигнал x_{p-1} превращает пару $q'_{p-2} q''_{p-2}$ в пару $q'_{p-1} q''_{p-1}$, то на следующем шаге будет вычеркнута пара $q'_{p-2} q''_{p-2}$ и так далее. Не позже, чем через p шагов исходная пара состояний $q' q''$ будет вычеркнута.

Если же состояния q' и q'' эквивалентны, то всякой входной последовательности будут соответствовать одинаковые выходные последовательности и не найдется пары состояний, исходя из которой цепочка вычеркиваний приведет к $q' q''$.

На основе последней таблицы выпишем все эквивалентные пары состояний (они соответствуют невычеркнутым клеткам таблицы): $q_1 q_3$, $q_2 q_5$, $q_2 q_6$, $q_5 q_6$. Класс эквивалентности образуется состояниями, которые попарно эквивалентны. В данном случае получаем классы: $\{q_1, q_3\}$ и $\{q_2, q_5, q_6\}$. Каждое состояние, не вошедшее ни в один из этих классов, эквивалентно лишь самому себе и само образует класс эквивалентности. Добавив эти классы, получаем требуемое разбиение. В рассматриваемом примере к классам $\{q_1, q_3\}$ и $\{q_2, q_5, q_6\}$ следует добавить $\{q_4\}$.

10.2. Построение минимального конечного автомата.

Пусть имеется некоторое разбиение всех состояний автомата M на множества Q_1, Q_2, \dots, Q_s . Разбиение назовем **замкнутым**, если для любого множества Q_v и любого входного сигнала x множество всех состояний, в которое под воздействием x переходят состояния из Q_v , целиком содержится в некотором множестве Q_w , зависящем от v и x .

Разбиение множества состояний на классы эквивалентности

является замкнутым. Действительно, если состояния q_i и q_j эквивалентны и при подаче x они переходят в состояния q'_i и q'_j , то последние также будут эквивалентны. Если бы это было не так и существовала входная последовательность $x_1 x_2 \dots x_p$, дающая в применении к состояниям q'_i и q'_j разные выходные последовательности, то последовательность $x = x_1 x_2 \dots x_p$, примененная к q_i и q_j , также приводила бы к разным выходным последовательностям, что противоречит эквивалентности q_i и q_j . Таким образом, состояния из одного класса эквивалентности при подаче одинакового входного воздействия снова попадают в один класс эквивалентности, что и указывает на замкнутость разбиения.

Имея разбиение множества состояний Q автомата M на классы эквивалентности Q_1, Q_2, \dots, Q_s , построим новый автомат M' . Для этого каждому классу Q_v сопоставим состояние q'_v ($v = 1, 2, \dots, s$), а в качестве входного и выходного алфавитов автомата M' возьмем соответствующие алфавиты автомата M . Чтобы назначить переход в автомате M' из состояния q'_v под действием символа x , рассмотрим соответствующий класс эквивалентности Q_v . Согласно свойству замкнутости входной символ x переводит все состояния из Q_v в некоторый класс эквивалентности Q_w . Им определяется следующее состояние q'_w . По свойству эквивалентности состояний подача входного символа x в каждом состоянии из Q_v приводит к одинаковому значению выхода. Оно принимается в качестве выходного значения, соответствующего переходу в автомате M' из состояния q'_v под действием x .

Автомат M' , построенный указанным способом по автомату M , заданному приведенной выше таблицей, с учетом разбиения на классы эквивалентности $\{q_1, q_3\}$, $\{q_2, q_5, q_6\}$, $\{q_4\}$ и сопоставления классам новых состояний: $\{q_1, q_3\} \longleftrightarrow q'_1$, $\{q_2, q_5, q_6\} \longleftrightarrow q'_2$, $\{q_4\} \longleftrightarrow q'_3$, представлен в следующей таблице:

Символы внешнего алфа- вита	Состояния		
	q'_1	q'_2	q'_3
0	$q'_{3,1}$	$q'_{2,0}$	$q'_{3,1}$
1	$q'_{2,0}$	$q'_{2,0}$	$q'_{1,0}$
2	$q'_{1,0}$	$q'_{2,1}$	$q'_{1,0}$

Таким образом, процедура минимизации конечного автомата нами описана.

Примем без доказательства ²⁾ следующее утверждение:

Всякий минимальный автомат с точностью до переобозначения состояний совпадает с автоматом M' , построенным выше.

² Доказательство см. в книге: Л.А. Шоломов Основы теории дискретных логических и вычислительных устройств. -М.:Наука, 1980.

Лекция 11. Основные понятия теории алгоритмов

Учебные вопросы:

11.1. Неформальное понятие алгоритма. Свойства алгоритма.

11.2. Способы описания алгоритмов.

11.3. Виды алгоритмических процессов.

Понятие алгоритма, определяемое требованиями дискретности и элементарности шагов, детерминированности, результативности и массовости, не является строгим: в формулировках встречаются слова "последовательность действий", "система величин", "простой" и т.п., точный смысл которых не установлен. В дальнейшем такое понятие алгоритма мы будем называть непосредственным или **интуитивным понятием алгоритма**, то есть соответствующим нашей интуиции и опыту.

Интуитивное понятие алгоритма нестрогое, но практически не возникало случаев, когда математики (прежде всего) разошлись бы во мнениях относительно того, является ли алгоритмом тот или иной конкретно заданный процесс. Однако ситуация оказывается принципиально иной, когда приходится иметь дело с задачами, решение которых не известно и относительно которых имеется предположение, что они по своей сути не могут быть решены алгоритмическими методами, то есть **алгоритмически неразрешимы**. Доказать алгоритмическую неразрешимость, то есть **отсутствие** алгоритма решения рассматриваемой проблемы, на основе интуитивного представления об алгоритме невозможно. Доказать существование алгоритма можно путем фактического описания процесса, решающего проблему. Доказать несуществование алгоритма таким путем невозможно. Для этого надо точно знать, отсутствие чего мы должны доказать.

В середине 30-х годов двадцатого века были предприняты попытки формализовать и тем самым уточнить это понятие. В результате было предложено множество различных формализаций понятия алгоритма. Решением этих проблем занимались Д. Гильберт, К. Гедель, А. Чёрч, С. Клини, Э. Пост, А. Тьюринг, А.А. Марков. Впоследствии было установлено, что различные формализмы эквивалентны в том смысле, что классы точно описываемых ими процессов (задач) совпадают. Это дает основания для **предположения о том, что класс задач, решаемых в любой из этих формальных систем, и есть класс всех задач, которые могут быть решены "интуитивно алгоритмическими ме-**

тодами". Явно эта гипотеза (правда, в несколько ином виде) была высказана А. Чёрчем и носит название тезиса Чёрча. Сейчас она является общепризнанной.

Формальное описание алгоритма создало предпосылки для разработки **теории алгоритмов**. Исследования в этой области стимулировали развитие вычислительной техники. С другой стороны, прогресс вычислительной техники благотворно повлиял на развитие теории алгоритмов.

11.1. Неформальное понятие алгоритма. Свойства алгоритмов

Понятие вычислительной машины тесно связано с понятием вычислительной процедуры, или алгоритма.

Алгоритмом называют точное и понятное исполнителю описание последовательности действий, позволяющих от исходных данных перейти к искомому результату.

Исполнителем алгоритма может быть человек, механическое, электрическое, электронное или иное устройство.

Исходные данные представляют собой, как правило, конечную систему величин, которая "перерабатывается" в систему выходных, искомых величин.

Простейшими алгоритмами являются правила, по которым выполняется то или иное из четырех арифметических действий в десятичной системе счисления.

В математике серия (класс) задач определенного типа считается решенной, если для ее решения найден алгоритм. Нахождение алгоритмов является естественной целью математики.

Однако не только в математике, но и в других областях человеческой деятельности встречаются процессы, протекающие по строго определенному формальному предписанию, то есть алгоритму.

Алгоритм обладает рядом свойств, среди которых наиболее важными являются следующие.

Дискретность алгоритма. Алгоритм рассматривается как процесс преобразования исходной системы величин, протекающий в дискретном времени³⁾ так, что в каждый следующий момент времени система

³⁾ Дискретность времени понимается как последовательность временных интервалов: $(t_0, t_1), (t_1, t_2), (t_2, t_3), \dots, (t_{n-1}, t_n)$, в течение каждого из которых совершается то или иное действие. Удобнее эти временные интервалы рассматривать как отдельные точки на непрерывной оси времени, отождествляя интервал с его началом: $t_0, t_1, t_2, t_3, \dots, t_n$. Если текущим является момент времени t , то следующий момент времени часто обозначают $t+1$.

величин получается по определенному закону (правилу) из системы величин, имевшихся в предыдущий момент.

Элементарность шагов алгоритма. Закон (правило) получения последующей системы величин из предыдущей должен быть простым и понятным исполнителю. Иными словами, решение задачи распадается на ряд шагов, каждый из которых должен быть достаточно простым.

Детерминированность(определенность) алгоритма. Система величин, получаемых в любой не начальный момент времени, однозначно определяется системой величин, полученных в предшествующие моменты времени. То есть после выполнения очередного шага однозначно определено, что делать на следующем шаге.

Результативность (направленность) алгоритма. Если способ получения последующей величины из какой-нибудь заданной величины не дает результата, то должно быть указано, что считать результатом алгоритма. Иными словами, алгоритм всегда должен давать результат, то есть он всегда должен **заканчиваться**, выдавая результат.

Массовость алгоритма. Начальная система величин может выбираться из некоторого потенциально бесконечного множества. Иными словами, алгоритм должен быть пригоден для решения всех задач из заданного класса, а не только для решения одной конкретной задачи.

11.2.Способы описания алгоритмов.

Применяются несколько способов описания алгоритма (то есть процесса) преобразования исходных данных в искомый результат.

1. **Словесный.**

2. **В виде графических схем (блок-схем).**

3. **В виде текстов на специальных алгоритмических языках.**

Приведем примеры описания алгоритма решения задачи нахождения остатка от деления натурального числа n на натуральное число m .

В словесной форме алгоритм решения задачи можно описать следующим образом:

Если $n < m$, то считать результатом число n и завершить процесс. В противном случае перейти к п.2.

Положить $r = n$.

Если $r = m$, то считать результатом число 0 и завершить процесс. В противном случае перейти к п.4.

Вычислить $r = r - m$.

Если $r < m$, то считать результатом число r и завершить процесс. В противном случае перейти к п. 4.

При описании алгоритмов в виде графических схем (блок-схем) исходят из следующего набора элементов - геометрических фигур, каждая из которых имеет определенное назначение (рис.1).

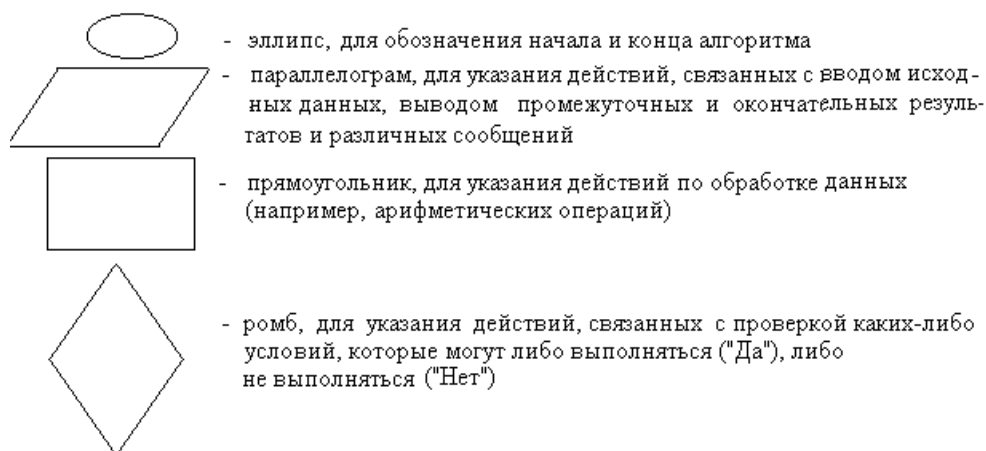


Рис. 1. Базовые графические элементы для представления алгоритмов в виде блок-схем.

Эти фигуры соединяются между собой с помощью стрелок, указывающих порядок выполнения действий. Стрелки считаются ориентированными сверху вниз и слева направо. Если направление стрелок иное, то на стрелке ставится значок \leftarrow или \uparrow .

Блок-схема алгоритма решения задачи нахождения остатка от деления натурального числа n на натуральное число m представлена на рис. 2.

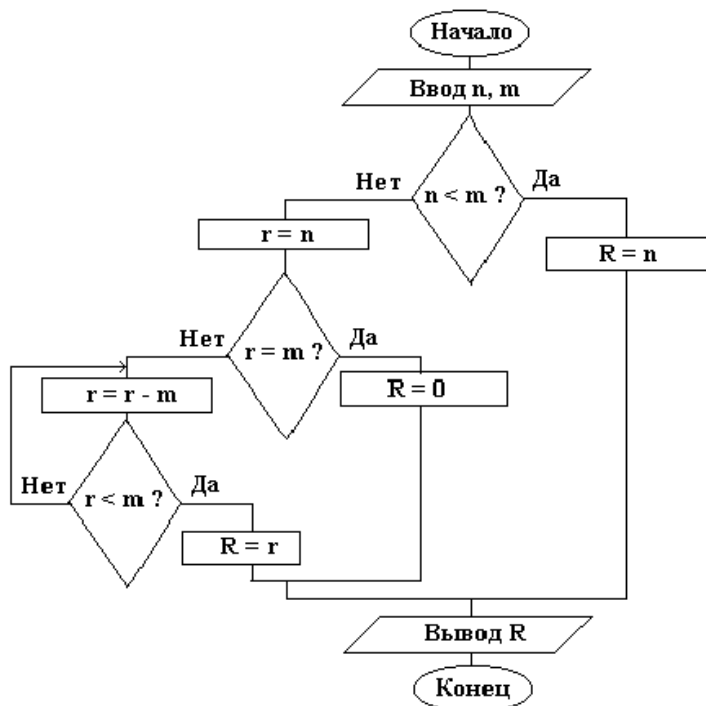


Рис. 2. Блок-схема алгоритма нахождения остатка от деления нату-

рального числа n на натуральное число m

На алгоритмическом языке **Pascal** запись рассматриваемого алгоритма представлена на рис.3.

```
function Rest (n,m: integer): integer;  
var  r: integer;  
label 1;  
begin  
    if n<m then Rest := n  
    else  
        begin  
            r := n;  
            if r = m then Rest := 0  
            else begin  
                1: r := r - m;  
                if r < m then Rest := r  
                else goto 1  
            end  
        end  
    end  
end.
```

Рис. 3. Описание алгоритма нахождения остатка от деления натурального числа n на натуральное число m на языке Pascal

Заметим, что все три описания семантически (т. е. по смыслу) эквивалентны между собой.

11.3. Виды алгоритмических процессов

Процесс обработки информации будем называть **алгоритмическим**, если существует алгоритм, описывающий этот процесс.

Различают три основных вида алгоритмических процессов:

- а) **линейные**;
- б) **разветвляющиеся**;
- в) **циклические**.

Процесс обработки информации называют **линейным**, если составляющие его действия выполняются последовательно друг за другом, т.е. "в одну линию". Графически линейный процесс представлен на рис.4.

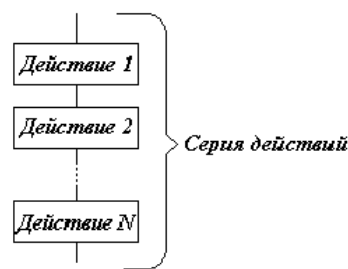


Рис. 4. Общий вид линейного процесса обработки информации.

Последовательность действий линейного процесса называют **серией действий** или просто **серией**.

В простейшем случае линейный процесс обработки информации - это вычисление по некоторой формуле или формулам.

Процесс обработки информации называют **разветвляющимся**, если в ходе его осуществляется проверка некоторого условия, в зависимости от результата которой выполняется та или иная серия действий, т.е. процесс продолжается по той или иной "ветви". Графически разветвляющийся процесс представлен на рис. 5 .

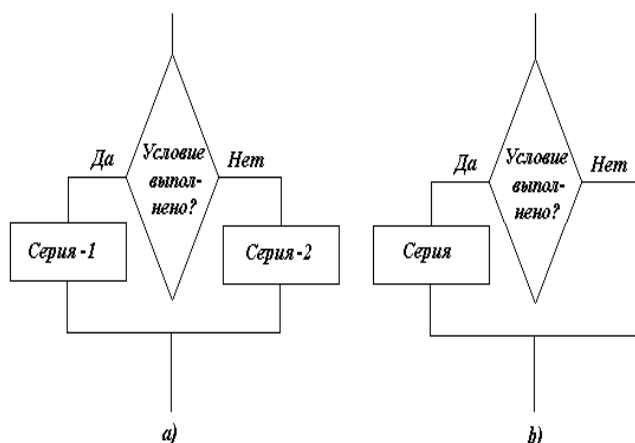


Рис.5. Графическое изображение разветвления процесса обработки информации: а) - разветвление типа "Если..., то..., иначе..."; б) - типа "Если..., то..." .

Процесс обработки информации называют **циклическим**, если в нем имеется повторяющаяся часть, или участок. Такой участок процесса называют **циклическим участком** или просто **циклом**.

Графически простейший цикл представлен на рис.6.

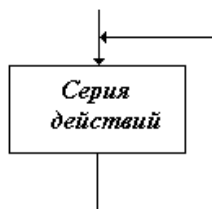


Рис.6. Простейший "бесконечный" цикл.

Это так называемый "бесконечный" цикл. Он фактически "неуправляем". Процесс, содержащий такой участок, никогда не закончится. С точки зрения практики обычного программирования таких участков следует избегать. А для этого надо уметь управлять циклом.

Возможны различные способы управления циклом. В основе любого из них лежит проверка некоторого условия, в зависимости от результата которой цикл либо завершается, либо продолжается. Проверяемое условие может быть сформулировано либо как условие завершения цикла, либо как условие продолжения цикла. В первом случае при выполнении условия осуществляется выход из цикла; если условие не выполняется, то цикл продолжается. Во втором случае при выполнении условия цикл продолжается; если условие не выполняется, то осуществляется выход из цикла. Таким образом, условия завершения и продолжения цикла являются противоположными: первое является отрицанием второго, и наоборот.

Проверка условия может осуществляться либо после выполнения серии действий, составляющей "тело" цикла, либо до выполнения серии действий. Различные способы управления циклом приведены на рис.7, 8

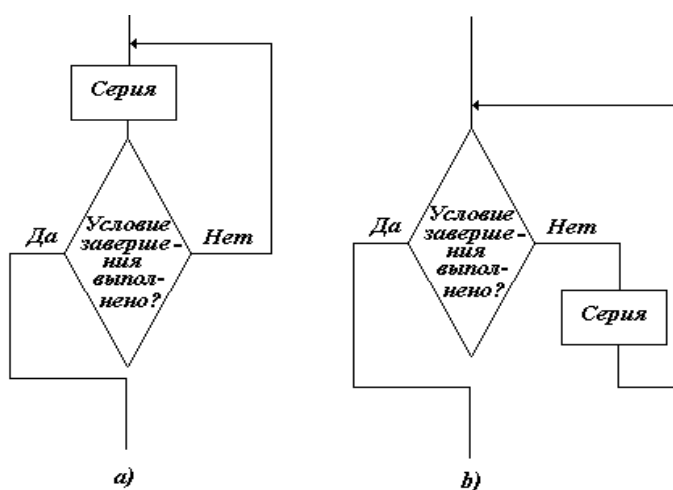


Рис. 7. Схемы управления циклом посредством условия завершения.

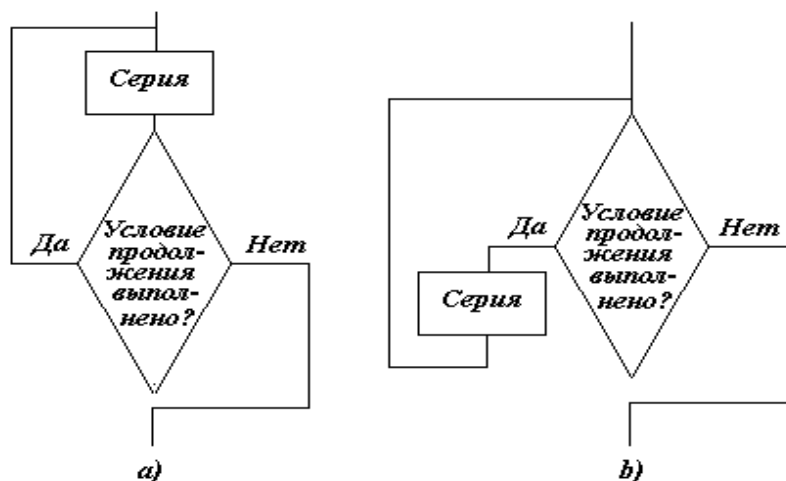


Рис. 8. Схемы управления циклом посредством условия продолжения

Выбор условий завершения или продолжения цикла определяется содержанием конкретного процесса обработки информации. Относительно любого цикла можно поставить вопрос: "Известно ли заранее (т.е. до начала цикла) число повторений цикла?". В зависимости ответа на этот вопрос строится та или иная стратегия управления циклом: управление циклом с заданным числом повторений или управление циклом, число повторений которого заранее неизвестно.

Можно показать, что любой из перечисленных способов управления циклом может быть сведен к любому, возможно, за счет некоторой избыточности. Чаще всего используется схема, приведенная на рис. 8.b. Она соответствует обороту "Пока выполняется условие Φ , выполняй действие F ".

Графические элементы, описывающие линейный процесс (следование), ветвление и цикл, называют **базовыми элементами**. Каждый из них характеризуется тем, что имеет **1** вход и **1** выход. Это дает возможность "конструировать" из этих элементов любые алгоритмы обработки информации путем взаимной подстановки одного элемента в другой по схеме: выход одного "подключается" к входу другого. Эта операция носит название **суперпозиции**. Посредством суперпозиции базовых элементов можно описать любой алгоритмический процесс. Это утверждение не может быть строго доказано, поскольку основывается на интуитивном, нестрогом понятии алгоритма.

ЛИТЕРАТУРА

1. Иванов Б.Н. Дискретная математика. Алгоритмы и программы.– М.: Лаборатория базовых знаний, 2002.
2. Нефедов В.Н., Осипова В.А. Курс дискретной математики. – М.: МАИ,1992.
3. Колесников Н.Г. Математические и логические основы информатики. – Краснодар: из-во КубГАУ, 2000.
4. Карпов В.Г., Мощенский В.А. Математическая логика и дискретная математика. – Минск: «Вышэйш. школа», 1977.
5. Кристофедис М. Теория графов. – М.: Мир,1978.
6. Оре О. Теория графов. – М.: Наука,1980.
7. Коршунов Ю.М. Математические основы кибернетики. – М.: Энергоатомиздат, 1987.
8. Виленкин Н.Я. Комбинаторика.-М.:Наука,1969.
9. Гаврилов Г.П., Сапоженко А.А. Сборник задач по дискретной математике.–М.: Наука,1977.

Аршинов Георгий Александрович
Аршинов Вадим Георгиевич

ДИСКРЕТНАЯ МАТЕМАТИКА
Курс лекций для бакалавров

Подписано в печать **23.04.08.** Бумага офсетная

Формат $60 \times 84^{1/16}$

Усл. печ. л. – 6,8

Тираж – 300

Заказ №

Типография Кубанского аграрного университета.
360044, г. Краснодар, ул. Клинина, 13.