



Co-funded by the
Tempus Programme
of the European Union



ШАРИПБАЙ А.А.

МАТЕМАТИКА ДЛЯ КОМПЬЮТЕРНЫХ НАУК
(Учебное пособие)

УДК 004.4(075.8)
ББК 32.973.26-018.1я73
Ш-25

Рецензенты:

Казиев Галым Зулхурнаевич – д.т.н., профессор. Директор исследовательского центра АО «Национальные информационные технологии».

Атанов Сабыржан Кубейсинович – д.т.н., профессор кафедры «Вычислительная техника и программное обеспечение» Евразийского национального университета имени Л.Н. Гумилева.

Ускенбаев Раиса Кабиевна – д.т.н., профессор, проректор Международного университета информационных технологий.

Автор:

Шарипбай А.А – директор НИИ «Искусственный интеллект», профессор кафедры «Информатики и Информационной безопасности» Евразийского национального университета имени Л.Н. Гумилева, доктор технических наук, профессор, академик МАИ, академик АПН РК, лауреат государственной премии РК.

Ш-25 Математика для компьютерных наук, Учебное пособие. –Астана, 2017. -101 с.

Содержание учебного пособия «Математика для компьютерных наук» составлено в соответствии с Силабусом одноименной обязательной дисциплины для профессиональной магистерской образовательной программы «Информатика как вторая компетенция» по специальности 6М060200-Информатика.

Учебное пособие предназначено для изучения математических основ науки информатики магистрантами указанной образовательной программы.

Учебным пособием могут пользоваться также студенты, магистранты и докторанты других специальностей, а также все те, кто самостоятельно желает изучить математическую основу информатики.

Учебное пособие издано на средства Международного проекта TEMPUS-544319-TEMPUS-1-2013-1-FR- TEMPUS-JPCR-(2013-4530001/00) “Profesional Master’s Degree in Computer Science as a second Competence in Central Asia”.

УДК 004.4(075.8)
ББК 32.973.26–018.1я73
ISBN 978-601-301-995-6

© Шарипбай А.А., 2017

Содержание

1. МНОЖЕСТВА	4
1.1. Определение множества.	4
1.2. Числовые множества.	5
1.3. Интервалы.	7
1.4. Операции над множествами.	11
1.5. Свойства операций над множествами.	14
2. НОТАЦИИ И КОДИРОВАНИЕ ИНФОРМАЦИИ.	21
2.1. Информация	21
2.2. Система счисления	24
3. ОСНОВЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ	35
3.1 Высказывания и логические связки.	35
3.2 Тавтологии и логические функции	37
4. ЛОГИЧЕСКИЕ ЗАКОНЫ	41
4.1. Тавтологий	41
4.2. Исчисление предикатов.	42
4.3. Логические основы компьютера	44
5. АЛГЕБРАИЧЕСКИЕ СТРУКТУРЫ НА МНОЖЕСТВЕ	50
5.1. Отношения	50
6. ГРАФЫ	59
6.1. Определения и способы представления графов	59
6.2. Деревья	63
8. ЯЗЫКИ И ГРАММАТИКИ	70
7.1. Формальное определение языка.	70
7.2. Формальные грамматики	75
8. АВТОМАТЫ	83
8.1. Абстрактный автомат	83
8.2. Конечный автомат	87
ЛИТЕРАТУРА	101

1. МНОЖЕСТВА

Цель: Рассмотреть понятие множества и элементы теории множеств, которые необходимы для определения фундаментальных понятий, структур данных и операций над ними, а также формализмы для описания и обработки объектов информатики.

План: Изучить множества и их виды, определить теоретико-множественные операции и их свойства, задать индикаторные функции и их свойства, ввести отношения над множествами.

1.1. Определение множества

Множество есть одно из первичных понятий математики, т. е. таких, которые лежат в основе логической системы и уже не определяются через другие понятия. Множество разработано в конце 19-го века. Теория множеств сейчас основная часть математики, и может быть использован в качестве основы, из которой почти может быть получена вся математика.

Интуитивно мы понимаем, что множество – это набор или совокупность объектов, объединенных каким-либо общим признаком (свойством). Объектами могут быть реальными физическими величинами или абстрактными математическими величинами, и называются *элементами* множества.

Элементы множества будут показаны в фигурных скобках $\{и\}$ без повторений. Например, числа 1, 2, и 3 есть различные объекты, когда рассматриваемые по отдельности, но когда они рассматриваются в совокупности они образуют единое множество размера три, записанное в виде $\{1,2,3\}$.

Имена множеств обозначается латинскими буквами и их элементов – малыми латинскими буквами или арабскими цифрами. В обоих случаях можно использовать индексы.

Запись $a \in A$ ($a \notin A$) означает, что элемент принадлежит (не принадлежит) множеству A . Например, пусть $A = \{1, 2, 3\}$: если $a=2$, то можно писать $a \in A$, а если $a=5$, то $a \notin A$.

1.2. Числовые множества

Если элементами множества являются только числовые значения, то такие множества называются *числовыми множествами*. Для удобства работы с числами их логически разбили на натуральные, целые, положительные целые, простые, рациональные, иррациональные, действительные.

1. *Натуральные числа*. Множество натуральных чисел обозначается буквой N . Натуральные числа являются самыми первыми числами, которыми люди начали пользоваться ещё в доисторические времена. Натуральные числа можно применить для счёта предметов и использовать в качестве их номеров. Наименьшее натуральное число – единица 1. Числа 9, 91, 847, 2 658 являются натуральными. Все вместе они составляют множество натуральных чисел, обозначаемое буквой N :

$$N = \{1, 2, 3, 4, \dots\}.$$

Но числа бывают не только натуральными. Индийцы изобрели число ноль и отрицательные числа. Теперь они для нас привычны, но когда-то европейцы – древние греки и римляне – долгое время обходились без нуля. Сейчас нам трудно это представить.

2. *Целые числа*. Множество целых чисел обозначается буквой Z :

$$Z = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\} = \{0, \pm 1, \pm 2, \pm 3, \dots\}.$$

Множество целых чисел состоит из трех подмножества отрицательных чисел Z^- , нуля 0 и подмножества положительных целых Z^+ , которое совпадает с множеством натуральных чисел, т.е. $Z^+ = N$. Таким образом множество целых чисел включает множество натуральных чисел.

3. *Положительные целые числа*. Множество положительных целых чисел обозначается буквой W . Это множество эквивалентно ранее определенному множеству Z^- . Так целое число является элементом множества положительных целых (или натуральных чисел) или нуля.

$$W = \{0, 1, 2, 3, 4, \dots\}.$$

4. *Целые простые числа.* Множество простых чисел обозначается буквой P . Простое число есть целое, которое делится только на самого себя и на единицу. Примерами простых чисел являются 3, 5, 7, 11, 13, 17.

5. *Рациональные числа.* Множество рациональных чисел обозначается буквой Q . Рациональное число – это дробное число, которое представляется в виде p/q (где p – целое, а q – натуральное). Например, $1/3$ – это «одна часть из трёх», $0,25$ – это двадцать пять сотых. Десятичные дроби также можно записать в виде p/q . Например, $0,25 = 25/100 = 1/4$. Рациональное число может иметь несколько различных дробных представлений. Например, $1/2$ эквивалентно $2/4$ или $132/264$. В десятичном представлении, рациональные числа принимают форму конечных или бесконечных периодических дробей. Целые числа (положительные и отрицательные) также можно записать в виде p/q , т.е. в виде дроби со знаменателем 1, например:

$$2 = 2/1, 0 = 0/1, -5 = -5/1.$$

Таким образом, множество рациональных чисел включает множество целых чисел.

6. *Иррациональные числа.* Множество иррациональных чисел обозначается буквой I . Любое вещественное число, которое не является рациональным является иррациональным. Эти числа могут быть записаны в десятичном виде, но не в дробях. Они являются бесконечными непериодическими десятичными дробями. Некоторые примеры иррациональных чисел:

$$\pi = 3.1415926535..., \sqrt{2} = 1.4142135623..., \sqrt{7} = 2.6457513110...$$

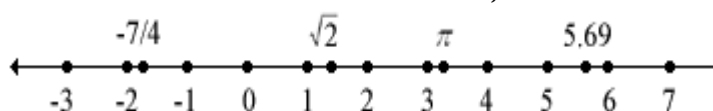
Замечание. Любой корень, который не совершенный корень иррациональное число. Таким образом, любые корни, такие как следующие примеры, иррациональны.

$$\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{6}, \sqrt{7}, \sqrt{8}, \sqrt{10}, ..., \sqrt[3]{2}, \sqrt[3]{13}, \sqrt[5]{20}, \sqrt[9]{10002229}$$

7. *Действительные (Вещественные) числа.* Множество действительных чисел обозначается буквой R . Каждое число (за

исключением комплексных чисел) содержится в множестве действительных чисел. Когда общий термин "число" используется, он относится к действительному числу. Все следующие типы или числа, также могут рассматриваться в качестве действительных чисел.

8. *Вещественно–числовая прямая.* Каждое вещественное число может быть связано с одной точки на *вещественно–числовой прямой*



1.3. Интервалы

Интервал – это есть множество, которое состоит из всех действительных чисел между данной парой чисел. Она также может рассматриваться как сегмент вещественно–числовой прямой. Конечная точка интервала является одной из двух точек, которые отмечают концов сегмента.

Интервал может включать в себя либо одну конечную точку, обе конечные точки или никакой конечной точки. Чтобы различать эти разные промежутки времени, мы используем интервальное обозначения.

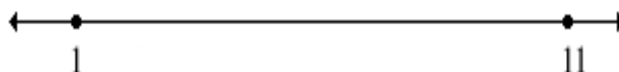
Открытый интервал не включает в себя конечные точки. Исключение из конечных точек указывается круглыми скобками (,) в интервальной нотации. Когда интервал представлен сегментом вещественно–числовой прямой, исключение конечной точкой показано открытой точкой. Например, интервал чисел между числами 3 и 8, за исключением 3 и 8, записывается как $(3, 8) = \{x: 3 < x < 8\}$ в интервале нотации. В сегменте реального числовой прямой, то были бы представлены линии ниже.



Закрытый интервал включает в себя конечные точки. Включение конечных указывается в квадратных скобках [] в интервальной нотации. Когда интервал представлен сегментом

вещественно–числовой прямой, включение конечной точкой показано замкнутой точкой. Например, интервал чисел между числами 1 и 11, в том числе как 1 и 11, записывается как $[1, 11] = \{x: 1 \leq x \leq 11\}$ в интервале нотации.

В сегменте вещественно–числовой прямой, это были бы представлены линии ниже.



Один из концов интервала может быть включен, в то время как другой исключен. Интервал $[a, b)$ представляет все числа между a и b , в том числе a , но не b . Аналогично, интервал $(a, b]$ будет представлять все числа между a и b , в том числе b , но не a . Эти интервалы представлены более подробно в таблице ниже.

Бесконечными интервалами являются те, которые не имеют конечную либо в положительном или отрицательном направлении, или в обоих. Интервал расширяется всегда в этом направлении. Бесконечные интервалы приведены в таблице ниже.

Interval Notation	Number Line Sketch	Set-builder Notation
(a, b)		$\{x \mid a < x < b\}$
$(a, b]$		$\{x \mid a < x \leq b\}$
$[a, b)$		$\{x \mid a \leq x < b\}$
$[a, b]$		$\{x \mid a \leq x \leq b\}$
(a, ∞)		$\{x \mid x > a\}$
$(-\infty, b)$		$\{x \mid x < b\}$
$[a, \infty)$		$\{x \mid x \geq a\}$
$(-\infty, b]$		$\{x \mid x \leq b\}$
$(-\infty, \infty)$		R

В математике, (действительный) интервал есть множество действительных чисел с тем свойством, что любое число, которое лежит между двумя числами в множестве также включено в множество. Например, множество всех чисел x , удовлетворяющих $0 \leq x \leq 1$ интервал, который содержит 0 и 1, а также все числа между ними.

Другими примерами интервалов являются множество всех действительных чисел R , множество всех отрицательных чисел, и пустое множество.

Интервалы также определяется на произвольной линейно упорядоченном множестве, например, на множестве целых чисел или рациональных чисел.

Конечные точки интервала.

Интервал чисел между a и b , в том числе a и b , часто обозначается $[a, b]$. Числа a и b называются конечными точками интервала.

Чтобы показать, что один из концов должен быть исключен из множества, соответствующие квадратные скобки могут быть либо заменены в скобках, или наоборот:

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

$$[a, b) = [a, b[= \{x \in R: a \leq x < b\},$$

$$(a, b] =]a, b] = \{x \in R: a < x \leq b\},$$

$$[a, b] = [a, b] = \{x \in R: a \leq x \leq b\}.$$

Заметим, что (a, a) , $[a, a)$ и $(a, a]$ каждый представляет пустое множество, в то время как $[a, a]$ обозначает множество $\{a\}$. При $a > b$, все четыре обозначения, как правило, принято представлять пустое множество.

1. *Бесконечные концы.* В обоих стилях записи, можно использовать бесконечную конечную точку, чтобы указать, что нет границы в этом направлении. В частности, можно использовать $a = -\infty$ или $b = +\infty$ (или оба). Например, $(-\infty, 0)$, $(0, +\infty)$ и $(-\infty, +\infty)$ представляют множество всех отрицательных, множество всех

положительных и множество всех действительных чисел, соответственно.

2. *Интервалы целое число.* Обозначение $[a .. b]$, когда a и b целые числа, или $\{a .. b\}$ или просто $a .. b$ иногда используется, чтобы указать интервал всех целых чисел между a и b , включая обоих. Это обозначение используется в некоторых языках программирования.

Число интервал, который имеет конечный нижний или верхний конечную всегда включает в себя, что конечную точку. Таким образом, исключение конечных можно явно обозначить как $a .. b - 1$, $a + 1 .. b$ или $a + 1 .. b - 1$. Альтернативный–кронштейн обозначения, как $[a .. b)$ или $[a .. b[$ редко используются для целочисленных интервалов.

Открытый интервал не включает свои конечные точки и указывается в круглых скобках. Например $(0,1)$ означает, что конечная точка больше, чем 0 и меньше 1. Закрытый интервал включает в себя свои конечные точки и обозначается в квадратных скобках. Например $[0,1]$ означает, что конечная точка больше или равно 0 и меньше или равно 1.

3. *Классификация интервалов.* Интервалы действительных чисел можно разделить на одиннадцать различных типов, перечисленных ниже; где a и b действительные числа, $a < b$:

- пусто: $[b, a] = (a, a) = [a, a) = (a, a] = \{\} = \emptyset$,
- вырождение: $[a, a] = \{a\}$,
- *собственно и ограничена:*
- открыт: $(a, b) = \{x \mid a < x < b\}$,
- закрыт: $[a, b] = \{x \mid a \leq x \leq b\}$,
- слева закрыта, справа открыта: $[a, b) = \{x \mid a \leq x < b\}$,
- открыто слева, справа закрыт: $(a, b] = \{x \mid a < x \leq b\}$,
- *слева ограничена и правой неограниченной:*
- открыто слева: $(a, \infty) = \{x \mid x > a\}$,
- левый закрыт: $[a, \infty) = \{x \mid x \geq a\}$,
- левой и правой неограниченной ограниченных:

- правый открыт: $(a, \infty) = \{x \mid x > a\}$,
- закрыто справа: $(-\infty, b] = \{x \mid x \leq b\}$,
- неограниченная на обоих концах: $(-\infty, +\infty) = R$.

В некоторых контекстах, интервал может быть определен как подмножество расширенных действительных чисел, множество всех действительных чисел, дополненных $-\infty$ и $+\infty$. В этой интерпретации, обозначения $[-\infty, b]$, $[-\infty, b)$, $[a, +\infty]$ и $(a, +\infty]$ все значимые и различны. В частности, $(-\infty, +\infty)$ обозначает множество всех простых действительных чисел, в то время как $[-\infty, +\infty]$ обозначает расширенные действительные.

Этот выбор влияет на некоторые из приведенных выше определений и терминов. Например, интервал $(-\infty, +\infty) = R$ замкнуто в сфере обычных действительных, но не в сфере расширенных действительных чисел.

1.4. Операции над множествами

Пусть нам даны два множества A и B , а затем над ними может быть определена следующая операции:

- (1) *Объединение* состоит из элементов A или B , записывается как

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

- (2) *Пересечение* состоит из элементов A и B , записывается как

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

- (3) *Разность* состоит из элементов множества A и не включает элементов B , записывается как

$$A \setminus B = \{x: x \in A \ \& \ x \notin B\}.$$

- (4) *Симметрическая разность* состоит только из элементов A или только из элементов B , записывается как

$$A \Delta B = \{x: (x \in A \ \& \ x \notin B) \vee (x \in B \ \& \ x \notin A)\}$$

- (5) *Дополнение* состоит из элементов универсума U и не включает элементов A , т.е.,

$$\overline{A} = \{x \mid x \in U \ \& \ x \notin A\}.$$

Графическую интерпретацию операций объединение, пересечение, разность, симметрическая разность и дополнение можно представить с помощью диаграммы Эйлера–Венна как на рисунке 1.1.

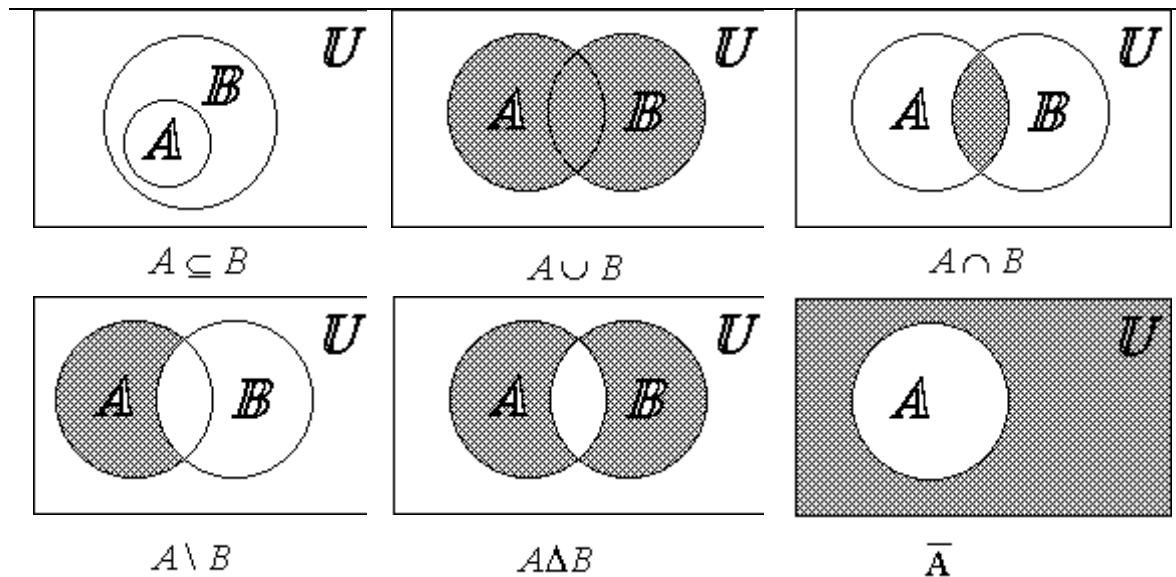


Рисунок 1.1. Диаграммы Эйлера–Венна.

Здесь универсум U изображается прямоугольником, а множества A и B – окружностями. Для выделения результата применяется штриховка.

(6) *Прямое произведение* состоит из всевозможных упорядоченных пар элементов A и B , записывается как

$$A \times B = \{(a, b): a \in A \text{ \& } b \in B\}$$

В которой универсум U изображается прямоугольником, а множества A и B – окружностями. Для выделения результата применяется штриховка.

Здесь показано, что множества A и B являются подмножествами U , и они записываются как $A \subseteq U$ и $B \subseteq U$.

Между множествами чисел имеются следующие отношения:

$$P \subseteq N \subseteq W \subseteq Z \subseteq Q \subseteq R.$$

Свойства интервалов. Пересечение любой коллекции интервалов всегда интервал. Объединение двух интервалов интервал, если и только если они имеют непустое пересечение или

открытая конечная точка одного интервала есть закрытая конечная точка другого

$$((a,b) \cup [b,c] = (a,c]).$$

Количество элементов множества A обозначается через $|A|$ и называется *мощностью* множества A .

Среди всех множеств есть два специальные множества:

1. $\emptyset$ – пустое множество, не содержащее ни одного элемента.
2. U – универсальное множество (универсум), содержащее все элементы этого типа.

Относительно теории универсум это множество, содержащее в качестве элементов все объекты, рассматриваемые в этой теории.

Например, универсумом является:

- 1) в теории чисел – множество всех целых чисел;
- 2) в теории языков – множество всех слов в заданном алфавите;
- 3) в геометрии – множество всех точек n -мерного геометрического пространства.

Если количество элементов множества конечно (существует неотрицательное целое число k , равное количеству элементов этого множества), то оно называется *конечным множеством*, в противном случае оно называется *бесконечным множеством*.

В частности, пустое множество является конечным множеством, количество элементов которого равно нулю, т.е. $|\emptyset|=0$.

В дальнейшем мы будем рассматривать только конечные множества. Множество всех подмножеств конечного множества A с k элементами записывается как $2^k = \{X \mid X \subseteq A\}$.

Если конечные множества $X_1, \dots, X_n$ попарно не пересекаются ($X_i \cap X_j = \emptyset$), то

$$|X_1 \cup X_2 \cup \dots \cup X_n| = |X_1| + |X_2| + \dots + |X_n|.$$

Если $X_1, \dots, X_n$ – конечные множества, то

$$|X_1 \times X_2 \times \dots \times X_n| = |X_1| \times |X_2| \times \dots \times |X_n|.$$

Индикаторной (характеристической) функцией для множества A называется функция $I_A(x)$ и определяется как:

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

Таким образом, $I_A: U \rightarrow \{0,1\}$.

Для $A \subseteq U$ и $B \subseteq U$ имеют место следующие свойства:

$$I_A(x) = I_B(x) \Leftrightarrow A = B;$$

$$I_A(x) \leq I_B(x) \Leftrightarrow A \subseteq B;$$

$$I_{\bar{A}}(x) = 1 - I_A(x);$$

$$I_{A \cup B}(x) = I_A(x) + I_B(x) - I_A(x) \cdot I_B(x);$$

$$I_{A \cap B}(x) = I_A(x) \cdot I_B(x);$$

$$I_{A \setminus B}(x) = I_A(x) - I_A(x) \cdot I_B(x);$$

$$I_{A \setminus B}(x) = I_A(x) - I_A(x) \cdot I_B(x);$$

$$I_{A \Delta B}(x) = I_A(x) + I_B(x) - 2I_A(x) \cdot I_B(x).$$

Индикаторы удобно задавать с помощью таблицы 1.1.

Таблица 1.1. Индикаторы.

$x \in A$	$x \in B$	$x \in A \cup B$	$x \in A \cap B$	$x \in A \setminus B$	$x \notin A$	$x \in A \Delta B$
0	0	0	0	0	1	0
0	1	1	0	0	1	1
1	0	1	0	1	0	1
1	1	1	1	0	0	0

1.5. Свойства операций над множествами

Операции над множествами обладают следующими свойствами:

I. Объединение, пересечение и разность:

- 1) $A \cup \emptyset = A$ – свойство нуля;
- 2) $A \cup A = A$ – идемпотентность;
- 3) $A \cup B = B$, если все элементы A содержатся в B ;
- 4) $A \cup B = B \cup A$ – коммутативность;
- 5) $(A \cup B) \cup C = A \cup (B \cup C) = A \cup B \cup C$ – ассоциативность;
- 6) $A \cap \emptyset = \emptyset$ – свойство нуля;
- 7) $A \cap A = A$ – идемпотентность;

- 8) $A \cap B = A$, если A все элементы A содержатся в B ;
- 9) $A \cap B = B \cap A$ – коммутативность;
- 10) $(A \cap B) \cap C = A \cap (B \cap C) = A \cap B \cap C$ – ассоциативность;
- 11) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ – дистрибутивность;
- 12) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ – дистрибутивность;
- 13) $A \cap (B \setminus C) = (A \cap B) \setminus (A \cap C)$ – дистрибутивность;
- 14) $A \cup \bar{A} = U$ – свойство дополнения;
- 15) $A \cap \bar{A} = \emptyset$ – свойство дополнения;
- 16) $\overline{A \cup B} = \bar{A} \cap \bar{B}$ – закон де Моргана;
- 17) $\overline{A \cap B} = \bar{A} \cup \bar{B}$ – закон де Моргана;
- 18) $\overline{\bar{A}} = A$ – инволютивность;
- 19) $A \setminus \emptyset = A$ – свойство разности;
- 20) $A \setminus A = \emptyset$ – свойство разности;
- 21) $A \setminus B = A \cap \bar{B} = \emptyset$ – свойство разности;
- 22) $B \setminus A = B \cap \bar{A} = B \setminus (B \cap A)$ – свойство разности;

II. Симметрическая разность и прямое произведение:

- 1) $A \Delta \emptyset = A$ – свойство нуля;
- 2) $A \Delta A = \emptyset$ – идемпотентность;
- 3) $A \Delta B = (A \cup B) \setminus (A \cap B)$ – свойство симметрической разности;
- 4) $A \Delta B = B \Delta A$ – коммутативность;
- 5) $(A \Delta B) \Delta C = A \Delta (B \Delta C) = A \Delta B \Delta C$ – ассоциативность;
- 6) $(A \cup B) \times C = (A \times C) \cup (B \times C)$ – дистрибутивность;
- 7) $A \times (B \cup C) = (A \times B) \cup (A \times C)$ – дистрибутивность;
- 8) $(A \cap B) \times C = (A \times C) \cap (B \times C)$ – дистрибутивность;
- 9) $A \times (B \cap C) = (A \times B) \cap (A \times C)$ – дистрибутивность;
- 10) $(A \setminus B) \times C = (A \times C) \setminus (B \times C)$ – дистрибутивность;
- 11) $A \times (B \setminus C) = (A \times B) \setminus (A \times C)$ – дистрибутивность.

Пусть заданы два множества A и B одного и того же типа.

Тогда мы можем вести следующие отношения:

- 1) $A=B$: A равно B , если A и B состоят из одних и тех же элементов, т.е. A и B являются подмножествами друг друга;

2) $A \subseteq B$: A содержится в B , если все элементы A принадлежат B или A равно B , это означает, что A является подмножеством B ;

3) $A \subset B$: A строго содержится в B , если все элементы A принадлежат B и A не равно B , т.е. некоторые элементы B не принадлежат A , это означает, что A является собственным подмножеством B .

Аналогично можно определить отношения включает $A \supseteq B$ и строго включает $A \supset B$.

Нетрудно заметить, что выше введенные отношения $=$, $\subseteq$ и $\subset$ являются подмножествами прямого произведения $A \times B$.

Таким образом, можно считать, что любое отношение – это некоторое подмножество в прямом произведении, выделяемое определенным законом.

Замечание 1.1. Пустое множество $\emptyset$ является собственным подмножеством любого конечного множества.

Множества могут быть описаны различными способами: по перечислению элементов, по нотации застройщика множества, по интервальной записи, путем построения графика на числовой прямой и / или диаграммах Венна.

По перечислению элементов: Перечисление элементов – это список элементов в множестве, разделенных запятыми и окруженных фигурными фигурные скобки. Например:

1) $\{2,3,4,5,6\}$ является список для множества чисел от 2 до 6 включительно;

2) $\{1,2,3,4, \dots\}$ является список для множества натуральных чисел, где три точки указывают, что их число продолжает в той же схеме на неопределенное количество;

3) набор маленьких латинских букв, обозначающий гласные звуки английского языка описывается как $\{a, e, i, o, u\}$.

По нотации застройщика множества: Нотация застройщика множества есть математическое сокращение для точного заявления всех элементов определенного множества, которые обладают

специфическим свойством. Кроме того, можно использовать двоеточие (:), чтобы представлять слова "итак, что". Например:

1) Утверждение "все x , которые являются элементами множества целых чисел, таких, что x находится между 2 и 6 включительно". $\{x \in \mathbb{Z} : 2 \leq x \leq 6\}$ есть множество чисел от 2 до 6 включительно, $\mathbb{Z}$ – множество целых чисел;

2) Утверждение "все n , которые являются элементами множества натуральных чисел, таких, что n меньше, чем числа 100". $\{n \in \mathbb{N} : n < 100\}$ есть множество натуральных чисел, меньших, чем число 100, $\mathbb{N}$ – множество целых чисел;

3) Утверждение "все x , которые являются элементами множества целых чисел, таких, что, их значения больше чем 0, положительное". $\{x \in \mathbb{Z} \mid x > 0\}$ есть множество всех положительных целых чисел.

По интервала записи: интервал связное подмножество чисел. Интервал обозначения альтернатива выражения ответ в виде неравенства. Если не указано иное, мы будем работать с вещественными числами.

Примеры 1.

1. Пусть $A = \{a, b, c, d, e, f\}$, $B = \{c, d\}$, то:

1) $A \cup B = \{a, b, c, d, e, f\}$;

2) $A \cap B = \{c, d\}$;

3) $B \times B = \{(c, c), (c, d), (d, c), (d, d)\}$;

4) $A \setminus B = \{a, b, e, f\}$;

5) $A \Delta B = \{a, b, f\}$.

6) $\bar{A}$ зависит от того, какой будет универсум U . Допустим, если $U = \{a, b, c, d, e, f, h\}$, то $\bar{A} = \{h\}$.

2. Пусть $A = \{1, 2\}$, $B = \{a, b\}$, $C = \{+, -\}$. Тогда дистрибутивность операции объединения $(A \cup B) \cup C = A \cup (B \cup C) = A \cup B \cup C$ определяется как:

$$(\{1, 2\} \cup \{a, b\}) \cup \{+, -\} = \{1, 2\} \cup (\{a, b\}) \cup \{+, -\} =$$

$$\{1,2\} \cup \{a,b\} \cup \{+,-\}.$$

Упражнения 1.1.

1. Пусть $A=\{1, 2, 4\}$, $B=\{3, 4, 5, 6\}$, $C = \{5, 7\}$ Выполните эти:

- 1) $A \times C$;
- 2) $A \cup \bar{A}$;
- 3) $A \Delta B$;
- 4) $(A \cap B) \times C$;
- 5) $(A \setminus B) \times C$.

2. Пусть задан универсум $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$ и его подмножества $A = \{x \mid 2 < x \leq 6\}$, $B = \{x \mid x - \text{четно}\}$, $C = \{x \mid x \geq 4\}$, $D = \{1, 2, 3\}$. Найти множества $A \cup B$, $C \times D$, $B \Delta C$, $(A \setminus B) \cup (C - D)$.

3. Курс лекции по экономике посещают 20 студентов, по математике – 30. Найти число студентов, посещающих лекции по экономике или математике, если:

- а) лекции проходят в одно и то же время;
- б) лекции проходят в разные часы и 10 студентов слушают оба курса.

Вопросы 1.1:

1. Что такое множество?
2. Что такое универсальное множество?
3. Как выглядит диаграмма Эйлера–Венна для объединения?
4. Как определяется прямое произведение множеств?
5. Как определяется симметричная разность множеств?
6. Как определяется индикаторная функция множества?
7. В чем заключается закон де Моргана?
8. Если $A=\{1,2,3\}$, $B=\{3,4\}$, то что означает $\{1,2,3,4\}$?
9. Если $A=\{1,2,4\}$, $B=\{4,3,2\}$, то что означает $\{2,4\}$?
10. Если $A=\{1,2,3,4\}$, $B=\{3,4,5,6\}$, то что означает $\{2,4\}$?
11. Выяснить, какие из следующих дистрибутивных законов справедливы для любых множеств A, B, C :
 - 1) $A \setminus (B \cup C) = (A \setminus B) \cup (A \setminus C)$;
 - 2) $A \setminus (B \cap C) = (A \setminus B) \cap (A \setminus C)$;

- 3) $A \Delta (B \cup C) = (A \Delta B) \cup (A \Delta C)$;
- 4) $A \Delta (B \cup A) = (A \Delta B) \cup (A \Delta C)$;
- 5) $A \setminus (B \Delta A) = (A \setminus B) \Delta (A \setminus C)$;
- 6) $A \cup BC = (A \cup B)(A \cup C)$;
- 7) $A \cup (B \setminus C) = (A \cup B) \setminus (A \cup C)$;
- 8) $A(B \setminus C) = AB \setminus AC$;
- 9) $A \cup (B \Delta C) = (A \cup B) \Delta (A \cup C)$;
- 10) $A(B \Delta C) = AB \Delta AC$;
- 11) $A \Delta (B \setminus C) = (A \Delta B) \setminus (A \Delta C)$?
12. Сколько элементов в каждом из множеств:
 - 1) $\{1, 2, 3, \{1, 2, 3\}\}$;
 - 2) $\{\emptyset\}$;
 - 3) $\{1, \{1\}, 2, \{1, 2, 3\} \emptyset\}$;
 - 4) $\emptyset$;
 - 5) $\{\emptyset, \{\emptyset\}\}$;
 - 6) $\{\{\emptyset, \{\emptyset\}\}\}$;
 - 7) $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$?

Тесты I.1:

1. Определить $|L|$, если L состоит из латинских строчных букв,
 - A) 1
 - B) 26
 - C) 28
 - D) 0
 - E) 30
2. Если $D = \{d \mid d - \text{целое число и } 0 \leq d \leq 9\}$, то чему равна $|D|$?
 - A) d
 - B) 9
 - C) 10
 - D) 0
 - E) 1
3. Какой результат получится после выполнения операции $A \cap B$ для заданных множеств $A = \{0, 2, 4, 6\}$ и $B = \{-2, -1, 0, 1, 2\}$?
 - A) $\{0, 2\}$

- B) $\{0, 2, 4, 6\}$
- C) $\{-2, -1, 0, 1, 2\}$
- D) $\{0, 2, 4, 6, -2, -1, 0, 1, 2\}$
- E) $\emptyset$

4. Какой результат получится после выполнения операции $A \cup B$ для заданных множеств $A = \{1, 3, 5\}$ и $B = \{2, 4, 6, 8\}$?

- A) $\{1, 2, 3, 4, 5, 6, 8\}$
- B) $\{1, 3, 5\}$
- C) $\{2, 4, 6, 8\}$
- D) $\emptyset$
- E) $\{1, 3, 8\}$

5. Какой результат получится после выполнения операции $A \setminus B$ для заданных множеств $A = \{a, б, в, г, д, е, ж\}$ и $B = \{г, е, ж\}$?

- A) $\{a, б, в, д\}$
- B) $\{a, б, в, г, д, е\}$
- C) $\{г, е, ж\}$
- D) $\{г, е\}$
- E) $\emptyset$

2. НОТАЦИИ И КОДИРОВАНИЕ ИНФОРМАЦИИ

Цель: дать правила нотации (обзначения) и кодирования информации для компьютерного хранения и обработки данных.

План: изучить способы и системы представления величин с помощью цифр, букв и других символов.

2.1. Информация

Известно, что информацию можно считать отражением свойств и отношений материальных и нематериальных объектов и субъектов окружающего мира. Информацию можно обозначать и понимать, т.е. каждая информация должна иметь свою форму и свое содержание.

Обычно для обозначения (нотации) информации используются некоторые символы или их последовательности. При этом человек различает символы по их начертаниям, а компьютер – по их кодам, состоящим из последовательности 0 и 1, так как физические устройства хранения информации в компьютере (ячейки памяти и регистры) могут находиться только в двух состояниях, которым соотносят 0 или 1. Используя ряд подобных физических устройств, можно хранить в памяти компьютера любую информацию с помощью двоичного кода в виде последовательности 0 и 1. Поэтому нотация любой (числовой, текстовой, графической, звуковой и др.) информации, с которой работает современные компьютеры, кодируется (преобразуется) в двоичный код, и декодируется (обратно преобразуется) в нотацию для облегчения восприятия человеком. Двоичный код информации хранится в оперативной памяти, а при вывода символа на принтер или на экран компьютера происходит декодирование, т.е. преобразование кода символа в его изображение.

Символьные данные представляются в памяти компьютера с помощью последовательности образованных из 0 и 1 кодов, входящих в их состав символов. Код каждого символа задается значениями индексов так называемой кодовой таблицей двумерного

массива, содержащего этот символ. В кодовой таблице порядок кодирования называют стандартом кодирования. То есть, каждый стандарт определяет свою кодовую таблицу. К таким стандартам относятся широко распространенные ASCII, ANSI, Unicode и другие. В стандарте ASCII каждый символ занимает 8 бит, т.е. 1 байт. Из-за того, что ASCII был предназначен для западных языков, его использование было ограничено в странах и регионах, чьи языки содержали символы, не включенные в $2^8 = 256$ символов, поддерживаемых ASCII.

Чтобы обойти это ограничение, Международная организация по стандартизации (ISO - International Standards Organization) создала новый стандарт кодировки символов, названный Latin-1, который содержал европейские символы, не вошедшие в набор ASCII. Microsoft расширила Latin-1 и назвала этот стандарт ANSI. Но ANSI по-прежнему осталась 8-битной кодировкой, которая может представлять только 256 уникальных символов. Многие языки имеют тысячи символов, особенно языки, такие как китайский, корейский и японский.

Для преодоления ограничений стандарта 8-битовой кодировки символов, Microsoft в сотрудничестве с такими компаниями, как Apple Computer, Inc., и IBM, создала некоммерческий консорциум Unicode целью которого стало определение нового стандарта на кодировку символов для международных наборов символов. Работа, проделанная в Unicode, была объединена с работой, которая велась в ISO, и результатом стал стандарт Unicode для кодировки символов.

Unicode является 16-битным стандартом, что дает возможность кодировки $2^{16} = 65\,536$ разных друг от друга символов. Поэтому с помощью этого стандарта можно кодировать символы алфавита любого национального языка. Он поддерживает даже архаические языки, такие как древнетюркские руники, санскрит и египетские иероглифы, и включает знаки препинания, математические и графические символы. В его таблице кодирования имеются коды

применяющихся в различных отраслях науки знаков и различных декоративных знаков.

Первая версия Unicode (1991 г.) представляла собой 16-битную кодировку с фиксированной шириной символа. Во второй версии Unicode (1996 г.) было решено значительно расширить кодовую область; для сохранения совместимости с теми системами, где уже был реализован 16-битный Unicode, и был создан UTF (Unicode Transformation Format)-16, являющийся одним из способов кодирования символов из Unicode в виде последовательности 16-битных слов. Данная кодировка позволяет записывать символы Unicode. Один символ кодировки UTF-16 представлен последовательностью двух байтов. Который из двух идёт впереди, старший или младший, зависит от порядка байтов. Для определения порядка байтов используется метка порядка байтов.

Традиционно для кодирования одного символа используется количество информации, равное 1 байту, то есть $I = 1 \text{ байт} = 8 \text{ битов}$. Если рассматривать символы как возможные события, то можно вычислить какое количество разных символов N можно закодировать в одном байте:

$$N = 2^I = 2^8 = 256 \text{ битов.}$$

Таким образом, каждому символу ставится в соответствие уникальный десятичный код от 0 до 255 или соответствующий ему двоичный код от 00000000 до 11111111.

При вводе в компьютер символьной информации происходит ее двоичное кодирование, код символа хранится в оперативной памяти и занимает 1 байт, а при вывода символа на принтер или на экран компьютера происходит декодирование, т.е. преобразование кода символа в его изображение. В общем смысле кодирование информации можно определить, как перевод информации, представленной сообщением в первичном алфавите, в последовательность кодов. Надо понимать, что любые данные – это так или иначе закодированная информация. Информация может быть представлена в разных формах: в виде чисел, текста, рисунка,

звука и др. Перевод из одной формы в другую – это кодирование. Надо понимать, что любые данные – это так или иначе закодированная информация. Информация может быть представлена в разных формах: в виде чисел, текста, рисунка и др. Перевод из одной формы в другую – это кодирование.

Если значение величины определяется во время построения общих правил интерпретации языка общения, то это значение будет постоянной, иначе значение – переменной.

Определение 2.1. *Нотация* есть:

1) система фигур или символов, используемых в специализированной области для представления чисел, количества, тона или значения;

2) акт или процесс с использованием такой системы.

2.2. Система счисления

В информатике обработка числовых величин может потребовать различные системы счисления чисел. Основаниями таких систем могут быть 2, 3, 4,

Определение 2.2. *Система счисления* — это способ записи чисел с помощью цифр и множества правил. Есть несколько способов записи чисел с помощью цифр.

Любая система счисления удовлетворяет следующие правила:

- наличие возможности записи значений чисел в заданном интервале;
- каждая последовательность цифр определяет только одно числовое значение;
- простота выполнения операций.

Все системы счисления подразделяются на: *позиционные системы счисления* и *непозиционные системы счисления*.

1. В позиционной системе счисления значения цифр зависят от их позиции в записи числа. Если в записи числа одна и та же цифра встречается несколько раз, то она определяет разное значение. Например, в трехзначном числе 333 самая левая цифра 3 определяют

три сотни, средняя цифра 3 – три десятки, а самая правая цифра 3 – три единицы.

2. В непозиционной системе счисления значения цифр не зависят от их места в записи числа. Например, запись числа с помощью римских цифр: в записи числа LXXXVIII – восемьдесят восемь цифра L обозначает пятьдесят, X – десять, V – пять, I – один.

Позиционная система счисления характеризуется своей основой. Основа определяет число цифр, применяемых в этой системе. Например, число цифр в десятичной системе равно десять, в восьмиричной системе – восемь, а в двоичной системе – два и т.п.

В любой позиционной системе счисления с основой q заданное число A можно представить следующим образом:

$$A_{(q)} = a_{n-1}q^{n-1} + \Lambda + a_1q^1 + a_0q^0 + a_{-1}q^{-1} + \Lambda + a_{-m}q^{-m} \quad (1)$$

где a_i – число цифр, применяемых в системе счисления, n – число разрядов в целой части, m – число разрядов в дробной части ($i=n-1, \dots, 1, 0, -1, \dots, -m$).

Среди них, мы заинтересованы в десятичной системе счисления и двоичной системе счисления. В таблице 2.1 даются эквивалентные десятичные и двоичные цифры.

Таблица 2.1. Эквивалентные десятичные и двоичные цифры.

Десятичная цифра	Двоичная цифра
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

Из этой таблицы видно, что запись одного числового значения в различных системах счисления требуют разное количество цифр. Например, десятичное число 16 в двоичной системе будет 10000.

Для перевода заданного целого числа из системы счисления с основой p на систему счисления с основой q нужно несколько раз делить это число на q пока остаток не станет меньше, чем q . Полученное частное взять в качестве самого старшего разряда числа с основой q , а в качестве значений остальных разрядов нужно взять остатки в направлении, начиная с последнего остатка до первого остатка и образовать цепочку слева направо.

Для перевода заданного правильного дробного числа с основой p на основу q нужно несколько раз умножать это число на q , пока значение разряда дробной части не будет равны нулю или до получения заданной точности. В качестве значения разрядов правильной дроби с новой основой q нужно образовать цепочку слева направо в направлении, начиная с первой появившейся целой части до последней появившейся целой части.

Например, перевод числа 25 в десятичной системе счисления на двоичную систему будет так:

$$\begin{array}{r}
 25 \quad | \quad 2 \\
 \hline
 -24 \quad | \quad 12 \quad | \quad 2 \\
 \hline
 1 \quad | \quad 12 \quad | \quad 6 \quad | \quad 2 \\
 \hline
 \quad | \quad 0 \quad | \quad 6 \quad | \quad 3 \quad | \quad 2 \\
 \hline
 \quad | \quad \quad | \quad 0 \quad | \quad 2 \quad | \quad 1 \\
 \hline
 \quad | \quad \quad | \quad \quad | \quad \quad | \quad 1
 \end{array}$$

То есть, $25_{(10)} = 11001_{(2)}$. Для проверки правильности число 11001 разложим в соответствии с выражением (1) следующим образом:

$$11001_{(2)} = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 16 + 8 + 0 + 0 + 1 = 25_{(10)}$$

.

Выполнение операций и их свойства в десятичной системе счисления и в двоичной системе счисления похожи. Свойства этих операций одинаковы со свойствами операций над десятичными числами.

Примеры 2.1:

1. Четырехзначное число 1952 десятичной системы выражается так:

$$1952_{(10)} = 1 \cdot 10^3 + 9 \cdot 10^2 + 5 \cdot 10^1 + 2 \cdot 10^0.$$

2. Число десятичной системы с трехзначной целой частью и трехзначной дробной частью $596.174_{(10)}$ выражается так:

$$596.174_{(10)} = 5 \cdot 10^2 + 9 \cdot 10^1 + 6 \cdot 10^0 + 1 \cdot 10^{-1} + 7 \cdot 10^{-2} + 4 \cdot 10^{-3}.$$

4. Перевод дробного числа 0.625 в десятичной системе счисления в двоичную систему счисления будет выглядеть так:

	0, 625
	* 2
	250
	* 2
	500
	* 2
	000
	* 2
	000

↓

Итак полученный результат такой: $0.625_{(10)} = 0.1010_{(2)}$.

3. Число двоичной системы с четырехзначной целой частью и трехзначной дробной частью $1010.101_{(2)}$ выражается так:

$$1010.101_{(2)} = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}.$$

Возможности всех позиционных систем счисления одинаковы. Разница между ними только в методах обозначения значений чисел, а виды операции над числами и их свойства будут одинаковыми.

Однако среди них десятичная система счисления является наиболее распространенной. Поэтому мы сначала рассмотрим хорошо знакомую нам десятичную систему счисления чисел,

операции над ними и свойства этих операций, так как они подходят для других систем счисления чисел.

Целые числа будут представлены арабскими цифрами, перед их отрицательными значениями будет написан знак "-", и перед их положительными значениями может быть написан знак "+".

Вещественные числа в зависимости от их способа представления делятся на две группы: действительные числа с фиксированной точкой и с плавающей точкой.

Представление вещественных чисел с фиксированной точкой состоит из целой и дробной части. Целая часть помещается слева от дробной части, и они отделены друг от друга запятой ".".

Перед ними для указания положительности или отрицательности их значений записывается «+» или «-». Обе части представляются арабскими цифрами.

Представление действительных чисел с плавающей точкой состоит из частей, называемых мантиссой, основой системы счисления и порядка.

Представление действительных чисел с плавающей точкой состоит из частей, называемых мантиссой, основой системы счисления и порядка.

Если обозначить мантиссу через M , порядок через p , основу системы счисления через q , то действительные числа будут следующими: $M * q^p$.

Чтобы понять, в таблице 2.2 рассмотрены примеры действительных чисел с плавающей точкой.

Таблица 2.2. Примеры действительных чисел с плавающей точкой.

№	Пример	Мантисса	Порядок	Значение
1.	$-12. * 10^3$	-12	3	-12000
2.	$0.3 * 10^{+2}$	0.3	2	30
3.	$254 * 10^{-2}$	254	-2	2.54
4.	$1.5 * 10^1$	1.5	1	15
5.	$+ 2.17 * 10^2$	2.17	+2	217

Одно и то же действительное число с плавающей точкой могут быть представлены различными способами. Например, то же количество 3.14 может быть записана:

$$314. \cdot 10^{-2} = 31.4 \cdot 10^{-1} = 3.14 \cdot 10^0 = 0.314 \cdot 10^1 = 0.0314. \cdot 10^2 = K$$

Чтобы иметь одну запись для представления вещественного числа с плавающей точкой, мы должны нормализовать его следующим условием:

$$q^{-1} \leq |M| < 1,$$

где $|M|$ – абсолютная величина.

Например, действительные числа с плавающей точкой и в нормализованном виде будут такими:

$$0.1364 \cdot 10^4 \text{ и } 0.617 \cdot 10^{-7}.$$

С целью упрощения арифметических операций в компьютере применяют специальные коды для представления чисел. Мы рассмотрим *прямой код*, *обратный код* и *дополнительный код* чисел.

Прямой код двоичного числа – это само двоичное число, а знак числа записывается двоичной цифрой: знак «–» – цифрой 1, знак «+» – цифрой 0. Например, отрицательное двоичное число -1011_2 в прямом коде запишется как 1.1011.

Представление чисел в компьютере по сравнению с формами, известными всем со школы, имеет два важных отличия:

- числа записываются в двоичной системе счисления;
- для записи и обработки чисел отводится конечное количество разрядов (в обычной – некомпьютерной арифметике такое ограничение отсутствует).

Сложение и умножение двоичных чисел производится согласно таблице сложения и умножения:

Сложение двоичных чисел	Умножение двоичных чисел
$0 + 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$
$1 + 1 = 10$	$1 \cdot 1 = 1$

В компьютере арифметическое устройство выполняет действия не с самими двоичными числами по правилам двоичной арифметики, а с их двоичными кодами по правилам арифметики двоичных кодов.

Отличия правил арифметики двоичных кодов от правил обычной арифметики заключаются в ограниченности разрядной сетки. Иначе говоря, для записи числа в памяти компьютера выделяется фиксированное количество двоичных разрядов. Память компьютера имеет байтовую структуру, однако, размер одной адресуемой ячейки обычно составляет несколько байт: 2, 4, 8 байт.

Вся информация в компьютере представляется в двоичных кодах. Из всего множества кодов мы рассмотрим прямой, обратный и дополнительные коды.

Для записи целого двоичного числа в прямом коде двоичные числа дополняются знаковым разрядом, который принимается равным "0" для положительных чисел и "1" – для отрицательных. При ручной записи чисел со знаком, знаковый разряд, для удобства, отделяется от значащих разрядов точкой.

Например, десятичное число (+12) в прямом двоичном коде запишется как (0.1100), а десятичное число – так (–12) – (1.1100).

Прямой код используется при хранении чисел в памяти ЭВМ, а также при выполнении операций умножения и деления.

Другими формами представления чисел со знаком являются обратный и дополнительный коды. Эти коды позволяют заменить вычитание целых чисел их сложением, исходя из принципа: $a - b = a + (-b)$.

Положительные числа, записанные в прямом, обратном и дополнительном кодах одинаковы.

Так, положительное десятичное число 12 в прямом, обратном и дополнительном двоичном кодах запишется: (0.1100).

Для перевода отрицательного числа из прямого кода в обратный следует в знаковом разряде сохранить единицу, а цифры значащих разрядов инвертировать, т.е. "1" заменить на "0", а "0" на "1".

Дополнительный код отрицательного числа получается из обратного кода числа прибавлением "1" к младшему разряду этого числа.

Правила сложения в дополнительном коде:

1. Сложение производится по правилам сложения двоичных чисел, включая знаковый разряд.

2. Если в результате сложения возникает перенос (переполнение) из знакового разряда, этот перенос игнорируется (отбрасывается).

3. Если знак суммы не совпадает со знаками слагаемых (эта ситуация может возникнуть только когда знаки одинаковы), имеет место переполнение разрядной сетки компьютера и результат должен быть признан неверным.

Сложение в обратном двоичном коде отличается от сложения в дополнительном коде лишь одним правилом: если в результате сложения возник перенос из знакового разряда, т.е. произошло переполнение, необходимо к младшему разряду суммы прибавить "1".

Примеры 2.1.

1. +5 – положительное целое число 5.

2. 3.14 – положительное действительное число с фиксированной точкой, целая часть 3, а дробная часть 14.

3. 0.2 – положительное действительное число с фиксированной точкой, целая часть 0, а дробная часть 2.

4. –1.001 – отрицательное действительное число с фиксированной точкой, целая часть 1, а дробная часть 001.

5. 0.0 – положительное действительное число, целая часть 0 и дробная часть 0.

Пример 2.2. Записать десятичное число (–12) в прямом, обратном и дополнительном двоичном кодах в шестirazрядной ячейке:

1) 01100 – прямой код;

2) 10011 – обратный код;

3) 10100 – дополнительный код.

В данном примере один разряд отведен под знак числа, пять разрядов под само число, под точку в разрядной сетке место не выделяется. Само число сдвинуто к правому краю, а в избыточный разряд (в прямом коде) записан «0». Затем прямой код инвертируется для перевода в обратный.

Перевод чисел из обратного (дополнительного) кода в прямой код производится по тем же правилами, что и в обратный (дополнительный) код из прямого.

Пример 2.3. Реализовать операцию: $15 - 7$ в прямом, обратном и дополнительном коде:

	10-е число	Прямой код	Обратный код	Дополнительный код
Данные	15 — 7	0.1111 — 1.0111	0.1111 + 1.1000	0.1111 + 1.1001
Промежуточный результат	8		10.0111 + 1	1 0.1000
Окончательный результат	8		0.1000	0.1000

Пример 2.4. Реализовать операцию: $7 - 15$ в прямом, обратном и дополнительном коде:

	10-ичное число	Прямой код	Обратный код	Дополнительный код
Данные	—15 +7	0.1111 1.0111	1.0000 + 0.0111	1.0001 + 0.0111
Промежуточный результат	—8		1.0111	1.1000
Окончательный результат	—8		1.1000	1.0111 + 1 1.1000

Задания 2.

1. Определить действительные числа с плавающей точкой и комплексные числа:

- 1) 40,23;
- 2) -5;
- 3) $3.3 \cdot 10^{-2}$;
- 4) $5.1 + 6i$;
- 5) $0.14 + 7i$.

2. Переведите заданное число из одной системы счисления в другую:

- 1) 10000001 из двоичной системы в десятичную систему.
- 2) 129 из десятичной системы в восьмиричную систему.
- 3) 1952 из десятичной системы в шестнадцатеричную систему.

3. Расставьте знаки арифметических операций так, чтобы было верно следующее равенство в двоичной системе: $1100 ? 11 ? 100 = 100000$.

Вопросы 2.

- 1. Что является системой счисления?
- 2. На какие группы подразделяются все системы счисления?
- 3. Можно ли одно и то же числовое значение представить в различной системе счисления?
- 4. Какие типы имеются у числовых величин?
- 5. На какие группы подразделяются действительные числа?

Тесты 2.

- 1. В какой системе кодируются данные в ANSI?
 - A) в двоичной системе
 - B) в троичной системе
 - C) в восьмиричной системе
 - D) в десятичной системе
 - E) в шестнадцатеричной системе
- 2. В какой системе кодируются данные в Unicode?
 - A) в шестнадцатеричной системе

- В) в троичной системе
- С) в восьмиричной системе
- Д) в десятичной системе
- Е) в двоичной системе

3. Сколько байтов используется для кодировки в Unicode?

- А) 2
- В) 1
- С) 3
- Д) 5
- Е) 4

4. Что такое система счисления?

А) Способ записи чисел с помощью цифр и множества правил.

В) Наличие возможности записи значений чисел в заданном интервале.

С) Каждая последовательность цифр определяет только одно числовое значение.

Д) Простота выполнения операций.

Е) Значения цифр не зависят от их места в записи числа.

5. Какая система счисления является самой маленькой?

- А) Двоичная.
- В) Восьмиричная.
- С) Шестнадцатиричная.
- Д) Троичная.
- Е) Десятичная.

3. ОСНОВЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ

Цель: дать математические обозначения, доказуемость математических суждений, природу математического доказательства в целом, вычислимость и прочие аспекты оснований математики.

План: изучить основы формальных систем, высказывания и логические связки, формулы, тавтологии.

3.1 Высказывания и логические связки

Применение в логике математических методов становится возможным тогда, когда суждения формулируются на некотором точном языке. Такие точные языки имеют две стороны: синтаксис и семантику. Синтаксисом называется совокупность правил построения объектов языка (обычно называемых формулами). Семантикой называется совокупность соглашений, описывающих наше понимание формул (или некоторых из них) и позволяющих считать одни формулы верными, а другие – нет.

Важную роль в математической логике играют понятия дедуктивной теории и исчисления. Исчислением называется совокупность правил вывода, позволяющих считать некоторые высказывания – формулы выводимыми. Формулы образуются с помощью логических связей.

Высказывание (утверждение) есть повествовательное предложение, в отношении которого можно сказать, истинно оно или ложно.

Содержание любой науки составляют высказывания об объектах ее предметной области. Логика высказываний абстрагируется от конкретного содержания утверждений и изучает структуру сложных высказываний и их логические связи.

Примеры высказываний: “Снег белый”, “ $2 > 3$ ” и т.д.

Высказывания могут связываться с друг другом с помощью логических связей "не", "и", "или", "импликация", "эквивалентно".

Математическую логику мы будем изучать с помощью метаязыка который отличается от предметного языка изучаемой логики. Предметный язык логики высказываний состоит из алфавита и формул:

Алфавит:

(1) $P, Q, R, \dots$ – переменные для простых высказываний (пропозициональные буквы);

(2) $\neg, \&, \vee, \rightarrow, \leftrightarrow$ – символы операций над высказываниями (логические связки);

(3) $(,)$ – вспомогательные символы (скобки).

Формулы, или сложные высказывания:

(1) $P, Q, R, \dots$ – пропозициональные буквы – элементарные формулы (атомы);

(2) если A, B – формулы, то $\neg A, A\&B, A\vee B, A\rightarrow B, A\leftrightarrow B$ – формулы.

В определении формул использованы *метабуквы* A, B , т.е. символы, не принадлежащие предметному языку.

Примеры формул: $\neg P, (P\&Q), (R\rightarrow (P\vee R))$.

Подформула – это часть формулы, сама являющаяся формулой.

Задав язык, мы построили формальную систему. Представим теперь ее как содержательную алгебру высказываний, для чего придадим смысл символам алфавита и формулам. Пропозициональные буквы и логические операции определим на области из двух элементов $\{И, Л\}$, И – Истина, Л – Ложь:

P	Q	P&Q	P∨Q	¬P	P→Q	P↔Q
И	И	И	И	Л	И	И
И	Л	Л	И	Л	Л	Л
Л	И	Л	И	И	И	Л
Л	Л	Л	Л	И	И	И

Значение формулы $E [P_1, \dots, P_n]$ при данной интерпретации входящих в нее пропозициональных букв $\gamma: \{P_1, \dots, P_n\} \Rightarrow \{И, Л\}$ мы определим индукцией по построению формулы:

$$E = P : E[\gamma] = \gamma(P);$$

$$E = \neg A : E[\gamma] = \neg A[\gamma];$$

$$E = A \& B : E[\gamma] = (A \& B)[\gamma] = A[\gamma] \& B[\gamma];$$

$$E = A \vee B : E[\gamma] = (A \vee B)[\gamma] = A[\gamma] \vee B[\gamma];$$

Если в формуле операция $\neg$ применяется только к атомам, то такая формула *называется формулой с тесными отрицаниями*.

3.2 Тавтологии и логические функции

Тавтология – эта формула, истинная при всех интерпретациях входящих в нее пропозициональных букв, другими словами, – столбец значений которой содержит одни истинные значения (обозначается $\models$).

Основные тавтологии.

$$1a. \models A \rightarrow (B \rightarrow A)$$

$$1б. \models (A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$$

$$2. \models A \rightarrow (B \rightarrow A \& B)$$

$$3a. \models A \& B \rightarrow A$$

$$3б. \models A \& B \rightarrow B$$

$$4a. \models A \rightarrow A \vee B$$

$$4б. \models B \rightarrow A \vee B$$

$$5. \models (A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow (A \vee B \rightarrow C))$$

$$6. \models (A \rightarrow C) \rightarrow ((A \rightarrow \neg C) \rightarrow \neg A)$$

$$7. \models \neg \neg A \rightarrow A$$

$$8. \models (A \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow (A \leftrightarrow B))$$

$$9a. \models (A \leftrightarrow B) \rightarrow (A \rightarrow B)$$

$$9б. \models (A \leftrightarrow B) \rightarrow (B \rightarrow A)$$

$$10. \models (A \rightarrow (\neg A \rightarrow C))$$

Логическая функция есть n –местная операция на множестве $\{0,1\}$.

Алфавит:

- (1) $x, y, x_1, x_2, \dots$ – предметные переменные;
- (2) $f, g, \dots, f_1, f_2, \dots$ – функциональные символы.

Терм:

- (1) $x, y, \dots, x_1, x_2, \dots$ – предметные переменные являются термами;
- (2) если $f^{(n)}$ – функциональный символ, $t_1, \dots, t_n$ – термы, то $f^{(n)}(t_1, \dots, t_n)$ – терм.

Значение терма:

- (1) если t – предметная переменная x , то $\text{Val } t = \gamma(x)$;
- (2) если $t = f^{(n)}(t_1, \dots, t_n)$, то $\text{Val } t = f^{(n)}(\text{Val } t_1, \dots, \text{Val } t_n)$.

Функция:

$f^{(n)}(x_1, \dots, x_n)$ *представима термом* $t(v_1, \dots, v_m)$, если $\{v_1, \dots, v_m\} \subseteq \{x_1, \dots, x_n\}$ и $t[\gamma] = f^{(n)}[\gamma]$ для всех интерпретаций

$$\gamma: \{x_1, \dots, x_n\} \Rightarrow \{0,1\}.$$

Примеры 3. Примерами высказываний являются следующие повествовательные предложения;

- 1) До издания этой книги я не читал её;
- 2) Число три меньше, чем пять;
- 3) Не верно, что компьютер умнее человека;
- 4) Программист не работал или компьютер был сломан;
- 5) Он посещал занятия и успешно сдал экзамен;
- 6) Если стекло разбито, то оно не восстановится;
- 7) Он говорит тогда и только тогда, когда услышит.

Задания 3.

- 1. Сравните истинностные таблицы формул $\neg P \vee Q$ и $P \rightarrow Q$.
- 2. Найдите тавтологии среди следующих формул:
 - a) $(P \& \neg P) \rightarrow (Q \vee R \rightarrow (R \rightarrow \neg P))$;
 - b) $(P \rightarrow \neg P) \sim \neg P$;

3. Вычислив только одну строку таблицы истинности, найдите формулы, не являющиеся тавтологиями:

- a) $P \vee Q \rightarrow P \& Q$;
- b) $(P \rightarrow Q) \rightarrow (Q \rightarrow P)$.

4. Найдите формулу, у которой в столбце значений стоит только ложь.

5. Найти значение формулы $(\neg B \rightarrow A)[\gamma]$, если имеются $(A \vee B)[\gamma] = И$ и $(A \rightarrow B)[\gamma] = И$.

Вопросы 3.

- 1. Будет ли тавтологией формула $(\neg P \vee Q) \& (R \rightarrow (P \leftrightarrow Q))$?
- 2. Существуют ли такие высказывания А, В и С, чтобы для них одновременно выполнялись условия в некоторой интерпретации γ $(A \rightarrow B)[\gamma] = И$; $(B \vee C)[\gamma] = Л$; $(B \leftrightarrow (A \& \neg C))[\gamma] = Л$?

3. Эквивалентны ли соединяющие высказывания:

$A \supset B$ и если А, то В?

$A \supset B$ и коль скоро А, то В?

$A \supset B$ и в случае А имеет место В?

$A \supset B$ и А влечет В?

Тесты 3.

1. Какое значение будет иметь выражение $2 > 5 \vee 2 < 6$?

- A) 2
- B) 1
- C) 5
- D) 6
- E) 0

2. Какой порядок выполнения операций в выражении $D \vee \neg F \wedge G$?

- A) сначала $\neg F$, затем $\neg F \wedge G$, а в конце $D \vee \neg F \wedge G$.
 - B) сначала $\neg F \wedge G$, а в конце $D \vee \neg F \wedge G$.
 - C) сначала $\neg F$, а в конце $D \vee \neg F \wedge G$.
 - D) сначала $\neg F$, затем $\neg F \wedge G$.
 - E) сначала $\neg G$, затем $\neg F \wedge G$, а в конце $D \vee \neg F \wedge G$.
3. Какой из них закон Де Моргана?

- A) $\neg(\neg p) \equiv p$
- B) $p \equiv p$
- C) $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- D) $p \wedge \neg p \equiv 0$
- E) $p \vee \neg p \equiv 1$

4. Что такое тавтология?

- A) Формула, истинная при всех интерпретациях входящих в нее пропозициональных букв.
- B) Столбец значений которой содержит одни ложные значения.
- C) Формула, которая может быть истинной или ложной.
- D) Формула, которая может быть только ложной.
- E) Формула, которая при любой интерпретации является неопределенной.

5. Что такое формула с тесными отрицаниями?

- A) Формула, в которой операция $\neg$ применяется только к атомам.
- B) Формула, ложная при всех интерпретациях.
- C) Формула, истинная при всех интерпретациях.
- D) Формула, неопределенная при всех интерпретациях.
- E) Формула, в которой операция $\neg$ применяется всем подформулам.

4. ЛОГИЧЕСКИЕ ЗАКОНЫ

Цель: дать понятия и определения логических законов, рассмотреть их виды.

План: изучить закон двойного отрицания, коммутативной закон, дистрибутивный закон, закон исключения констант, закон противоречия, закон исключенного третьего, принцип двойственности, логическое следствие, правила логического вывода, правило модус поненс.

4.1. Тавтологий

Логические законы состоят из следующих тавтологий

- 1) $\models A \vee \neg A$ (закон исключённого третьего)
- 2) $\models A \rightarrow A$ (закон тождества)
- 3) $\models \neg(A \vee B) \sim \neg A \& \neg B$ (1-й закон де Моргана)
- 4) $\models \neg(A \& B) \sim \neg A \vee \neg B$ (2-й закон де Моргана)
- 5) $\models A \& A \sim A, \models A \vee A \sim A$
- 6) $\models A \rightarrow B \sim \neg A \vee B$
- 7) $\models (A \leftrightarrow B) \sim (A \rightarrow B) \& (B \rightarrow A)$
- 8) $\models (A \rightarrow B) \sim (\neg B \rightarrow \neg A)$ (закон контрапозиции)
- 9) $\models A \& B \sim B \& A$ (коммутативность конъюнкции)
- 10) $\models A \vee B \sim B \vee A$ (коммутативность дизъюнкции)
- 11) $\models A \& (B \& C) \sim (A \& B) \& C$ (ассоциативность конъюнкции)
- 12) $\models A \vee (B \vee C) \sim (A \vee B) \vee C$ (ассоциативность дизъюнкции)
- 13) $\models A \& (B \vee C) \sim (A \& B) \vee (A \& C)$ (1-й закон дистрибутивности)
- 14) $\models A \vee (B \& C) \sim (A \vee B) \& (A \vee C)$ (2-й закон дистрибутивности)
- 15) $\models A \& (A \vee B) \sim A, \models A \vee (A \& B) \sim A$ (законы поглощения)
- 16) $\models A \& И \sim A, \models A \& Л \sim Л, \models A \vee И \sim И, \models A \vee Л \sim А.$
- 17) $\models A \rightarrow (B \rightarrow C) \sim A \& B \rightarrow C.$

Пусть E формула с тесными отрицаниями, не содержащая других операций, кроме $\neg$, $\&$, $\vee$. Формула E^X есть результат замены в E везде конъюнкции на дизъюнкцию, дизъюнкции на конъюнкцию и каждой пропозициональной буквы на ее отрицание. Тогда $\models \neg E \sim E^X$.

Принцип двойственности. Пусть формулы E, F не содержат других операций, кроме $\neg$, $\&$, $\vee$ и являются формулами с тесными отрицаниями. Формулы E', F' , полученные из E, F одновременной заменой всюду $\&$ на $\vee$ и $\vee$ на $\&$, называются двойственными к формулам E, F , соответственно. Тогда имеют место утверждения:

- а) Если $\models \neg E$, то $\models E'$. б) Если $\models E$, то $\models \neg E'$.
 в) Если $\models E \sim F$, то $\models E' \sim F'$. г) Если $\models E \rightarrow F$, то $\models F' \rightarrow E'$.

Логическое следствие. Пусть заданы формулы $A_1, A_2, \dots, A_m$ и B . Если из одновременной истинности формул $A_1, A_2, \dots, A_m$ следует истинность формулы B , то формула B является логическим следствием формул $A_1, A_2, \dots, A_m$ и обозначается $A_1, A_2, \dots, A_m \models B$, ($m \geq 1$), здесь $A_1, A_2, \dots, A_m$ – посылки, а B – заключение.

Правила логического вывода. В исчислении высказываний используется единственное правило вывода, называемое *modus ponens*, и представляющее собой процедуру перехода от двух формул вида $A, A \rightarrow B$ (посылок) к формуле B (заключению):

$$\frac{A, A \rightarrow B}{B} \quad (\text{modus ponens})$$

Требования, которым должны удовлетворять правила вывода, – из истинных посылок должны получаться истинные заключения.

4.2. Исчисление предикатов

Предикаты это логические функции $J^{(n)}(x_1, \dots, x_n)$, заданные на непустой области D и принимающие значения во множестве $\{И, Л\}$.

Предикат $J^{(n)}(x_1, \dots, x_n)$ становится высказыванием после означивания входящих в него переменных на элементах множества D .

Алфавит:

- (1) $x, y, z, \dots, x_1, x_2, \dots$ – предметные переменные;
- (2) $P^{(n)}(x_1, \dots, x_n), \dots$ – предикатные буквы ($n=0,1, \dots$);
- (3) $\&, \vee, \neg, \rightarrow, \leftrightarrow, \forall, \exists$ – логические связки и кванторы;
- (4) $(,)$ – вспомогательные символы.

Формулы:

- (1) $P^{(n)}(x_1, \dots, x_n), \dots$ – элементарные формулы или атомы;
- (2) если A, B – формулы, то $A \& B, A \vee B, \neg A, A \rightarrow B, A \leftrightarrow B$ – формулы;
- (3) если $A(x)$ – формула со свободной переменной x , то $\forall x A(x), \exists x A(x)$ – формулы.

Свободные и связанные переменные. Переменные, находящиеся в области действия квантора по этой переменной называются *связанными*, иначе – *свободными*.

Интерпретация формулы. Значение формулы $E[P_1, \dots, P_m; x_1, \dots, x_n]$ при интерпретации предикатных букв $\tau: P^{(n)} \Rightarrow J^{(n)}$ и означивании $\gamma: \{x_1, \dots, x_n\} \Rightarrow D$ ($D \neq \emptyset$) предметных переменных, обозначается $E[\tau, \gamma]$, определим индукцией по построению формулы E :

- 1) $E = P^{(n)}(x_1, \dots, x_n)$, то $E[\tau, \gamma] = J[\gamma]$;
- 2) $E = (A \& B)[P_1, \dots, P_m; x_1, \dots, x_n]$, то $E[\tau, \gamma] = A[\tau, \gamma] \& B[\tau, \gamma]$.

Аналогично для остальных логических связок.

- 3) $E = \forall x_1 A[P_1, \dots, P_m; x_1, \dots, x_n]$, то $E[\tau, \gamma] = \forall x_1 A[\tau, x_1, \gamma] = I$, где $\gamma: \{x_2, \dots, x_n\} \Rightarrow D$, если $A[\tau, a, \gamma] = I$ для любого $a \in D$.

- 4) $E = \exists x_1 A[P_1, \dots, P_m; x_1, \dots, x_n]$, то $E[\tau, \gamma] = \exists x_1 A[\tau, x_1, \gamma] = I$, где $\gamma: \{x_2, \dots, x_n\} \Rightarrow D$, если $A[\tau, a, \gamma] = I$ для некоторого $a \in D$.

Формула $E[P_1, \dots, P_m; x_1, \dots, x_n]$ называется *общезначимой* или

тавтологией, если для любой области $D \neq \emptyset$, для любых интерпретаций τ предикатных букв и любом означивании γ предметных переменных в области D , $E[\tau, \gamma] = И$.

4.3. Логические основы компьютера

Логические основы компьютера состоит из алгебры логики, которая возникла в середине XIX века в трудах английского математика Джорджа Буля. Ее создание представляло собой попытку решать традиционные логические задачи алгебраическими методами, используя логические операциями являются $\neg$, $\&$, $\vee$, обозначающие слова и словосочетания "не", "и", "или". С помощью этих логических операций можно построить любое сложное логическое выражение.

Аппаратная реализация указанных логических операций осуществляется следующими логическими элементами компьютера, которые представлены на рисунке 4:

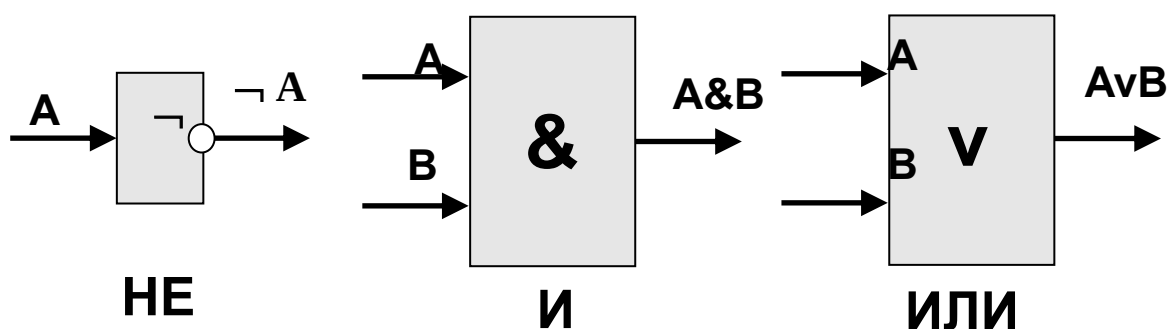


Рисунок 4. Логические элементы компьютера.

Примеры 4.1.

Покажем, что формула $P(x, y) \rightarrow Q(x)$ не 1–общезначима, следовательно, не общезначима.

Решение. $D = \{1\}$ – одноэлементная область, I_1 и I_2 – интерпретации буквы P , а J_1 и J_2 – интерпретации буквы Q :

x	y	I_1	I_2	J_1	J_2
1	1	И	Л	И	Л

Истинностная таблица формулы $P(x, y) \rightarrow Q(x)$:

x	y	$P(x, y)$	$Q(x)$	$P(x, y) \rightarrow Q(x)$
1	1	И	И	И
1	1	И	Л	Л
1	1	Л	И	И
1	1	Л	Л	И

Примеры 4.2.

Покажем, что формула $\forall x \exists y P(x, y) \rightarrow \exists y \forall x P(x, y)$ не общезначима.

Решение. Пусть $D = \{1, 2\}$, тогда интерпретации предикатной буквы $P(x, y)$ зададим следующей таблицей:

X	Y	J_1	J_2	J_3	J_4	...	J_7	...
1	1	И	И	И	И	...	И	...
1	2	И	И	И	И	...	Л	...
2	1	И	И	Л	Л	...	Л	...
2	2	И	Л	И	Л	...	И	...

В частности, для интерпретации J_7 получим: при $x=1$: $\exists y J_7(1, y) \equiv \text{И}$; при $x=2$: $\exists y J_7(2, y) \equiv \text{И}$, тогда $\forall x \exists y J_7(x, y) = \text{И}$. При $y=1$: $\forall x J_7(x, 1) = \text{Л}$, при $y=2$: $\forall x J_7(x, 2) = \text{Л}$, тогда $\exists y \forall x J_7(x, y) = \text{Л}$. Откуда $\forall x \exists y J_7(x, y) \rightarrow \exists y \forall x J_7(x, y) = \text{Л}$.

Примеры 4.3.

Покажем, что формула $\forall x (\exists x P(x) \rightarrow P(x))$ не 2-общезначима.

Решение. $D = \{1, 2\}$, J_1, J_2, J_3, J_4 – интерпретации буквы P :

	J_1	J_2	J_3	J_4
x				
1	И	И	Л	Л
2	И	Л	И	Л

Истинностная таблица формулы $x (\exists x P(x) \rightarrow P(x))$:

x	$P(x)$	$\exists x P(x)$	$\exists x P(x) \rightarrow P(x)$	$\forall x (\exists x P(x) \rightarrow P(x))$
1	J_1	И	И	И
2	J_1		И	
1	J_2	И	И	Л
2	J_2		Л	
1	J_3	И	Л	Л
2	J_3		И	
1	J_4	Л	И	И
2	J_4		И	

Примеры 4.4.

Пусть P – ложное высказывание $1 = 5$, Q – тоже ложное высказывание $3 = 7$ и R – истинное утверждение $4 = 4$. Показать, что условные высказывания: «если P , то Q » и «если P , то R », – оба истинны.

Решение. Если $1 = 5$, то, прибавляя 2 к обеим частям равенства, мы получим, что $3 = 7$. Следовательно, высказывание «если P , то Q » справедливо. Вычтем теперь из обеих частей равенства $1 = 5$ число 3 и придем к $-2 = 2$. Поэтому $(-2)^2 = 2^2$, т. е. $4 = 4$. Таким образом, «если P , то R » тоже верно.

Задания 4.

1. Переведите каждое из следующих рассуждений в логическую символику и проанализируйте результат на правильность:

1) Я бы заплатил за работу по ремонту телевизора, только если бы он стал работать. Он же не работает. Поэтому я платить не буду.

2) Если бы он ей не сказал, она ни за что не узнала бы. А не спроси она его, он бы и не сказал ей. Но она узнала. Значит, она его спросила.

3) Он сказал, что придет, если не будет дождя. Но идет дождь. Значит, он не придет.

2. Проверьте правильность рассуждения: Иванов не сделает эту работу, если ее сделает Петров. Петров и Сидоров сделают эту работу в том и только том случае, если ее сделает Иванов. Сидоров, эту работу сделает, а Иванов нет. Следовательно, Петров не сделает эту работу.

3. Выводами из каких формул являются следующие последовательности формул: $A \supset (B \supset C)$, A , $B \supset C$, B , C .

Вопросы 4.

1. Будут ли эквивалентными следующие высказывания

- 1) $A \wedge B$ и A и B ?
- 2) $A \wedge B$ и не только A , но и B ?
- 3) $A \wedge B$ и B , хотя и A ?
- 4) $A \wedge B$ и B , несмотря на A ?
- 5) $A \wedge B$ и как A , так и B ?

2. Будут ли эквивалентными следующие высказывания

- 1) $A \vee B$ и A или B ?
- 2) $A \vee B$ и A или B ?
- 3) $A \vee B$ и A , если не B ?
- 4) $A \vee B$ и A и B ?
- 5) $A \vee B$ и A или B ?

3. Будут ли эквивалентными следующие высказывания

- 1) $A \sim B$ и A , если и только если B ?
- 2) $A \sim B$ и если A , то B , и обратно?
- 3) $A \sim B$ и A , если B , и B , если A ?
- 4) $A \sim B$ и A равносильно B ?
- 5) $A \sim B$ и A тогда и только тогда, когда B ?

4. Для какого из указанных значений X : $X=1$, $X=6$, $X=5$, $X=3$, $X=4$ истинно высказывание $(X > 3) \& (X < 5)$?

5. Для каких заданных слов «Информатика», «Психология», «Экономика» будет истинно высказывание «Первая буква согласная и вторая буква согласная»?

6. Какие из следующих высказываний истинны, а какие ложны?

- (a) Сумма внутренних углов любого треугольника равна 180° .
- (b) У всех кошек есть хвост.
- (c) Найдется целое число x , удовлетворяющее $x^2 = 2$.
- (d) Существует простое четное число.
- (e) Снег белый.
- (f) Земля вращается вокруг Луны.
- (g) Париж столица Франции.
- (h) Управлять, значит знать.

Тесты 4.

1. Чем характеризуется закон «исключения третьего»:

1) то есть импликация двух высказываний эквивалентна обратной импликации их отрицаний?

2) то есть для любого высказывания имеет место одно из двух, либо оно истинно, либо ложно, третье места не имеет?

3) то есть всякое высказывание есть логическое следствие самого себя?

4) то есть отрицать отрицание некоторого высказывания это все равно, что утверждать это высказывание?

2. Что такое интерпретация – это:

1) понятия, применение которых к выражениям логического исчисления существенно зависит от выбора интерпретации?

2) это отображение, сопоставляющее каждому элементарному высказыванию p некоторое значение истинности?

3) понятия, применение которых к выражениям логического исчисления существенно зависит от выбора интерпретации?

4) отношение между объектами, проявляющиеся в том, что состояния или свойства любого из них меняются при изменении состояния и свойств других?

3. Какой является логическая связка «или»:

1) соединительной?

2) исключаящей?

3) разделительной?

4) вспомогательной?

5) отрицающий?

4. Чем характеризуется закон «двойного отрицания»:

1)то есть для любого высказывания имеет место одно из двух, либо оно истинно, либо ложно, третье места не имеет?

2)то есть всякое высказывание есть логическое следствие самого себя?

3)то есть отрицать отрицание некоторого высказывания это все равно, что утверждать это высказывание?

4)то есть всякое высказывание есть логическое следствие самого себя?

5. АЛГЕБРАИЧЕСКИЕ СТРУКТУРЫ НА МНОЖЕСТВЕ

Цель: дать определения понятия отношения и рассмотреть различные виды отношений и их свойства: рефлексивность, симметричность, транзитивность, антисимметричность.

План: изучить отношение эквивалентности, отношения частичного порядка, бинарные отношения, бинарные операции, свойства бинарных операций: ассоциативность, коммутативность, дистрибутивности, рассмотреть алгебраические структуры на множестве, изоморфизм и группы.

5.1. Отношения

Любое отношение, заданное между множествами, является некоторым подмножеством прямого произведения, выделяемым определенным законом.

Пусть заданы два множества X и Y , и пусть $R \subset X \times Y$ – подмножество их декартова произведения. Тогда тройка (X, Y, R) называется бинарным отношением между X и Y .

Утверждение $(x, y) \in R$ обычно записывается в виде xRy и читается " x соотносится с y ."

Например, между множествами X , Y можно определить следующие бинарные отношения:

4) $X = Y$: X равно Y , если X и Y состоят из одних и тех же элементов, т.е. X и Y являются подмножествами друг друга;

5) $X \subseteq Y$: X содержится в Y , если все элементы X принадлежат Y или X равно Y , это означает, что X является подмножеством Y ;

6) $X \subset Y$: X строго содержится в Y , если все элементы X принадлежат Y и X не равно Y , т.е. некоторые элементы Y не принадлежат X , это означает, что X является собственным подмножеством Y .

Аналогично можно определить отношения включает $X \supseteq Y$ и строго включает $X \supset Y$.

Нетрудно заметить, что выше введенные отношения $=$, $\subseteq$ и $\subset$ являются подмножествами прямого произведения $A \times B$.

Замечание 5.1. Пустое множество $\emptyset$ является собственным подмножеством любого конечного множества.

Примеры 5.1:

- 1) если $X = \{a, b, c\}$, $Y = \{b, a, c\}$, то $A = B$;
- 2) если $X = \{1, 2, 3, 4\}$, $Y = \{3, 1, 4, 2\}$, то $A \subseteq B$;
- 3) если $X = \{1, 2, 3\}$, $Y = \{3, 1, 4, 2\}$, то $A \subset B$

Определение 5.2. Пусть $A_1, A_2, \dots, A_n$ – произвольные множества, не обязательно различные, $n \geq 1$. Тогда n -арным отношением на множествах $A_1, A_2, \dots, A_n$ является подмножество

$$R^n \subseteq A_1 \times A_2 \times \dots \times A_n,$$

где R^1 – унарное отношение на A_1 , R^2 – бинарное отношение на A_1, A_2 , R^3 – трinarное отношение на A_1, A_2, A_3 и т.д.

Всякое унарное отношение на множестве A является характеристическим свойством некоторого его подмножества. Множество всех унарных отношений на A совпадает с множеством всех подмножеств множества A .

Примеры 5.2:

1. Унарное отношение $R_1^1 = \{n: n \in N \& n < 100\}$ определяет множество натуральных чисел, меньших, чем число 100;

2. Унарное отношение $R_2^1 = \{n: \forall k \in N (n = 2 * k)\}$ определяет множество четных натуральных чисел;

3. Унарное отношение $R_3^1 = \{n: \forall k \in Z (n = 2 * k + 1)\}$ определяет множество нечетных целых чисел.

Определения 5.3. Бинарное отношение определяется над парами множества и может представлено одним из трех способов:

1) *префиксная запись* – знак отношения вставляется перед участниками бинарного отношения;

2) *инфиксная запись* – знак отношения вставляется между участниками бинарного отношения;

3) *постфиксная запись* – знак отношения вставляется после участниками бинарного отношения.

Бинарные отношения над парами элементов часто представляют с помощью таблиц: строки соответствуют первым элементам пары, столбцы – вторым элементам пары, а наличие отношения между конкретными элементами строки и столбца отмечается специальным знаком, например, знаком «1» или др.

Примеры 5.3:

1. Если $a \in A$ и $b \in B$ находятся в бинарном отношении R , то это можно записать как:

Rab – префиксная запись;

aRb – инфиксная запись;

abR – постфиксная запись.

2. Если множества $A = \{a_1, \dots, a_r\}$ и $B = \{b_1, \dots, b_s\}$ находятся в бинарном отношении R , то его можно представить с помощью таблицы 5, в которой элементы a_i представляются строками, элементы b_j – столбцами, а отношения $a_i R b_j$ отмечается “1”:

Таблица 5. Двоичное отношение.

R	b_1	b_2	...	b_{s-1}	b_s
a_1	1		...	1	
a_2		1	...		
...	...	...	...	...	...
a_{r-1}	1		...	1	
a_r			...		1

Определения 5.4. Бинарное отношение R на множестве S :

- 1) *рефлексивно*, если для каждого $s \in S$ имеет место sRs ;
- 2) *транзитивно*, если для любых $s, t, u \in S$ из sRt и tRu следует sRu ;
- 3) *симметрично*, если для любых $s, t \in S$ из sRt следует tRs ;
- 4) *антисимметрично*, если из aRb и bRa следует $a=b$.

Примеры 5.4:

1. Отношение $>$ над числами не является рефлексивным;
2. Отношения $=, \geq, >$ над числами являются транзитивными;
3. Отношение $=$ над числами является симметричным.

Определение 5.5. Бинарное отношение R называется отношением эквивалентности, если оно удовлетворяет свойствам рефлексивности, транзитивности и симметричности.

Каждому отношению эквивалентности на множестве S соответствует единственное разбиение данного множества на смежные классы.

Примеры 5.5. Отношение $=$ в любом числовом множестве будет отношением эквивалентности, т.е., для любых $k, m, n \in \mathbb{N}$:

1. Рефлексивность: $n = n$.
2. Транзитивность: из $k < m$ и $m < n$ следует $k < n$.
3. Симметричность: из $k = m$ следует $m = k$.

Определение 5.6. Бинарное отношение R на некотором множестве S , удовлетворяющее свойствам рефлексивности, транзитивности и антисимметричности, называется отношением частичного порядка.

Примеры 5.6. Отношениями частичного порядка является:

1. Отношение $\subseteq$ для подмножеств некоторого множества;
2. Отношение $\supseteq$ для подмножеств некоторого множества;
3. Отношение $=$ на множестве целых чисел;
4. Отношение $\leq$ на множестве натуральных чисел;
5. Отношение $\geq$ на множестве целых чисел.

Замечание 5.2. Бинарное отношение может задаться тройкой множеств $\langle R, A, B \rangle$, где $R \subseteq A \times B$ – график отношения и записываться $(a, b) \in R$ или aRb . Тогда можно определить:

Область определения: $\text{Dom } R = \{x \in A: \exists y \in B (x, y) \in R\}$;

Область значения: $\text{Run } R = \{y \in B: \exists x \in A (x, y) \in R\}$;

Обратное отношение: $R^{-1} = \{(y, x) \in B \times A: (x, y) \in R\}$;

Композиция отношения: $R \subseteq A \times B, S \subseteq B \times C$,

$$R \cdot S = \{(x, z) \in A \times C: \exists y \in B [(xRy) \& (ySz)]\}.$$

Определение 5.9. Бинарное отношение (X, Y, R) называется

1) *инъективным*, если

$$\forall x_1, x_2 \in X \forall y \in Y \quad x_1 R y \wedge x_2 R y \Rightarrow x_1 = x_2;$$

2) *полным слева*, если

$$\forall x \in X \exists y \in Y \quad x R y;$$

3) *сюръективным* (или *полным справа*), если

$$\forall y \in Y \exists x \in X \quad x R y;$$

4) *функциональным*, если

$$\forall x \in X \forall y_1, y_2 \in Y \quad x R y_1 \wedge x R y_2 \Rightarrow y_1 = y_2;$$

5) *биективным*, если оно полно слева и справа, а также инъективно и функционально.

6) *функцией*, если оно полно слева и функционально, иначе говоря, бинарное отношение $f \subseteq X \times Y$ называется *функцией* из X в Y , если $Dom f = X$, $Run f = Y$ и $(x, y) \in f$, $(x, z) \in f \Rightarrow y = z$.

Функция $f: X \rightarrow Y$ называется:

1) *сюръективной*, если для любого $y \in Y$ существует $x \in X$ такой, что $y = f(x)$, т.е., $\forall y \in Y \exists x \in X (y = f(x))$;

2) *инъективной*, для любых $x_1, x_2 \in X$ из того, что $x_1 \neq x_2$ следует $f(x_1) \neq f(x_2)$, т.е., $\forall x_1 \in X \forall x_2 \in X (x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2))$;

3) *биективной*, если она сюръективна и инъективна.

Любую бинарную функцию можно связать с треномным отношением, например, если задана бинарная функция $f(x, y)$, то ее можно связать с треномным отношением $R^3(x, y, z)$ так, чтобы $z = f(x, y)$.

Примеры 5.9. Пусть $x, y, z \in N$ – натуральные числа и задана бинарная функция $f(x, y) = z$, тогда:

2) если x есть 3, y есть 5, z есть 8, а f есть операция сложение $+$, то вместо записи $f(x, y) = z$ можно писать $3 + 5 = 8$ и с ней связать треномное отношение **Add**(3, 5, 8), для которого справедливо $8 = 3 + 5$.

3) если x есть 8, y есть 2, z есть 4, а f есть операция деление «:», то вместо записи $f(x, y) = z$ можно писать $8 : 2 = 4$ и с ней связать треномное отношение **Dev**(8, 2, 4), для которого справедливо $4 = 8 : 2$.

Ясно, что для некоторых $x, y \in N$ результатом выполнения операции деления является не целое число и для них трехместное

отношение $Dev(x, y, z)$ не выполняется. Поэтому для таких случаев необходимо определить дополнительные условия о результатах.

Определение 5.10. Транзитивным замыканием отношения R на множестве A называется пересечение всех транзитивных отношений, содержащих R как подмножество (иначе, минимальное транзитивное отношение, содержащее R как подмножество).

Транзитивное замыкание существует для любого отношения. Для этого отметим, что пересечение любого множества транзитивных отношений транзитивно. Более того, обязательно существует транзитивное отношение, содержащее R как подмножество.

Транзитивное замыкание обладает следующими свойствами:

1) Транзитивное замыкание рефлексивного отношения рефлексивно, т.к. транзитивное отношение содержит исходное отношение;

2) Транзитивное замыкание симметричного отношения симметрично. Действительно, пусть имеется транзитивное отношение aRb , значит существуют $x_1, x_2, \dots, x_n$ такие, что $aRx_1, x_1Rx_2, \dots, x_nRb$. Но из симметричности отношения R следует $bRx_n, x_nRx_{n-1}, \dots, x_1Ra$, следовательно, bRa .

3) Транзитивное замыкание не сохраняет антисимметричность, например, для отношения $\{(a,b), (b,c), (c,a)\}$ на множестве $\{a, b, c\}$.

4) Транзитивное замыкание транзитивного отношения – оно само.

Отношение $R^* = R^+ \cup R^0$, где $R^0 = \{(\varepsilon, \varepsilon) : \varepsilon \in A\}$ иногда называют рефлексивно–транзитивным замыканием, хотя часто под "транзитивным замыканием" подразумевается именно R^* . Обычно различия между этими отношениями не являются значительными.

Пусть $(M, \cdot)$ – множество M с определённой на нём бинарной операцией умножения « $\cdot$ ». Элемент $e \in M$ называется нейтральным относительно операции $\cdot$, если $x \cdot e = e \cdot x = x$, $\forall x \in M$. Иногда различают *нейтральный слева* элемент e_l , для которого $e_l \cdot x = x$, $\forall x \in M$, и *нейтральный справа* элемент e_r , для которого $x \cdot e_r = x$, $\forall x \in M$.

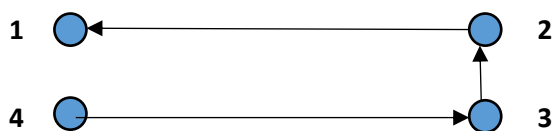
В общем случае может существовать произвольное количество элементов, нейтральных слева или справа. Если одновременно существуют и нейтральный слева элемент e_l , и нейтральный справа элемент e_r , то они обязаны совпадать (так как $e_r = e_l \cdot e_r = e_l$).

Примеры 5.10:

1. Для любых $a, b, c \in \mathbb{N}$ из $a < b$ и $b < c$ следует $a < c$.
2. Для любых $x, y \in \mathbb{N}$ из $x = y$ следует $y = x$.
3. Если A – множество городов, и на них задано отношение xRy , означающее "существует автобусный маршрут из x в y ", то транзитивным замыканием этого отношения будет отношение "существует возможность добраться автобусом из x в y ".

Задания 5.

1. Выпишите упорядоченные пары, принадлежащие следующим бинарным отношениям на множествах $A = \{1, 3, 5, 7\}$ и $B = \{2, 4, 6\}$:
 - (а) $U = \{(x, y): x + y = 9\}$;
 - (б) $V = \{(x, y): x < y\}$.
2. Найдите все упорядоченные пары, принадлежащие отношению, которое определяется на множестве $A = \{1, 2, 3, 4, 5, 6\}$ следующим множеством $R = \{(x, y): x \text{ — делитель } y\}$.
3. Перечислите упорядоченные пары, принадлежащие R , если отношение R будет представлено как на нижнем рисунке.



4. Определите, какие из следующих отношений на множестве людей рефлексивны, симметричны или транзитивны:
 - (а) «...имеет тех же родителей, что и...»;
 - (б) «...является братом...»;
 - (с) «... старше или младше, чем...»;
 - (d) «... не выше, чем...».
 - (d) «... старше, чем сын...».

Вопросы 5:

1. В чем заключается частичная упорядоченность множества?

2. Из чего состоит мощность объединения двух множеств?
3. Как определяется эквивалентность двух множеств?
4. Какие множества являются равномощными?
5. Какие элементы являются несравнимыми?
6. Что такое эквивалентность двух множеств?
7. Что можно сказать о свойствах (рефлексивности, симметричности, и транзитивности) следующих отношений:
 - (а) « x делит y » на множестве натуральных чисел;
 - (б) « $x \neq y$ » на множестве целых чисел;
 - (в) «количество лет x совпадает с возрастом y » на множестве всех людей.

Тесты 5.

1. На одном лекционном потоке количество студентов, изучающих немецкий, французский и английский языки, таково: английский язык изучают 50 человек, французский – 30, немецкий – 20; французский и английский – 10, немецкий и французский – 8, немецкий и английский – 13 и 3 человека изучают все три языка.

Определить:

- 1) сколько студентов на потоке изучают только английский язык?
- 2) сколько студентов на потоке изучают только французский язык?
- 3) сколько студентов на потоке изучают только немецкий язык?
- 4) сколько студентов на потоке изучают ровно 2 языка?
- 5) сколько студентов на потоке не изучают немецкий язык?

2. В цехе предприятия работают 15 человек. Из них: 6 человек имеют

дипломы наладчиков станков с ЧПУ, 8 имеют дипломы слесарей, 5 – фрезеровщиков, 3 человека имеют одновременно дипломы наладчиков станков с ЧПУ и слесарей, 2 человека имеют дипломы наладчика станков с ЧПУ и фрезеровщика, 4 человека имеют дипломы слесаря и фрезеровщика, 1 человек имеет все три вида дипломов. Определить:

1) сколько работников цеха не имеют ни одного вида из этих трех дипломов?

2) сколько работников цеха имеют ровно по два диплома?

3) сколько работников цеха имеют только один из дипломов – наладчика станков с ЧПУ?

4) сколько работников цеха имеют только один из дипломов – слесаря?

5) сколько работников цеха имеют только один из дипломов – фрезеровщика?

3. Студенты 1 курса (60 человек) получают новые книги в библиотеке.

28 человек взяли по учебнику физики, 20 человек – информатики, 23 человека – математики; при этом у 4 студентов оказались учебники по математике и информатике, у 6 – учебники по математике и физике, у 5 – учебники по физике и информатике, а у 3 человек – учебники по всем 3 дисциплинам. Определить:

1) сколько студентов получили только учебник по информатике?

2) сколько студентов получили только учебник по физике?

3) сколько студентов получили только учебник по математике?

4) сколько студентов получили ровно 2 учебника?

5) сколько студентов остались без учебников?

6. ГРАФЫ

Цель: рассмотреть понятие графа, виды графов и их свойств.

План: изучить формальные определения и способов представления графов, разобрать различные виды графов и способы применения графов для решения различных задач.

6.1. Определения и способы представления графов

Граф это динамическая сетевая связная структура данных, представленная множеством пар, называемых *вершинами* и *ребрами*. Каждая вершина может быть связана с несколькими другими вершинами или с самой собой при помощи ребер и вершины, не образующую иерархию. Формально граф определяется как множество пар $G = (X, A)$, где X – множество вершин, A – множество ребер, фактически есть отношение на множестве X , т.е. $A \subseteq X \times X$. Если $x_i \in X$ и $x_j \in X$ – вершины, то (x_i, x_j) – ребра.

Имеются несколько видов графа. Если из каждой вершины графа исходит равное количество ребер и в каждую вершину заходят равное количество ребер, то такой граф называется *регулярным графом*. Если для каждого ребра графа определено направление, то такой граф называется *ориентированным графом*. Если каждое ребро графа имеет вес, то такой граф называется *взвешанным графом*, т.е. можно определить функцию $w: E \rightarrow R$, где R – множество действительных чисел, w – вес ребра и $w \geq 0$.

Матрица смежности это один из способов представления графа в виде матрицы.

Матрица смежности графа G с конечным числом вершин n (пронумерованных числами от 1 до n) – это квадратная матрица A размера n , в которой значение элемента a_{ij} равно числу ребер из i -й вершины графа в j -ю вершину.

Иногда, особенно в случае неориентированного графа, петля (ребро из i -й вершины в саму себя) считается за два ребра, то есть

значение диагонального элемента a_{ij} в этом случае равно удвоенному числу петель вокруг i -й вершины.

Матрица смежности простого графа (не содержащего петель и кратных ребер) является бинарной матрицей и содержит нули на главной диагонали.

В теории графов применяются:

– *Матрица инцидентностей*. Это матрица A с n строками, соответствующими вершинам, и m столбцами, соответствующими рёбрам. Для ориентированного графа столбец, соответствующий дуге (x,y) содержит -1 в строке, соответствующей вершине x и 1 , в строке, соответствующей вершине y . Во всех остальных 0 . Петлю, т.е. дугу (x,x) можно представлять иным значением в строке x , например, 2 . Если граф неориентированный, то столбец, соответствующий ребру (x,y) содержит 1 , соответствующие x и y и нули во всех остальных строках.

– *Матрица смежности*. Это матрица $n \times n$ где n – число вершин, где $a_{ij} = 1$, если существует ребро, идущее из вершины x в вершину y и $a_{ij} = 0$ в противном случае, т.е.:

a_{ij} – число рёбер, связывающих вершины v_i и v_j , причем в А) некоторых приложениях каждая петля (ребро $\{v_i, v_i\}$ при некотором i) учитывается дважды;

Б) матрица смежности пустого графа, не содержащего ни одного ребра, состоит из одних нулей.

На рисунке 6.2 показаны матрицы инцидентности и матрицы смежности для непрерывного графа, показанного на рисунке 6.1

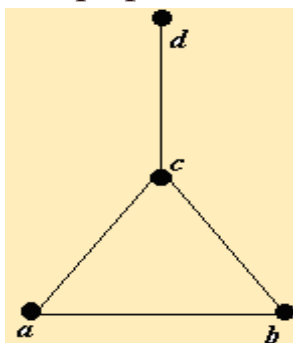


Рисунок 6.1. Пример графа

	u	v	w	x
a	1	1	0	0
b	1	0	1	0
c	0	1	1	1
d	0	0	0	1

Матрица инцидентности

	a	b	c	d
a	0	1	1	0
b	1	0	1	0
c	1	1	0	1
d	0	0	1	0

Матрица смежности

Рисунок 6.2. Матрицы представления графа

Пусть дан граф $G=(X,A)$, где $X=\{x_i\}$, $i = 1, 2, \dots, n$ – множество вершин, $A=\{a_j\}$, $j = 1, 2, \dots, m$ – множество дуг.

Подграфом $G'=(X',A')$ исходного графа G называется такой граф G' , для которого $X' \subseteq X$ и $A' \subseteq A$. Примеры подграфов показаны на рис. 6.2,б, а исходный граф – на рис. 6.2,а.

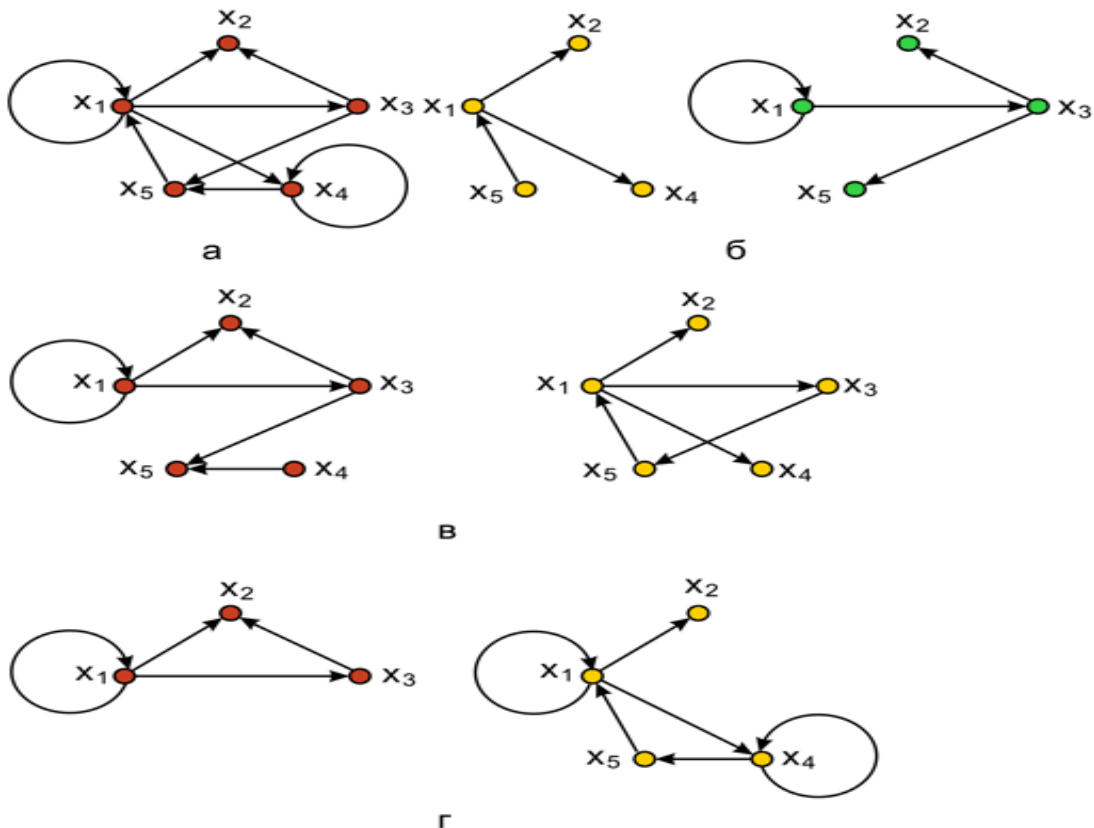


Рисунок 6.2. Виды подграфов: а – исходный граф; б – подграфы; в – остовные подграфы; г – порожденные подграфы

Если A – матрица смежности графа G , то матрица A^n обладает следующим свойством: элемент в i -й строке, j -м столбце равен числу путей из i -й вершины в j -ю, состоящих из ровно n ребер.

Путь в графе это последовательность ребер, ведущая от одной вершины к другой вершине, такая, что каждые два соседних ребра имеют общую вершину, и никакое ребро не встречается более одного раза, т.е. формально путь в графе есть такая последовательность вершин $(x_1, x_2, x_3, \dots, x_{m-1}, x_m)$, что пары $\{(x_1, x_2), (x_2, x_3), \dots, (x_{m-1}, x_m)\}$ станут ребрами. Две вершины $x_i \in X$ и $x_j \in X$ в графе называются *связными (несвязными)*, если в нем существует (не существует) путь, ведущий из x_i в x_j . Этот путь может быть в обоих направлениях. Если каждые две вершины в графе связны, то такой граф называется *связным графом*. Если же в графе найдется хотя бы одна пара несвязных вершин, то граф называется *несвязным*. Если все пары вершин графа связны в обоих направлениях, то такой граф будет *сильносвязным графом*.

Путь без повторяющихся ребер называется *цепью*, а цепь без повторяющихся вершин называется *простой*.

Цепь, в которой совпадают концевые вершины, называется *циклом*, а цикл в котором нет повторяющихся вершин, кроме концевых, называется *простым*, т.е. путь обратно входит в эту же вершину, то такой путь называется *замыканием (циклом)*, т.е. в замыкании начальная и конечная вершины совпадают. Если замыкание не проходит ни через одну из вершин графа более одного раза, то оно называется *простым замыканием*. Если замыкание исходит из одной вершины и напрямую входит в эту вершину обратно, оно называется *петлей*, т.е. в петле есть единственная вершина.

Длиной пути называется число ребер этого пути. Если веса ребер являются их длинами, то длина пути вычисляется так:

$$w(x_1, x_2, x_3, \dots, x_{m-1}, x_m) = \sum_{i=1}^{m-1} w(x_i, x_{i+1}).$$

В графах можно решать следующие задачи: *сравнение двух графов, нахождение наикратчайшей пути из одной вершины в другую, нахождение числа замкнутых путей* и др.

6.2. Деревья

Дерево это граф, в котором все вершины связаны, а пути незамкнуты, т.е. связный граф без циклов и без петель.

В дереве вершины подразделяются на следующие виды:

1) *корень* – вершина, из которой исходит одно или несколько ребер, но не входит ни одного ребра, т.е. вершина, которая не имеет ни одного предка, но может иметь много потомков;

2) *ветвь* – вершина, в которую входит одно ребро, но от него может исходить много ребер, т.е. вершина, которая имеет единственного предка и может иметь много потомков;

3) *лист* – вершина, в которую входит только одно ребро, но не исходит ни одного ребра, т.е. вершина, которая имеет единственного предка, но не имеет ни одного потомка.

В дереве направление пути проходит от корня через ветви до листьев. Внутри дерева могут быть несколько деревьев, которых будем называть *поддеревьями*.

Теперь можно дать следующее рекурсивное определение (со ссылкой на самого себя):

1. *Рекурсивный базис*: множество $\{v\}$, состоящее только из одной вершины v является деревом, где его единственная вершина есть одновременно корень и лист.

2. *Рекурсивный шаг*: если v – вершина и $A_1, A_2, \dots, A_n$ – деревья, то можно построить новое дерево, в котором корнем является вершина v , а ребрами – исходящие из этой вершины и входящие в корни деревьев $A_1, A_2, \dots, A_n$.

3. *Рекурсивное заключение*: Деревья получаются только правилами 1 и 2.

Это определение дерева можно представить на рисунке 6.3 следующим образом:

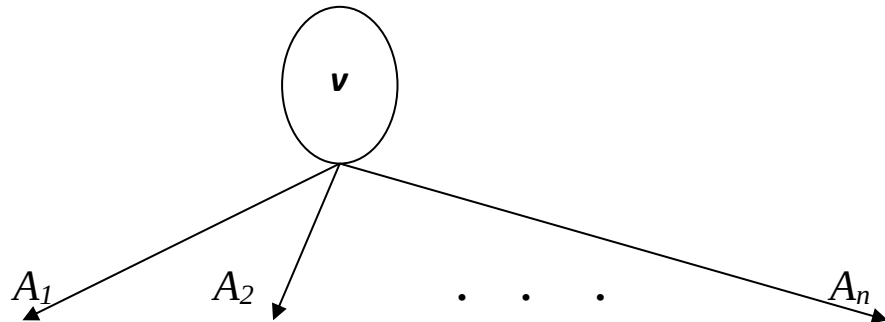


Рисунок 6.3. Определение дерева

Из этого определения явно видно, что дерево является иерархической связной динамической структурой данных, представленной с единственной корневой вершиной и её потомками. Максимальное количество потомков каждой вершины и определяет размерность дерева.

Среди деревьев особо выделяется, так называемые *двоичные (бинарные) деревья*. Его можно определить следующим образом:

Двоичное дерево – дерево, в котором каждая вершина имеет не более двух потомков. Эта вершина называется *родительской вершиной*, а потомки называются *левым наследником* и *правым наследником*. Дадим рекурсивное определение бинарного дерева. Бинарным деревом называется следующее множество вершин:

- либо ничего не содержит (пустое множество);
- либо состоит из корня, который соединяется с двумя бинарными деревьями, называемыми левостороннее поддерево и правостороннее поддерево.

Итак, бинарное дерево является либо пустым, либо состоит из данных и двух поддеревьев, каждое из которых может быть пустым. Если в некоторой вершине оба поддерева пустые, то она есть лист. Формально бинарное дерево определяется так:

$\langle \text{биндеревое} \rangle ::= \text{nil} \mid (\langle \text{данные} \rangle \langle \text{биндеревое} \rangle \langle \text{биндеревое} \rangle)$,
где nil – пусто.

В деревьях решаются следующие задачи: *обход деревьев, поиск дерева, добавление новой вершины к дереву, уничтожение вершины дерева, сравнение деревьев* и др .

Двоичные деревья используются в алгоритмах поиска: каждая вершина двоичного дерева поиска соответствует элементу из некоторого отсортированного набора, все его левые потомки – меньшим элементам, а все его правые потомки – большим элементам. Каждая вершина в дереве однозначно идентифицируется последовательностью неповторяющихся вершин

от корня и до неё – путем. Длина пути и является уровнем вершины в иерархии дерева. Для практических целей обычно используют два подвида бинарных деревьев: *двоичное дерево поиска* – *binary search tree (BST)* и *двоичная куча*.

Двоичное дерево поиска обладает следующими свойствами:

- левое поддерев и правое поддерев являются двоичными деревьями поиска;
- у всех вершин левого поддерева произвольной вершины v значения ключей данных меньше, чем значение ключа данных самой вершины v ;
- у всех вершин правого поддерева той же вершины v значения ключей данных больше, чем значение ключа данных вершины v .

Очевидно, что данные в каждой вершине должны обладать ключами, на которых определена операция сравнения.

Двоичная куча или сортирующее дерево обладает следующими свойствами:

- значение в любой вершине не меньше, чем значения в вершинах её потомков;
- глубина листьев (расстояние до корня) отличается не более чем на один слой;
- последний слой заполняется слева направо.

Такая куча называется *max-heap*. Существуют также кучи, где значение в любой вершине, наоборот, не больше, чем значения её потомков. Такие кучи называются *min-heap*.

Примеры 6.2:

1. Бинарное отношение над конечными объектами может быть представлено в виде ориентированного графа как на рисунке 6.4. Ниже показано отношение делимости над целыми числами от 1 до 12: 2 и 3 делится на 1; 4 и 6 делится на 2; 6 делится на 2 и 3; 12 делится на 4 и 6.

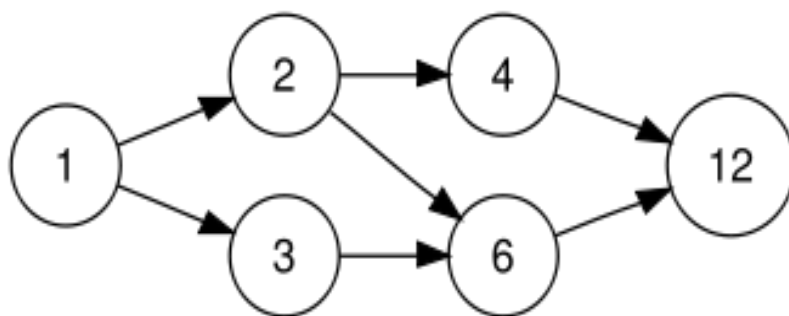


Рисунок 6.4. Представление бинарного отношения

2. Представление бинарного дерева показано на рисунке 6.5.

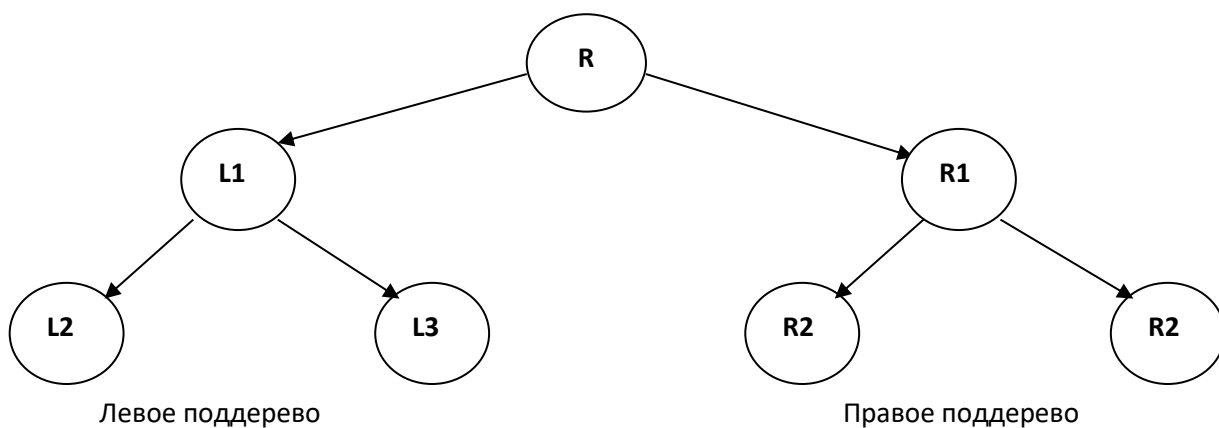


Рисунок 6.5. Бинарное дерево.

3. Обход бинарного дерева арифметического выражения $((3+1)*3/(9-5)+2-(3*(7-4)+6))$ свеху вниз и слева направо представлен на рисунке 6.6.

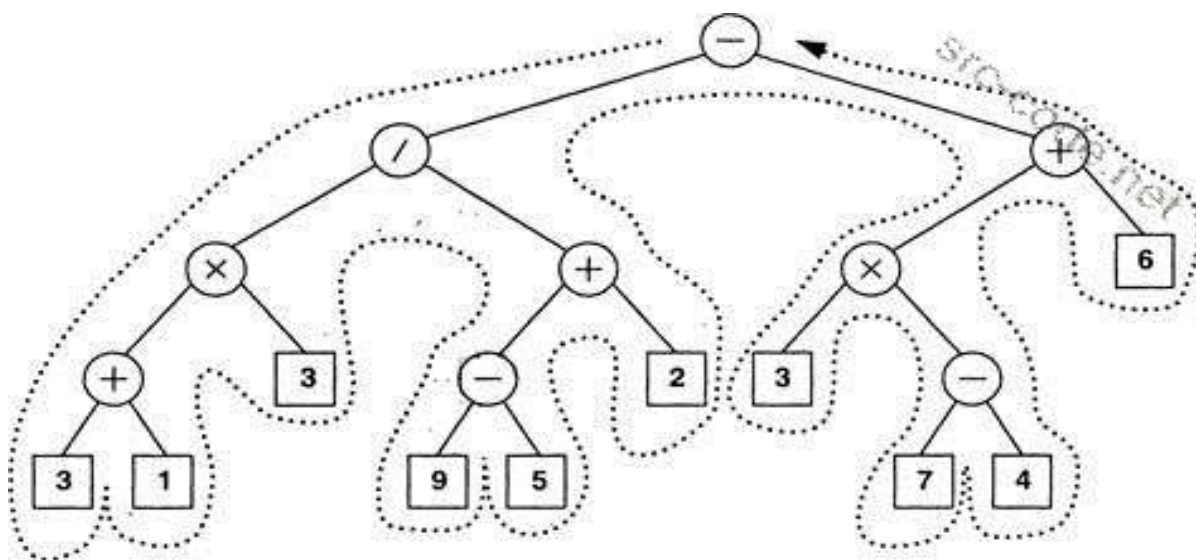


Рисунок 6.6. Обход дерева

Задания 6.1:

1. Постройте ориентированный взвешанный граф для описания структуры идентификатора.
2. Постройте дерево для выражения $((a / (b + c)) + (x * (y - z)))$.
3. Определите матрицу смежности A для неориентированного графа, который содержит петлю вокруг вершины 1, в которой зависимости от конкретных приложений элемент a_{11} может считаться равным либо одному (как показано ниже), либо двум.

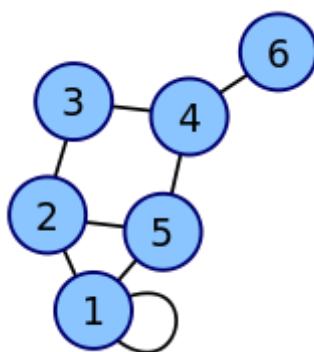


Рисунок 6.1. Неориентированный граф.

Помощь:

1. Не умоляя общности, для облегчения построения требуемого графа будем рассматривать не буквы, а только одну букву и не цифры, только одну цифру, которые будут служить весами нужного взвешанного графа.

2. В соответствующем бинарном дереве листьями служат операнды, а остальными вершинами – операции.

3. Матрица смежности

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

Задания 6.2:

На конечном множестве $N = \{1, 2, 3, 4, 5\}$ задано бинарное отношение

$$R = \{(1,2), (1,4), (1,5), (2,3), (3,2), (3,4), (4,4), (4,5), (5,3), (5,4)\}.$$

Для данного отношения запишите область определения и область значений. Нарисуйте граф этого отношения. Составьте для него матрицу смежности и инцидентности.

Вопросы 6:

1. Как образуется путь в графе?
2. Какие ребра называются кратными?
3. Какая вершина называется изолированной?
4. Чему равна степень изолированной вершины?
5. Что означает степень вершины?
6. Какой граф называется циклическим?
7. Что представляет собой матрица инцидентности?

Тесты 6:

1. На какие виды подразделяются графы?
А) ориентированный граф, неориентированный граф;
В) ориентированный граф, определенный граф;
С) определенный граф, неориентированный граф;
D) определенный граф, неопределенный граф;
Е) неопределенный граф, неориентированный граф.

2. Что такое дерево?

- А) граф без петель и без циклов;
- В) граф без весов;
- С) граф без сетей и циклов;
- Д) граф взвешанный;
- Е) граф ориентированный.

3. Что такое бинарное дерево?

- А) дерево, в котором каждая вершина имеет не более двух потомков;
- В) дерево, в котором имеются две вершины;
- С) дерево, в котором нет цикла;
- Д) дерево, в котором нет петли;
- Е) дерево, в котором одна вершина не имеет прямых потомков.

8. ЯЗЫКИ И ГРАММАТИКИ

Цель: дать основные понятия языка и грамматики, определить операции над языками и правила обработки текста.

План: определить элементы (алфавит, символ, слово и язык) грамматики и операции над ними, рассмотреть примеры, ввести понятие выводимости цепочки.

7.1. Формальное определение языка

1. *Алфавитом* является конечное непустое множество элементов, называемых символами (буквами). В дальнейшем, алфавит будет обозначаться прописными (заглавными) латинскими буквами, а элементы – строчными (маленькими).

2. *Словом (цепочкой)* в заданном алфавите называется конечная последовательность элементов этого алфавита. В дальнейшем, слова будут обозначаться строчными греческими буквами.

3. Слово, не содержащее ни одного символа называется *пустым словом*, и обозначается греческой буквой ε .

4. *Длина слова* ω задается числом его символов и обозначается $|\omega|$. Причем каждый символ считается столько раз, сколько он встречается в ω . Длина пустого слова равна нулю, т.е. $|\varepsilon|=0$.

5. Если α и β есть слова в алфавите T , то слово $\alpha\beta$ называется *конкатенацией (сцеплением)* слов α и β , которая получается в результате приписывания слова β в конец слова α . Обычно в записи знак $\cdot$ опускают и пишут просто $\alpha\beta$.

6. Если α – слово и $n \geq 0$ – целые положительное число, то через $\alpha^0 \Rightarrow \varepsilon$ и $\alpha^n \Rightarrow \underbrace{\alpha \cdot \alpha \cdot \dots \cdot \alpha}_n$ (знак $\Rightarrow$ читается “равно по определению”).

7. Если τ и ω есть слова в алфавите T и для некоторого слова ξ в T выполняется $\omega = \tau\xi$, то слово τ называется *префиксом (началом)* слова ω и обозначается $\tau \sqsubseteq \omega$, т.е. $\tau \sqsubseteq \omega \Leftrightarrow \exists \xi (\omega = \tau\xi)$.

8. Если τ и ω есть слова в алфавите T и для некоторого слова ζ в T имеет место $\omega = \zeta\tau$, то слово τ называется *постфиксом (концом)* слова ω и обозначается $\tau \sqsupseteq \omega$, т.е. $\tau \sqsupseteq \omega \Leftrightarrow \exists \zeta (\omega = \zeta\tau)$.

9. Если τ и ω есть слова в алфавите T и для некоторых слов ζ и ξ выполняется $\omega = \zeta\tau\xi$, то слово τ называется *подсловом (подцепочкой)* слова ω и обозначается $\tau \subseteq \omega$, т.е. $\tau \subseteq \omega \Leftrightarrow \exists \zeta \exists \xi (\omega = \zeta\tau\xi)$.

10. Если τ и ω есть слова в алфавите T и τ является подсловом слова ω , то через $|\omega|_\tau$ обозначается количество вхождений слово τ в слове ω .

Замечание 7.1.

Для любого непустого слово ω в алфавите T и пустого слово ε выполняются отношения $\varepsilon \sqsubseteq \omega$ и $\varepsilon \sqsupseteq \omega$.

Примеры 7.1: Пусть задан алфавит $T = \{a, b, c, d\}$. Тогда:

1. $a, bb, ccc, abcd, dcba$ и другие последовательности, состоящие из букв a, b, c, d являются словами в T .

2. $|a|=1, |bb|=2, |ccc|=3, |abcd|=4$.

3. Если $\alpha = ccc$ и $\beta = dddd$, то $\alpha\beta = cccdddd$.

4. $ab^2 = abb, (ab)^3 = ababab$.

5. Если $\alpha = ab, \beta = abcd, \xi = cd$, то $\beta = \alpha\xi$, т.е. $\alpha \sqsubseteq \beta$.

6. Если $\alpha = cd, \beta = abcd, \zeta = ab$, то $\beta = \zeta\alpha$, т.е. $\alpha \sqsupseteq \beta$.

7. $\varepsilon \sqsubseteq abcd, a \sqsubseteq abcd, ab \sqsubseteq abcd, abc \sqsubseteq abcd, abcd \sqsubseteq abcd$.

8. $abcd \sqsupseteq \varepsilon, abcd \sqsupseteq d, abcd \sqsupseteq cd, abcd \sqsupseteq bcd, abcd \sqsupseteq abcd$.

9. Если $\tau = ca$ и $\xi = da$, то $\tau \sqsubseteq \omega$ и $\omega = cada$.

10. Если $\tau = 12$ и $\zeta = 11$, то $\tau \sqsupseteq \omega$ и $\omega = 1112$.

11. Если $\tau = bbb, \zeta = aaa$ и $\xi = ccc$, то $\tau \subseteq \omega$ и $\omega = aaabbbccc$.

12. Если $\omega = abcdabcd$ и $\tau = cd$, то $\tau \subseteq \omega$ и $|\omega|_\tau = 2$.

Если T – алфавит, то множество всех цепочек (слов) конечной длины в этом алфавите определяется как

$$T^* \Leftrightarrow \bigcup_{k=0}^{\infty} T^k,$$

где $T^0 \Leftrightarrow \{\varepsilon\}$ – множество цепочек длины 0, $T^k \Leftrightarrow T \cdot T \cdot \dots \cdot T$ –
 $\underset{k}{\cdot}$

множество цепочек длины $k, k \geq 1$ – натуральные числа.

Через T^+ обозначается множество всех возможных непустых цепочек, т.е. $T^+ = T^* \setminus \{\varepsilon\}$. Например, если $T = \{a\}$, то множество T^+ определяется как $T^+ = \{a, aa, aaa, \dots\}$.

Очевидно, что не все цепочки из множества T^+ могут быть осмысленными единицами (словами, словосочетаниями, фразами, предложения и текстами) некоторого языка. Заметим, что осмысленными единицами языка могут быть только те цепочки, которые удовлетворяют грамматическим правилам и имеют семантические значения. Поэтому любой конкретный язык L является собственным подмножеством множества T^+ , т.е. $L \subset T^+$. Например, если в качестве элементов алфавита T взять казахских букв и различные специальные знаки, то казахский язык будет собственным подмножеством множества T^+ , в котором содержатся только слова, словосочетания, фразы, предложения и тексты, имеющие смысл в казахском языке.

Поскольку каждый язык является множеством цепочек конечной длины в заданном алфавите, то можно рассматривать заданных над одним и тем же алфавитом операции *объединения, пересечения, разности, дополнения, прямого произведения, симметрической разности, конкатенации и итерации* языков.

Определения 7.2. Пусть заданы два языка L_1 и L_2 в алфавите T , т.е., $L_1 \subseteq T^+$ и $L_2 \subseteq T^+$, а также U – универсум, тогда можно определить:

1. Объединение: $L_1 \cup L_2 \hat{=} \{x: x \in L_1 \vee x \in L_2\}$;
2. Пересечение: $L_1 \cap L_2 \hat{=} \{x: x \in L_1 \ \& \ x \in L_2\}$;
3. Разность: $L_1 \setminus L_2 \hat{=} \{x: x \in L_1 \ \& \ x \notin L_2\}$;
4. Дополнение: $\overline{L} \hat{=} \{x: x \in U \ \& \ x \notin L\}$;
5. Прямое произведение: $L_1 \times L_2 \hat{=} \{(a, b): a \in L_1 \ \& \ b \in L_2\}$;
6. Симметрическая разность: $L_1 \Delta L_2 \hat{=} \{x: x \in (L_1 \setminus L_2) \vee x \in (L_2 \setminus L_1)\}$;
7. Конкатенация: $L_1 \cdot L_2 \hat{=} \{a \cdot b: a \in L_1 \ \& \ b \in L_2\}$;
8. Итерация – звездочка Клини (Kleene star): $L^* \hat{=} \bigcup_{k=0}^{\infty} L^k$, где

$$L^0 \Rightarrow \{\varepsilon\}, L^k \Rightarrow \underbrace{L \cdot L \cdot \dots \cdot L}_k, k \geq 1.$$

Примеры 7.2: Пусть заданы языки $L_1 = \{aa, bb\}$, $L_2 = \{cc, dd\}$

1. Если $L_1 = \{aa, bb\}$, $L_2 = \{cc, dd\}$, то

$$L_1 \cdot L_2 = \{aacc, aadd, bbcc, bbdd\}.$$

2. Если $L = \{a^k b a^L : 0 < k < l\}$, то

$$L^2 = \{a^k b a^L b a^m : 0 < k < L - 1, m > 1\}.$$

3. Если $T = \{a, b\}$, $L = \{aa, ab, ba, bb\}$, то

$$L^* = \{\tau \in T^* : |\tau| = 2\}, L^* = \{\tau \in T^* : |\tau|_a = 1 \ \& \ |\tau|_b = 1\}.$$

Замечания 7.2:

1. Множество слов конечной длины в алфавите T является частично упорядоченным множеством с отношением $\leq (\geq)$ и все его цепочки сравнимы по длине относительно $\leq (\geq)$.

2. Множество слов конечной длины в алфавите T является частично упорядоченным множеством с отношением $\subseteq (\supseteq)$ и все подмножества слов в алфавите T сравнимы относительно $\subseteq (\supseteq)$.

Определения 7.3. Пусть $L \subseteq T^*$, тогда можно ввести следующие обозначения и понятия:

1. L^R – операция *обращение* над языком L определяется как $L^R = \{\tau^R : \tau \in L\}$.

2. $\text{Pref}(L)$ – *множество префиксов* языка L определяется как $\text{Pref}(L) \Rightarrow \{\tau : \exists \xi (\xi \in L \ \& \ \tau \sqsubseteq \xi)\}$, где $\sqsubseteq$ – отношение префикса.

3. $\text{Suf}(L)$ – *множество постфиксов* языка L определяется как $\text{Suf}(L) \Rightarrow \{\tau : \exists \zeta (\zeta \in L \ \& \ \tau \supseteq \zeta)\}$, где $\supseteq$ – отношение постфикса.

4. $\text{Substr}(L)$ – *множество всех подслов (подцепочек)* языка L определяется как $\text{Substr}(L) \Rightarrow \{\tau : \forall \xi (\xi \in L \ \& \ \xi \neq \varepsilon \Rightarrow \tau \subseteq \xi)\}$.

5. Функция $f: K \rightarrow L$ называется *биекцией*, если каждый элемент множества L является образом ровно одного элемента множества K относительно функции f .

3. Множества K и L называются *равномощными*, если существует биекция из K в L .

Примеры 7.3. Пусть задан язык:

1. $L = \{a^m b a^n : m \leq n\}$ является языком в алфавите $\{a, b\}$, тогда множество всех его подцепочек содержит цепочки $b, ba, aba, baa, abaa, baaa, aabaa, abaaa$ и т. д.

2. $L = \{ab^n : n \geq 0\}$ является языком в алфавите $\{a, b\}$, тогда множество всех его подцепочек содержит цепочки $a, ab, abb, abbb, abbbb, abbbbb$ и т. д.

3. $L = \{abc, a\}$ является языком в алфавите $\{a, b, c\}$, тогда множество всех подслов этого языка L будет равно

$$\text{Substr}(L) = \{\varepsilon, a, b, c, ab, ac, bc, abc\}.$$

Задания 7.4:

1. Пусть заданы $T = \{a, b\}$ и $L = \{aa, ab\}$. Найти L^3 .

2. Пусть задан алфавит $T = \{a, b, c, d\}$ и язык $L = \{\tau \in T^* : |\tau|_a = 1 \ \& \ |\tau|_b = 1\}$. Верно ли, что $abcdcacdcabbacba \in L^*$.

3. Существует ли такой язык L , что выполняется неравенство $L^* \neq \{x^n : x \in L, n \geq 0\}$.

4. Существует ли такой язык L , что выполняется неравенство $(L^R)^* \neq (L^*)^R$?

5. Перечислить слова языка $L_1 \cap L_2$, где $L_1 = \{(ab)^n : n \geq 0\}$ и $L_2 = \{a^m b^m : m \geq 0\}$.

6. Языков $L_1 = \{(abc)^n a : n \geq 2\}$ и $L_2 = \{ab(cab)^n ca : n \geq 1\}$.

7. Пусть $T = \{a, b, c\}$. $L_1 = \{\tau \in T : |\tau| = 4\}$ и $L_2 = \{\tau \in T^* : |\tau|_c = 1\}$. Вычислить число цепочек языка $L_1 \setminus L_2$.

8. Пусть заданы языки L_1 и L_2 . Выяснить равномощность языков $L_1 \cdot (L_2^R)$ и $L_1^R \cdot L_2$.

9. Для языков L_1 и L_2 определите результат их *конкатенации и объединения*:

$L_1 = \{d, de, dee\}$ и $L_2 = \{\varepsilon, d, e, de, d\}$;

$L_1 = \{\varepsilon, d, de, dee\}$ и $L_2 = \{d, e, dee, d\}$;

$L_1 = \{\varepsilon, d, e, de, ded\}$ и $L_2 = \{\varepsilon, d, e, de, ed\}$;

$L_1 = \{d, e, dd, de, ee, ded\}$ и $L_2 = \{d, e, dd, de, ed\}$.

$L_1 = \{d, e, ded, ddde, eedd\}$ и $L_2 = \{d, e, ddd, ded, eee\}$.

$L_1 = \{\varepsilon, d, e, ddd, eee\}$ и $L_2 = \{ddee, eedd\}$.

10. Пусть задан язык $L = \{dcc, dcd, ddc, ddd\}$. Нужно определить какой из следующих языков получается из итерации L^* ?

$\{\omega: \omega = d^i \text{ и } |\omega| \text{ делится на } 3\} \cup \{\varepsilon\};$

$\{\omega: \omega = d^i \text{ и } |\omega| \geq 3\} \cup \{\varepsilon\};$

$\{\omega: \omega = \omega_1 \omega_2 \omega_3 \dots \omega_{3n} \text{ и } \omega_{3i+1} = d \text{ для всех } i < n\} \cup \{\varepsilon\};$

$\{\omega: \omega = d^i \text{ и } |\omega| \geq 12\}.$

$\{\omega: \omega = d^i \text{ и } |\omega| \geq 4\}.$

$\{\omega: \omega = d^i \text{ и } |\omega| \text{ делится на } 4\}.$

11. Пусть $L = \{abcd, ad\}$ является языком в алфавите $\{a, b, c, d\}$. Найти множество всех подслов этого языка $\text{Substr}(L)$.

12. Пусть $L = \{a^k b^m c^n: k \leq m \leq n\}$ является языком в алфавите $\{a, b, c\}$. Найти множество всех подслов этого языка $\text{Substr}(L)$.

7.2. Формальные грамматики

Важный класс механизмов порождения языков образуют формальные грамматики (formal grammar), которых в 1959 году ввел впервые американский лингвист Хомский [24].

В формальной грамматике, порождающей язык L , используются два непересекающихся множества символов:

1) конечное множество терминалов (terminals) – постоянных величин T , из которых образуются цепочки языка L ;

2) непересекающееся с множеством T конечное множество нетерминалов (nonterminals) – переменных величин N , которыми обозначают грамматические понятия, категории и т.п. языка L .

3) Процесс порождения цепочек языка L описывается конечным множеством правил подстановки (rewriting rule) P , каждое из которых состоит из пар цепочек (α, β) . В такой паре первая компонента α является цепочкой, содержащей хотя бы один нетерминал, а вторая компонента может быть любой цепочкой, образованной из терминальных и/или нетерминальных символов. Она может быть и пустой цепочкой.

Соглашения 7.1. Принимаются следующие соглашения :

(1) строчные латинские курсивные буквы $a, b, \dots, z$ и арабские цифры $0, 1, \dots, 9$ обозначают терминалы;

(2) прописные латинские курсивные буквы $A, B, \dots, X, Y, Z$ обозначают нетерминалы, при этом через S обозначается начальный нетерминальный символ;

(3) строчные греческие буквы $\alpha, \beta, \dots, \omega$ обозначают цепочки, которые могут содержать как терминалы, так и нетерминалы, здесь ε – пустая цепочка;

(4) правило подстановки, являющееся парой цепочек (α, β) из множества P , записывается в виде $\alpha \rightarrow \beta$;

4) (6) правила вида $\alpha \rightarrow \varepsilon$ называются ε (эпсилон)–правилами;

5) (7) эти соглашения распространяются также и на буквы с нижними и верхними индексами;

(8) правила вида $\alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_m \rightarrow \beta$ является сокращением m правил вида $\alpha_1 \rightarrow \beta, \alpha_2 \rightarrow \beta, \dots, \alpha_m \rightarrow \beta$ или правил, записанных в столбик следующим образом:

$$\alpha_1 \rightarrow \beta$$

$$\alpha_2 \rightarrow \beta$$

...

$$\alpha_m \rightarrow \beta$$

(9) правила вида $\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$ является сокращением n правил вида $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \dots, \alpha \rightarrow \beta_n$ или:

$$\alpha \rightarrow \beta_1$$

$$\alpha \rightarrow \beta_2$$

...

$$\alpha \rightarrow \beta_n$$

(10) правило вида $\alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_m \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$ является сокращением $m \times n$ правил, полученных из соглашения (6) и (7).

Определение 7.4. Формальной грамматикой называется следующая четверка $G = \langle T, N, P, S \rangle$, где:

T – непустое конечное множество терминальных символов (терминалов);

N – непустое конечное множество нетерминальных символов (нетерминалов), причем $T \cap N = \emptyset$, $\emptyset$ – пустое множество;

P – непустое конечное множество правил подстановки вида $\alpha \rightarrow \beta$, где $\alpha \in (T \cup N)^* \times N \times (T \cup N)^*$, $\beta \in (T \cup N)^*$, то есть,

$$P \subseteq \{(\alpha, \beta): \alpha \in (T \cup N)^* \times N \times (T \cup N)^* \& \beta \in (T \cup N)^*\};$$

S – начальный нетерминал, $S \in N$.

Правила вывода грамматики можно рассматривать как элементарные операции, которые, будучи применены в определенной последовательности к исходной цепочке, порождают лишь правильные цепочки. Сама последовательность правил, использованных в процессе порождения некоторой цепочки, является её выводом.

Язык, определяемый (порождаемый) грамматикой, является множеством конечных цепочек, которые состоят только из терминалов. Все эти терминальные цепочки выводятся, начиная с одной особой цепочки, состоящей только из одного начального нетерминала S .

Примеры 7.4. Пусть грамматика $G = \langle T, N, P, S \rangle$ с параметрами: $T = \{+, -, /, *, (,), a\}$, $N = \{S, E, T, F\}$, $P = \{S \rightarrow E, E \rightarrow E+T \mid E-T \mid T, T \rightarrow T*F \mid T/F \mid F, F \rightarrow a \mid (E)\}$ порождает скобочное алгебраическое выражение в инфиксной записи.

Для определения языка с помощью грамматики используются понятие *выводимой цепочки* и *отношение непосредственной выводимости*.

Определения 7.5:

1. В грамматике $G = \langle T, N, P, S \rangle$ *выводимые цепочки* рекурсивно определяются так:

1) S – выводимая цепочка грамматики G ;

2) Если $\alpha\beta\gamma$ – выводимая цепочка грамматики G и в P имеется правило $\beta \rightarrow \sigma$, то $\alpha\sigma\gamma$ – тоже выводимая цепочка грамматики G .

2. Выводимая цепочка грамматики G , не содержащая нетерминальных символов из N , называется *терминальной цепочкой*, порождаемой грамматикой G .

3. Если $\alpha = \gamma\xi\delta$, $\beta = \gamma\eta\delta$ и $\alpha \rightarrow \beta$, $\xi \rightarrow \eta$ – правила вывода грамматики G , то говорят, что между цепочками α и β установлено *отношение непосредственной выводимости*, которое означает, что в грамматике G цепочка β непосредственно выводится из цепочки α путем замены ξ на η и обозначается это отношение через $\alpha \Rightarrow_G \beta$. Если грамматика заранее известна, то показатель G в отношении непосредственной выводимости опускается и это отношение записывается как $\alpha \Rightarrow \beta$.

Запись вида $\alpha \Rightarrow^k \beta$ является k -той степенью отношения $\alpha \Rightarrow \beta$, если существуют $k+1$ цепочки $\alpha_0, \alpha_1, \dots, \alpha_k$ такие, что $\alpha = \alpha_0$, $\alpha_k = \beta$ и $\alpha_{i-1} \Rightarrow \alpha_i$ ($1 \leq i \leq k$). Эта последовательность цепочек называется *выводом длины k* цепочки β из цепочки α в грамматике G .

Если существует $i \geq 1$ (или $i \geq 0$) выполняется отношение $\alpha \Rightarrow^i \beta$, то это записывается как $\alpha \Rightarrow^+ \beta$ (или $\alpha \Rightarrow^* \beta$). Здесь через $\Rightarrow^+$ обозначается транзитивное замыкание отношения $\Rightarrow$, а через $\Rightarrow^*$ – рефлексивное и транзитивное замыкание отношения $\Rightarrow$. При этом запись вида $\varphi \Rightarrow^+ \psi$ ($\varphi \Rightarrow^* \psi$) читается как: « ψ выводима из φ нетривиальным образом» (« ψ выводима из φ »).

Замечание 7: $\alpha \Rightarrow^* \beta$ тогда и только тогда, когда $\alpha \Rightarrow^i \beta$ для некоторого $i \geq 0$, и $\alpha \Rightarrow^+ \beta$ тогда и только тогда, когда $\alpha \Rightarrow^i \beta$ для некоторого $i \geq 1$.

Определения 7.6:

1. Каждая цепочка, которая выводится из начального нетерминала грамматики, называется *сентенциальной формой*.

2. Выводимые цепочки, не содержащие нетерминальных символов, называются *терминальными цепочками*. Поэтому язык $L(G)$ можно определить как множество терминальных цепочек, выводимых в грамматике G .

3. Языком $L(G)$, порождаемым грамматикой G , называется множество терминальных цепочек, которые выводятся из одного

начального нетерминала S посредством применения правило подстановки из множества P , т.е. формально записывается как

$$L(G) = \{\tau: \tau \in T^*, S \Rightarrow^* \tau\}.$$

Это означает, что любая принадлежащая языку $L(G)$ цепочка является сентенциальной формой.

Примеры 7.5.

1. Пусть задана грамматика $G_1 = \langle T, N, P, S \rangle$, где $T = \{0, 1\}$ – множество терминалов, $N = \{A, S\}$ – множество нетерминалов, $P = \{S \rightarrow 0A1, 0A \rightarrow 00A1, A \rightarrow \varepsilon\}$ – множество правил подстановки. Если рассмотрим вывод вида $S \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 0011$, то можно увидеть, что на первом шаге нетерминал S по правиле $S \rightarrow 0A1$ заменяется на цепочку $0A1$, на втором шаге цепочка $0A$ по правиле $0A \rightarrow 00A1$ заменяется на цепочку $00A11$, а на третьем шаге нетерминал A по правиле $A \rightarrow \varepsilon$ заменяется на пустую цепочку ε . Таким образом, можно говорить, что $S \Rightarrow^3 0011$, $S \Rightarrow^+ 0011$, $S \Rightarrow^* 0011$ и цепочка 0011 принадлежит языку $L(G_1) = \{0^n 1^n: n > 1\}$.

2. Грамматика с правилами $P_1 = \{S \rightarrow 01S, S \rightarrow 0\}$ и грамматика с правилами $P_2 = \{S \rightarrow 0A, A \rightarrow 10A, A \rightarrow \varepsilon\}$ эквивалентны.

3. Две грамматики для порождения алгебраических выражений, образованных операндами i, n и операциями $+, *$ с одинаковыми терминальными символами $T = \{i, n, (,), +, *\}$ и нетерминальными символами $N = \{S, F, H\}$, но с разными правилами:

$$P_1 = \{S \rightarrow S+F, S \rightarrow F, F \rightarrow F*H, F \rightarrow H, F \rightarrow H, H \rightarrow i, H \rightarrow n, H \rightarrow (S)\};$$

$$P_2 = \{S \rightarrow S+F|F, S \rightarrow S+F|S*F|F, F \rightarrow F*H|H, H \rightarrow i|n|(S)\};$$

эквивалентны.

Задания 7.2:

1. Построить все сентенциальные формы для грамматики с правилами:

$$S \rightarrow A+B|B+A, A \rightarrow a, B \rightarrow b.$$

2. Построить вывод заданной цепочки $a-b^*a+b$ для грамматики с правилами:

$$S \rightarrow K|F+S|K-S, K \rightarrow F|F*K, F \rightarrow a|b.$$

3. Построить вывод заданной цепочки $aaabbbccss$ для грамматики с правилами:

$$S \rightarrow aSBC|abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc.$$

4. Описать язык, порождаемый грамматикой

$$S \rightarrow FF, F \rightarrow aFb, F \rightarrow ab.$$

5. Описать язык, порождаемый грамматикой

$$S \rightarrow Sc, S \rightarrow A, A \rightarrow aAb, A \rightarrow \varepsilon.$$

6. Описать язык, порождаемый грамматикой

$$S \rightarrow \varepsilon, S \rightarrow a, S \rightarrow b, S \rightarrow aSa, S \rightarrow bSb.$$

7. Описать язык, порождаемый грамматикой

$$S \rightarrow SA, SAA \rightarrow ASb, ASA \rightarrow b, A \rightarrow a.$$

8. Описать язык, порождаемый грамматикой

$$S \rightarrow aSA, S \rightarrow abc, bA \rightarrow bbc, cA \rightarrow Aa.$$

9. Описать язык, порождаемый грамматикой

$$S \rightarrow aAS, S \rightarrow B, Aa \rightarrow aaA, AB \rightarrow B, B \rightarrow a.$$

Вопросы 7.

1. Что такой префикс цепочки?

2. Что такой постфикс цепочки?

3. Если $\omega = abcdabefabhgabik$ и $\tau = ab$, то чему равна $|\omega|_\tau$?

4. Как определяется множество всех цепочек?

5. Что такой язык в заданном алфавите?

6. Как определяется $\overline{L}$?

7. Как определяется $L_1 \setminus L_2$?

8. Как определяется $L_1 \Delta L_2$?

9. Как определяется $L_1 \cdot L_2$?

10. Как определяется L^+ ?

11. Чему равно L^0 ?

12. Чему равно L^k ?

13. Как строится обращение языка L^R ?

14. Существует ли такой язык L , что выполняется $(L^R)^* \neq (L^*)^R$?
15. Как получается префикс языка $\text{Pref}(L)$?
16. Как получается постфикс языка $\text{Suf}(L)$?
17. Как определяется $\text{Substr}(L)$?
18. Какое множество определяет регулярное выражение $(ab)^+$?
19. Какое множество определяет регулярное выражение $(aa \vee bb)^*$?
20. Какое множество определяет регулярное выражение $a(\varepsilon \vee a) \vee b$?
21. Какое множество определяет регулярное выражение $a(a \vee b)^*$?
22. Какое множество определяет регулярное выражение ε^* ?
23. Какое множество определяет регулярное выражение $((a \vee bc)^* \cdot a)$?
24. Какое множество определяет регулярное выражение $(c \cdot (ab \vee cd)^*)$?

Тесты 7.

1. Как правильно определяется алфавит формальных языков?
 - А) некоторое конечное множество символов;
 - В) любая конечная последовательность символов;
 - С) подмножество цепочек конечной длины;
 - Д) набор токенов, которые могут представлять определенную лексему в исходной программе
 - Е) набор всех операторов сравнения

2. Как правильно определяется цепочки символов в алфавите V ?
 - А) некоторое конечное множество символов;
 - В) любая конечная последовательность символов алфавита V ;
 - С) подмножество цепочек конечной длины алфавита V ;

- Д) набор токенов, которые могут представлять определенную лексему в исходной программе;
- Е) набор всех операторов сравнения.

3. Что обозначается символом $|\alpha|$?

- А) длина строки алфавита;
- В) конечное множество символов;
- С) последовательность символов;
- Д) язык алфавита V ;
- Е) длина цепочки.

4. Пусть задан язык $L = \{dcc, dcd, ddc, ddd\}$. Нужно определить какой из следующих языков получается из итерации L^* ?

- А) $\{\omega: \omega = \varepsilon \vee dcc \vee dcd \vee ddc \vee ddd\}$;
- В) $\{\omega: \omega = d^3 \text{ и } |\omega| \text{ делится на } 3\} \cup \{\varepsilon\}$;
- С) $\{\omega: \omega = d^3 \text{ и } |\omega| \geq 12\}$.
- Д) $\{\omega: \omega = d^4 \text{ и } |\omega| \geq 4\}$.
- Е) $\{\omega: \omega = d^4 \text{ и } |\omega| \text{ делится на } 4\}$.

5. Какое множество определяет регулярное выражение $\varepsilon \vee \phi$?

- А) $\{\varepsilon\} \cup \emptyset$;
- В) $\{\varepsilon\}$;
- С) $\emptyset$;
- Д) $\{\varepsilon\}^*$;
- Е) $\{\varepsilon\} \cup \emptyset^*$.

6. Какое множество определяет регулярное выражение ϕ^* ?

- А) $\{\varepsilon\}$;
- В) $\{\varepsilon\}^*$;
- С) $\emptyset^*$;
- Д) $\{\varepsilon\} \cup \{\varepsilon\}^*$;
- Е) $\{\varepsilon\} \cup \emptyset$.

7. Какие правила вывода являются верными для языка $L(R) = \{(ac)^n (cb)^n \mid n > 0\}$?

- A) $P: S \rightarrow aQb \mid accb, Q \rightarrow cSc$;
- B) $P: S \rightarrow aSb \mid accb, Q \rightarrow cSc$
- C) $P: S \rightarrow aQb \mid accb, Q \rightarrow cQc$;
- D) $P: S \rightarrow aQb \mid accb, Q \rightarrow cc$;
- E) $P: S \rightarrow aQ \mid accb, Q \rightarrow cSc$.

8. Пусть $G = (\{a,b,c\}, \{A,B,S\}, P, S)$, где правила вывода P имеют вид: $S \rightarrow AB, A \rightarrow a, A \rightarrow ac, B \rightarrow b, B \rightarrow cb$. Определите $L(G)$?

- A) $L(G) = \{ab, acb, accb\}$;
- B) $L(G) = \{ab, abc, accb\}$;
- C) $L(G) = \{ab, acb, ac\}$;
- D) $L(G) = \{ab, b, accb\}$;
- E) $L(G) = \{a, acb, accb\}$.

8. АВТОМАТЫ

Цель: дать понятие универсального автомата и конечный автомат

План: изучить состав и структуру абстрактного автомата, дать формальное определение недетерминированного и детерминированного конечного автомата и языка, распознаваемого такими автоматами.

8.1. Абстрактный автомат

Обычно под словом «автомат» понимается устройство, после включения самостоятельно выполняющее ряд заданных операций. Однако мы имеем дело с абстрактным автоматом, применяемым как математическая модель любых цифровых (дискретных) устройств, в которых все сигналы квантованы по уровню, а все действия квантованы по времени.

Абстрактный автомат (далее – автомат) может распознавать некоторое множество или преобразует одно множество в другое и состоит из ленты, головки и управляющего устройства, а также может иметь рабочую (внешнюю и внутреннюю) память.

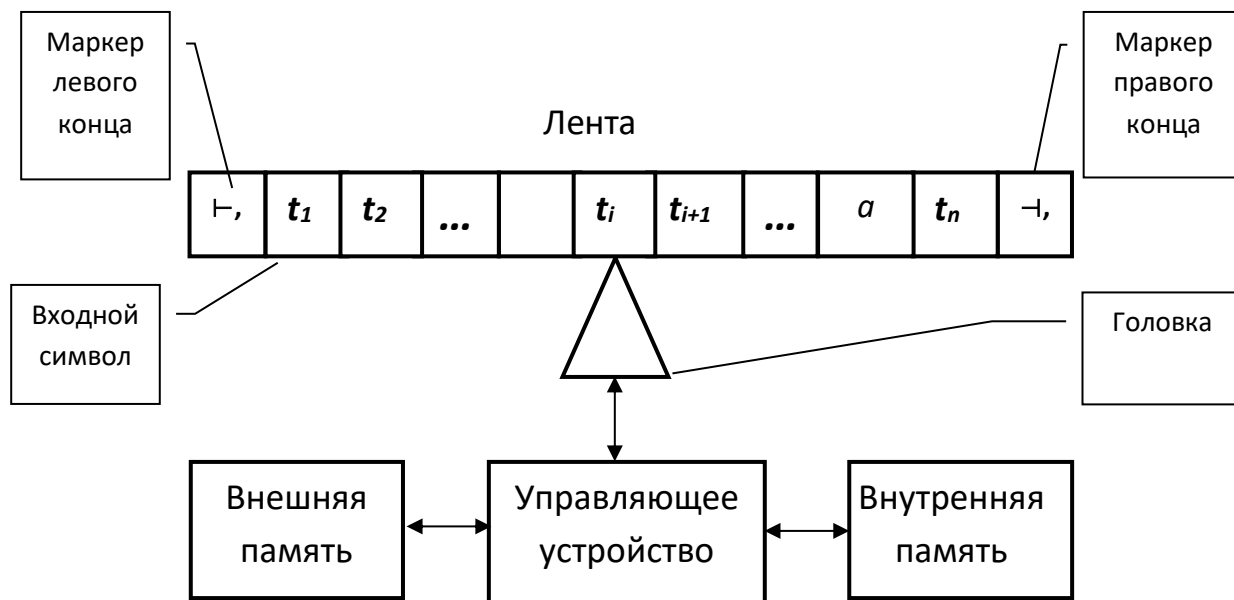


Рисунок 8.1. Состав и структура конечного

Лента – линейная последовательность ячеек, каждая из которых может хранить только один символ из некоторого конечного входного (выходного) алфавита.

Лента бесконечна, но занято на ней в каждый момент времени только конечное число ячеек. Граничные слева и справа от занятой области ячейки могут занимать специальные маркеры для обозначения начала и конца ленты. Маркер может стоять только на одном конце ленты или может отсутствовать вообще.

Входная (рабочая) головка – устройство, которое в каждый момент времени может обозревать только одну ячейку ленты. Головка может сдвинуться на одну ячейку влево или сдвинуться на одну ячейку вправо, либо остаться неподвижной на месте. Обычно предполагается, что головка только читает, т.е. во время работы автомата символы на ленте не меняются. Но можно рассматривать и такие распознаватели, головка которых

и читает и пишет. Поэтому головка может быть читающей и пишущей.

Рабочая память – вспомогательное хранилище информации, предназначенное для чтения и записи данных. Рабочая память может быть организована в виде динамической структуры данных (очереди или стека).

Управляющее устройство – устройство, которое управляет поведением автомата и имеет конечную внутреннюю память для хранения конечного множества состояний. Он управляет поведением автомата с помощью функции (отношения), описывающего, как меняются состояния в зависимости от текущего состояния и текущего входного символа, читаемого головкой, и текущей информации, извлеченной с рабочей памяти, если она имеется. Управляющее устройство также определяет направление сдвига рабочей головки и то, какую информацию записать в рабочую память.

Автомат определяется заданием конечного множества состояний управляющего устройства, конечного множества допустимых входных символов, начального состояния и множества заключительных состояний, а также функции переходов, которая по текущему состоянию и текущему входному символу, являющимися ее аргументами, указывает все возможные следующие состояния, являющиеся значениями этой функции. Работу автомата удобно описывать с помощью его конфигурации. Конфигурация автомата включает в себя:

- состояние управляющего устройства;
- содержимое входной ленты с положением входной головки;
- содержимое рабочей памяти вместе с положением рабочей головки, если она есть;
- содержимое выходной ленты, если она есть.

Конфигурация автомата может быть начальной, текущей и заключительной.

В *начальной конфигурации* внутренняя память содержит заранее записанный символ, обозначающий начальное состояние управляющего устройства; управляющее устройство находится в начальном состоянии; головка читает самый левый входной символ на ленте; если имеется рабочая память, то она имеет заранее установленное начальное содержимое.

В *текущей конфигурации* внутренняя память содержит заранее записанные символы текущих состояний управляющего устройства; управляющее устройство находится в одном из текущих состояний; головка читает ни самый левый и ни самый правый текущий входной символ ленты; если имеется рабочая память, то она имеет заранее установленное текущее содержимое.

В *заключительной конфигурации* внутренняя память содержит заранее записанные символы, обозначающие заключительные состояния управляющего устройства; управляющее устройство находится в одном из заключительных состояний; головка обзревает правый концевой маркер или если маркер отсутствует, сошла с конца входной ленты; если имеется рабочая память, то она удовлетворяет некоторым условиям.

Перед началом работы автомат находится в начальной конфигурации, т.е. во внутреннюю память записывается символ обозначения начального состояния управляющего устройства, во входную ленту – входная цепочка, и если предусмотрена рабочей памяти, то в нее – соответствующую информацию.

Автомат работает по программе, состоящей из конечной последовательности *тактов*. Каждый такт строится из текущей (начальной) и следующей (заключительной) конфигураций.

В начале такта в памяти читается символ текущего состояния управляющего устройства, во входной ленте читается текущий входной символ и исследуется информация в рабочей памяти, если она есть. Затем, в зависимости от текущего состояния и прочитанной информации, определяются действия автомата:

(1) входная головка сдвигается вправо, влево или стоит на месте;

(2) записывается новый символ в текущую ячейку входной ленты или прежний символ не изменяется;

(3) в рабочую память записывается некоторая информация, если она есть;

(4) записывается символ на выходную ленту, если она есть.

(5) управляющее устройство переходит в другое состояние и номер (символ) этого состояния записывается во внутреннюю память.

Итак, за один такт работы автомата входная головка может сдвинуться на одну ячейку влево, вправо или остаться на месте. В ходе работы автомата содержимое ячеек входной ленты не меняются, а содержимое ячеек выходной ленты и рабочей ленты могут изменяться.

Если автомат просматривает входную цепочку, выполняя последовательность тактов, которая начинается с начальной конфигурации и заканчивается заключительной конфигурацией, то он распознает эту цепочку.

Языком, распознаваемым автоматом называется множество цепочек, распознаваемых этим автоматом.

Примеры 8.1:

1. Таксафон служит примером реализации автомата: распознает введенную монету и переходит в состояние набора номера.

2. Банкомат может служить автоматом: распознает введенную карточку и переходит в состояние ввода пинкода.

3. Турникет в метро является автоматом: распознает введенный жетон и переходит в состояние открытие входа.

8.2. Конечный автомат

Конечные автоматы являются распознавателями регулярных языков. Сначала даются формальные определения недетерминированных и детерминированных конечных

автоматов, затем описываются языки, распознаваемые ими, а в конце доказывается их эквивалентность.

Конечные автоматы являются одним из простейших и наиболее распространенными среди распознавателей. В составе конечного автомата имеются *конечная лента, внутренняя память, внешняя память, головка, управляющее устройство*.

Конечный автомат может быть недетерминированным или детерминированным, но его головка должна быть односторонней и может сдвигаться только вправо. Их формальные определения даются так:

Определение 8.1. *Недетерминированный конечный автомат* (НКА) определяется семеркой $M = \langle Q, T, I, F, \vdash, \dashv, \Delta \rangle$, где:

Q – конечное множество состояний управляющего устройства;

T – конечное множество входных символов, $Q \cap T = \emptyset$;

I – множество начальных состояний управляющего устройства, $I \subseteq Q$;

F – множество конечных состояний управляющего устройства, указывающих, что входная цепочка распознается, $F \subseteq Q$;

$\vdash, \dashv$ – маркер начало, маркер конца ленты, $\vdash, \dashv \notin T$;

Δ – множество отношений переходов $\Delta \subseteq Q \times T^* \times \wp(Q)$, $\wp(Q)$ – множество всех подмножеств множества Q .

Детерминированный конечный автомат (ДКА) является частным случаем НКА.

Определение 8.2. Конечный автомат $M = \langle Q, T, I, F, \vdash, \dashv, \Delta \rangle$ называется *детерминированным*, если:

- (1) множество начальных состояний I содержит ровно один элемент;
- (2) для каждого перехода $\langle q, t, p \rangle \in \Delta$ выполняется $|t| = 1$;
- (3) для любого состояния $q \in Q$ и для любого символа $t \in T$ существует не более одного состояния $p \in Q$ со свойством $\langle q, t, p \rangle \in \Delta$;
- (4) остальные символы остаются такими же, как и в НКА.

Замечания 8.1:

1. Иногда вместо множества отношения переходов Δ , принимающего логическое значение ‘истина’ или ‘ложь’,

используется функция переходов δ , принимающая значение в виде символа из множества Q , где $\delta: Q \times T^* \rightarrow \mathcal{P}(Q)$ – в случае НКА и $\delta: Q \times T^* \rightarrow Q$ – в случае ДКА. От функции δ можно легко перейти к отношению Δ , положив

$$\Delta = \{ \langle q, \tau, \delta(q, \tau) \rangle : q \in Q, \tau \in T^* \}$$

2. В дальнейшем мы без особого упоминания, будем использовать как отношения переходов, так и функции переходов в зависимости от контекста. При этом для любых $q \in Q, p \in Q$ и $\tau \in T^*$ можно писать:

1) для отношения переходов: $\langle q, \tau, \{p\} \rangle$ – в случае НКА, $\langle q, \tau, p \rangle$ – в случае ДКА;

2) для функции переходов: $\delta(q, \tau) = \{p\}$ – в случае НКА, $\delta(q, \tau) = p$ – в случае ДКА.

3. Если мы намереваемся вместо отношения переходов использовать функцию переходов, то в формальном определении КА следует символ Δ заменить на δ , а остальные символы оставить без изменения со своими прежними значениями, т.е. получим $M = \langle Q, T, I, F, \vdash, \neg, \delta \rangle$.

Переходы КА можно изображать в виде *диаграммы*, в которой каждое состояние обозначается кружком, а переход – стрелкой. Стрелка из состояния $q \in Q$ в состояние $p \in Q$ помеченная цепочкой $\tau \in T^*$, показывает, что $\langle q, \tau, p \rangle$ (или $\delta(q, \tau) = p$) является переходом данного НКА. Каждое начальное состояние распознается по ведущей в него короткой стрелке. Каждое заключительное состояние отмечается на диаграмме двойным кружком.

1. Эквивалентны ли следующие грамматики

$$S \rightarrow ab, S \rightarrow aKSb, K \rightarrow bSb, KS \rightarrow b, K \rightarrow \varepsilon$$

и

$$S \rightarrow aAb, A \rightarrow \varepsilon, A \rightarrow b, A \rightarrow S, A \rightarrow bSbS ?$$

2. Эквивалентны ли следующие грамматики

$$S \rightarrow aD, D \rightarrow bba, D \rightarrow baDa, D \rightarrow aDaDa$$

и

$$S \rightarrow aaE, S \rightarrow abD, E \rightarrow bDD, D \rightarrow aaEa, D \rightarrow abDa, D \rightarrow ba ?$$

3. Какому классу принадлежит грамматика

$$S \rightarrow abba, S \rightarrow baa ?$$

4. Какому классу принадлежит грамматика

$$S \rightarrow AD, A \rightarrow aA, A \rightarrow \varepsilon, D \rightarrow bDc, D \rightarrow \varepsilon$$

5. Эквивалентны ли грамматика с правилами

$$S \rightarrow AB, A \rightarrow a|Aa, A \rightarrow a|Aa$$

и грамматика с правилами

$$S \rightarrow AS|SB|AB, A \rightarrow a, B \rightarrow b ?$$

6. Эквивалентны ли грамматика с правилами

$$S \rightarrow cE, E \rightarrow ddc, E \rightarrow dcEc, E \rightarrow cEcEc$$

и грамматика с правилами

$$S \rightarrow ccA, S \rightarrow cdB, A \rightarrow dBB, B \rightarrow ccAc, B \rightarrow cdBc, B \rightarrow dc?$$

Как можно описать однозначной грамматикой язык, который порожден неоднозначной грамматикой $E \rightarrow E + E | E * E | (E) | i$.

Примеры 8.2:

1. Для КА M_1 с одним переходом и параметрами: $Q = \{q, p\}$; $T^* = \{\tau\}$, $I = \{q\}$, $F = \{p\}$, $\delta(q, \tau) = p$ диаграмма показана на рисунке 8.2.

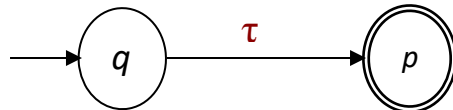


Рисунок 8.2. Диаграмма КА M_1 с одним переходом.

На рисунке 8.3 показана диаграмма переходов НКА M_2 , в которой в качестве меток ребер использованы регулярные выражения aaa , ab , b , ε . Такое представление облегчает построение диаграммы, делая её компактной и наглядной.

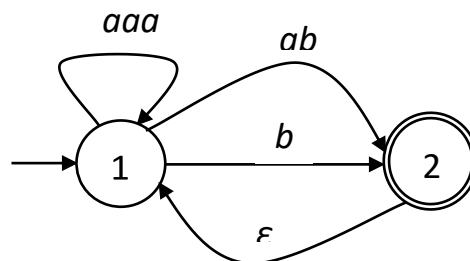


Рисунок. 8.3. Диаграмма КА M_2 с регулярным

3.КА M_3 для распознавания идентификаторов, которые состоят только из букв и цифр и начинаются с буквы, будет иметь следующие параметры $Q=\{1,2\}$, $T=\{b,d\}$, $I=\{1\}$, $F=\{2\}$, $\delta(1,b)=2$, $\delta(2,b)=2$, $\delta(2,d)=2$, где b – буква, d – цифра. Диаграмма КА M_3 представлена на рисунке 8.4.

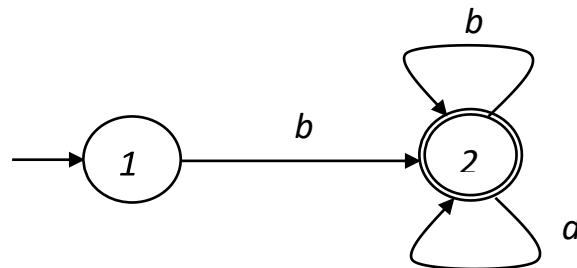


Рисунок 8.4. Диаграмма КА M_3 для идентификатора.

Замечание 8.3. Если в диаграмме имеются несколько переходов с общим началом и концом, то эти переходы называются *параллельными*. Обычно параллельные переходы на диаграмме изображаются одной стрелкой. При этом метки переходов разделяются запятыми.

Переходы КА можно представить в виде функции с помощью таблицы или команд.

На рисунке 8.5 представлена диаграмма КА M_4 с параллельными переходами для цепочек ab, b .

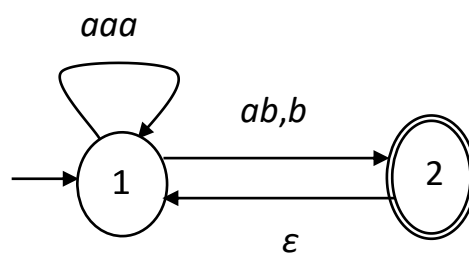


Рис. 8.5. Диаграмма КА M_4 . с параллельными переходами

Соглашение 8.1. Среди всех состояний КА выделяются q_s начальное состояние и q_f финальное состояние, здесь s и f понимаются не как числовые переменные, а как мнемонические знаки начала (*start*) и конца (*final*).

Примеры 8.3. В таблице 8.1 показана функция переходов δ КА M_5 , определенная множествами $Q = \{q_s, q_1, q_2, q_3\}$ и $T = \{t_1, t_2, t_3\}$.

Таблица 8.1. Значения функции переходов δ КА M_5 .

δ		В х о д		
		t_1	t_2	t_3
С о с т о я н и е	q_s	q_2	q_2	q_2
	q_1	q_3	q_s	q_s
	q_2	q_2	q_2	q_2
	q_3	q_3	q_2	q_s

Функция переходов из таблицы 8.1 можно представить в виде команд следующим образом:

$$\delta(q_s, t_1) = q_2, \delta(q_s, t_2) = q_2, \delta(q_s, t_3) = q_2,$$

$$\delta(q_1, t_1) = q_3, \delta(q_1, t_2) = q_s, \delta(q_1, t_3) = q_s,$$

$$\delta(q_2, t_1) = q_2, \delta(q_2, t_2) = q_2, \delta(q_2, t_3) = q_2,$$

$$\delta(q_3, t_1) = q_3, \delta(q_3, t_2) = q_2, \delta(q_3, t_3) = q_s.$$

Пусть задан КА M с начальным состоянием $q_s \in Q$, текущим состоянием $q \in Q$, заключительным состоянием $q_f \in Q$ и неиспользованной текущей входной цепочкой $\tau \in T^*$. Тогда можно дать следующее определение.

Определения 8.3:

1. Если головка обозревает самый левый символ входной цепочки τ , то пара $(q_s, \tau) \in Q \times T^*$ называется *начальной конфигурацией* КА;

2. Если головка обозревает текущий символ входной цепочки τ , то пара $(q, \tau) \in Q \times T^*$ называется *текущей конфигурацией* КА;

3. Если входная цепочка τ прочитана полностью, то пара $(q_f, \varepsilon) \in Q \times T^*$ называется *заключительной конфигурацией* КА;

Замечание 8.4. Содержательно конфигурация представляет собой "мгновенное описание" КА. Если представить, что исходная цепочка, принадлежность которой рассматриваемому языку надо

проверить, дано в ленте, то в конфигурации (q, τ) цепочка τ есть та часть исходной цепочки, которая пока осталась в ленте.

Такт КА определяется состоянием управляющего устройства и обозреваемым в этот момент входным символом. Сам такт состоит в изменении состояния управляющего устройства и сдвига головки на одну ячейку вправо.

Такт КА M задается бинарным отношением $\models_M$, определенным над его конфигурациями из множества $Q \times T^*$. При этом, если известен автомат, то букву M при отношении $\models_M$ можно опустить.

Пусть $t \in T$ – еще не прочитанный самый левый символ входной цепочки и для $q \in Q, p \in Q$ выполняется $\langle q, t, p \rangle \in \Delta$, то для цепочек $\tau \in T^*$ выполняется отношение $(q, \tau) \models (p, \tau)$, задающее такт автомата, которое означает, что автомат находясь в состоянии q и головка обозревает во входной ленте символ t , то КА M переходит в состояние p и головка сдвигается на ячейку вправо. При этом если $\tau = \varepsilon$, то считается входная цепочка прочитан полностью.

Примеры 8.4. Пусть $\tau = abba$. Тогда в диаграмме КА M_2 на рисунке 8.3 имеется такт в виде отношения $(1, abba) \models (2, ba)$.

Определение 8.4. $\models^k$ является k -той степенью отношения $\models$, если существует цепочка из $k+1$ конфигураций

$$(q_0, \tau_0), (q_1, \tau_1), (q_2, \tau_2), \dots, (q_{k-1}, \tau_{k-1}), (q_k, \tau_k)$$

такая, что для любого i ($1 \leq i \leq k$) выполняется отношение

$$(q_{i-1}, \tau_{i-1}) \models (q_i, \tau_i), \text{ где } q_0 = q_s, \tau_0 = \tau, q_k = q_f, \tau_k = \varepsilon.$$

Если для любого $i \geq 1$ или $i \geq 0$ выполняется $(q_0, \tau) \models^i (q_i, \varepsilon)$, то можно записать $(q_0, \tau) \models^+ (q_i, \varepsilon)$ или $(q_0, \tau) \models^* (q_i, \varepsilon)$ соответственно. Здесь через $\models^+$ обозначается транзитивное замыкание отношения $\models$, а через $\models^*$ – рефлексивное и транзитивное замыкание отношения $\models$.

Определение 8.5. Автомат M распознает входную цепочку τ , если выполняется отношение $(q_s, \tau) \models^* (q_f, \varepsilon)$.

Примеры 8.5. Пусть $\tau = aaaaab$. Тогда в КА M_2 на рисунке 8.3 выполняются следующие отношения $(1, aaaaab) \models (1, ab)$ и $(1, ab) \models (2, \varepsilon)$.

Определение 8.6. Если язык L состоит только из распознанных автоматом M входных цепочек, то этот язык распознается автоматом M и он обозначается через $L(M)$, т.е.

$$L(M) = \{\tau: \tau \in T^* \text{ \& } (q_s, \tau) \models^* (q_f, \varepsilon)\}.$$

Лемма 8.1. Если верно $(q_1, x) \models^* (q_2, \varepsilon)$ и $(q_2, y) \models^* (q_3, \varepsilon)$, то верно $(q_1, xy) \models^* (q_3, \varepsilon)$.

Доказательство. Для этого нужно провести индукцию по количеству тактов в программе работы КА, ведущей из конфигурации (q_1, x) в конфигурацию (q_2, ε) .

Примеры 8.6. Пусть для $M_6 = \langle \{q_s, q_1, q_f\}, \{0, 1\}, q_s, \{q_f\}, \vdash, \dashv, \delta \rangle$ имеются следующие отношения переходов:

$$\begin{aligned} &\langle q_s, 0, \{q_1\} \rangle, \langle q_s, 1, \{q_s\} \rangle, \langle q_1, 0, \{q_f\} \rangle, \langle q_1, 1, \{q_s\} \rangle, \\ &\langle q_f, 0, \{q_f\} \rangle, \quad \langle q_f, 1, \{q_f\} \rangle. \end{aligned}$$

КА M_6 распознает всех цепочек из нулей и единиц, в составе которых имеются подряд два нуля. Здесь состояния можно интерпретировать следующим образом:

q_s – начальное состояние показывает, что «подряд стоящие два нуля еще не встретились и начальный символ является нулем»;

q_1 – состояние показывает, что «подряд стоящие два нуля еще не встретились и начальный символ является нулем»;

q_f – финальное состояние показывает, что «подряд стоящие два нуля уже появились».

Можно заметить, что КА M_6 попав в состояние q_f остается в этом состоянии.

Для входной цепочки 01001 единственно возможная цепочка конфигураций, начиная с конфигурации $(q_0, 01001)$ будет так $(q_s, 01001) \vdash (q_1, 1001) \vdash (q_s, 001) \vdash (q_1, 01) \vdash (q_f, 1) \vdash (q_f, \varepsilon)$.

Итак, $01001 \in L(M_6)$.

Диаграмма этого автомата показана на рисунке 8.6.

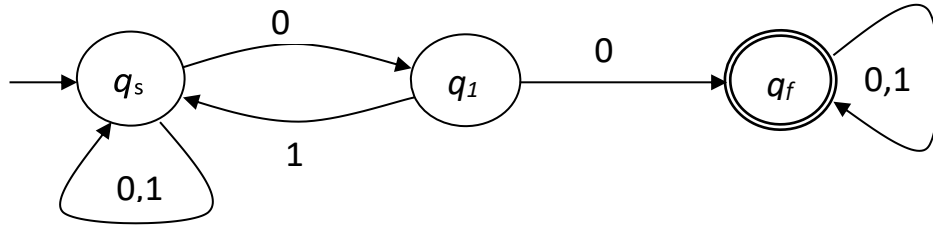


Рисунок 8.6. Диаграмма КА M_6 .

Определения 8.7:

1. *Путь* (path) КА – это кортеж $\langle q_0, r_1, q_1, r_2, \dots, q_n \rangle$, где $n \geq 0$ и $r_i = \langle q_{i-1}, \tau_i, q_i \rangle \in \Delta$ для каждого i , $1 \leq i \leq n$. При этом q_0 – начало пути, q_n – конец пути, $\tau_1 \dots \tau_n$ – метка пути, n – длина пути.

2. Путь называется *успешным* если его начало принадлежит I , а конец принадлежит F .

Замечание 8.5. Для любого состояния $q \in Q$ существует путь $\langle q \rangle$. Его метка ε , начало и конец совпадают.

Примеры 8.7. Рассмотрим КА M_2 на рисунке 8.3 Пусть $\tau = baaab$. Тогда путь $\langle 1, \langle 1, b, 2 \rangle, 2, \langle 2, \varepsilon, 1 \rangle, 1, \langle 1, aaa, 1 \rangle, 1, \langle 1, b, 2 \rangle, 2 \rangle$ является успешным. Его меткой является $baaab$, а длина равна 4, т.е.: $q_0=1, q_1=2, q_2=1, q_3=1, q_4=2$;

$r_1=\langle 1, b, 2 \rangle, r_2=\langle 2, \varepsilon, 1 \rangle, r_3=\langle 1, aaa, 1 \rangle, r_4=\langle 1, b, 2 \rangle$;

$\tau_1=b, \tau_2=\varepsilon, \tau_3=aaa, \tau_4=b$.

Используя понятие «путь» можно дать альтернативные определения уже введенным выше понятиям распознаваемым цепочки и языка.

Определения 8.8:

1. Цепочка $\tau \in T^*$ *распознается* (is recognized) КА M , если она является меткой некоторого успешного пути.

2. КА M *распознает* (recognizes) язык $L(M)$, если он состоит только из меток всех успешных путей.

Замечание 8.6. Если $I \cap F \neq \emptyset$, то язык, распознаваемый КА $M = \langle Q, T, \vdash, \neg, I, F, \Delta \rangle$ содержит пустую цепочку ε .

Примеры 8.8. Если КА $M_7 = \langle Q, T, \vdash, \neg, I, F, \Delta \rangle$ задан как $Q = \{q_1, q_2\}, T = \{a, b\}, I = \{q_1\}, F = \{q_1, q_2\}, \Delta = \{\langle q_1, a, q_2 \rangle, \langle q_2, b, q_1 \rangle\}$, то он является детерминированным и распознает следующий язык: $L(M_7) = \{(ab)^n : n \geq 0\} \cup \{(ab)^n a : n \geq 0\}$.

Диаграмма этого автомата показана на рисунке 8.7.

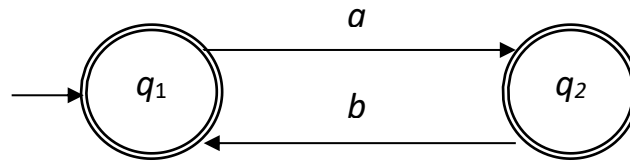


Рисунок 8.7. Диаграмма КА M_7 .

Определение 8.9. ДКА $M = \langle Q, T, \vdash, \dashv, I, F, \Delta \rangle$, называется *полным*, если для каждого состояния $q \in Q$ и для каждого символа $t \in T$ найдется такое состояние $p \in Q$, что $\langle q, t, p \rangle \in \Delta$, т.е. $\delta(q, t) = p$.

Примеры 8.9. На рисунке 8.8 показана диаграмма автомата M_8 со следующими параметрами $\Delta = \{ \langle 1, a, 2 \rangle, \langle 1, b, 3 \rangle, \langle 2, a, 3 \rangle, \langle 2, b, 1 \rangle, \langle 3, a, 3 \rangle, \langle 3, b, 3 \rangle \}$, $Q = \{1, 2, 3\}$, $T = \{a, b\}$, $q_s = \{1\}$, $F = \{1, 2\}$.

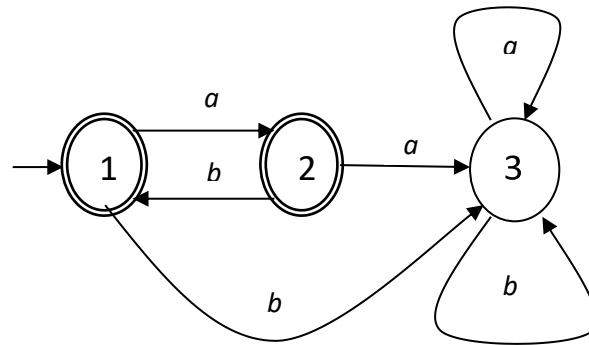


Рисунок 8.8. Диаграмма КА M_8

Задания 8:

1. Найти КА, распознающий язык $\{\alpha\beta: \alpha \in \{a, b\}^*, \beta \in \{a, b\}^*\}$.
2. Найти КА, распознающий язык $\{a, b\}^* \setminus (\{a^n: n \geq 0\} \cup \{b^n: n \geq 0\})$.
3. Найти КА, распознающий язык $\{a\xi b: \xi \in \{a, b\}^* \cup \{b\xi a: \xi \in \{a, b\}^*\}$.
4. Найти КА, распознающий язык $\{\tau \in \{a, b\}^*: |\tau|_a \geq 3\}$.
5. Найти КА, распознающий язык $\{a^m b^n a^m b^n: m, n \geq 1\}$.
6. Перечислить все конфигурации (q, τ) , удовлетворяющие условию $(1, abaacdsc) \models^* (q, \tau)$, в КА M_9 на рисунке 8.9.

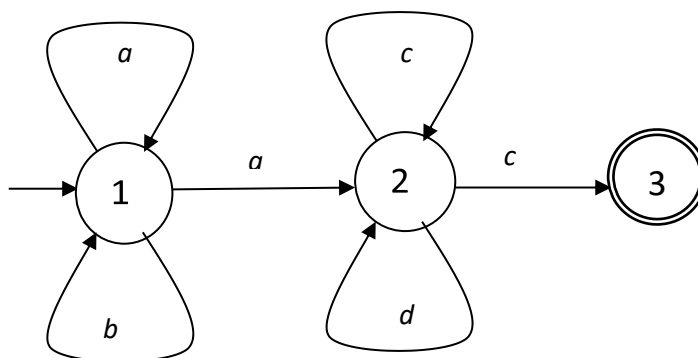


Рисунок 8.9. Диаграмма КА M_9 .

7. Определить такт автомата, если он задан в виде $M = \langle \{q_0, q_1, q_2, q_f\}, \{a, b, c\}, \delta, q_0, \{q_f\} \rangle$, где $\delta(q_0, a) = \{q_1, q_2\}$, $\delta(q_1, a) = \{q_1\}$, $\delta(q_1, b) = \{q_f\}$, $\delta(q_2, c) = \{q_f\}$, $L(M) = \{ac\} \cup \{a^n b : n \geq 1\}$.

8. Найти полный детерминированный конечный автомат для языка $(a \vee b)^*(aab \vee aba \vee abb)(a \vee b)^*$.

9. Найти полный детерминированный конечный автомат для языка $(b \vee c)((ab)^* c \vee (ba)^*)^*$.

10. Найти полный детерминированный конечный автомат для языка $(b \vee c)^*((a \vee b)^* c (b \vee a)^*)^*$.

Вопросы 8:

1. Является ли детерминированным КА M_{10} на рисунке 8.10.

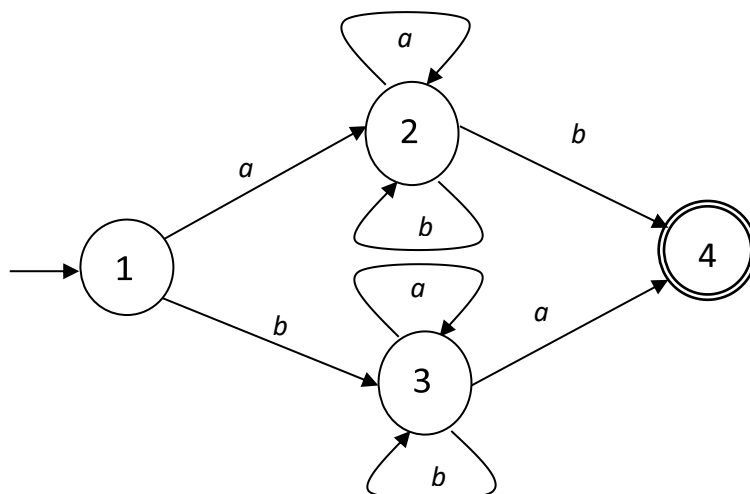


Рисунок 8.10. Диаграмма КА M_{10} .

2. Существуют ли КА, состояния q_1, q_2 и цепочки α, β, δ , такие что выполняются отношения $(q_1, \alpha\beta) \models^* (q_2, \beta)$ и $\neg (q_1, \alpha\delta) \models^* (q_2, \delta)$?
3. Как связаны $|Q|, |T|, |\Delta|, |\tau|$ и число достижимых из (q, τ) конфигураций, в смысле $\models^*$?
4. Каким автоматом можно распознать язык, порожденный регулярным выражением $(abab)\vee(aba)^*$?
5. Что содержит входная лента?
6. Чем определяется направление сдвига головки?
7. Из чего состоит конфигурация автомата?
8. Какие виды конфигурации имеются?
9. Из чего состоит распознаваемый автоматом язык?
10. Является ли полным детерминированный конечный автомат M_{11} с алфавитом $T = \{a, b, c\}$ на рисунке 8.11.

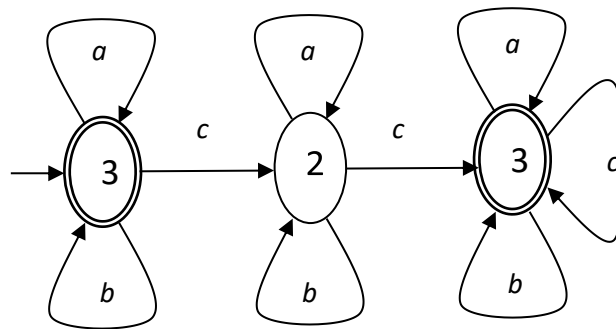


Рисунок 8.11. Диаграмма КА M_{11} .

11. Является ли полным детерминированный конечный автомат M_{12} с алфавитом $T = \{a, b\}$ на рисунке 8.12.

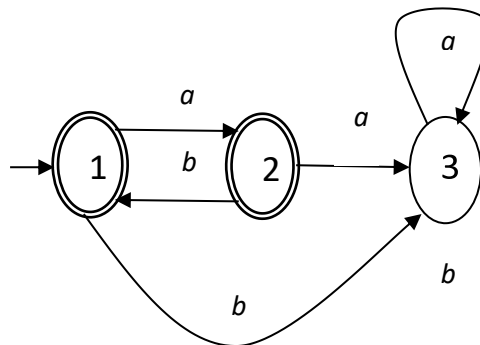


Рисунок 8.12 Диаграмма КА M_{12} .

12. Как выглядит граф перехода конечного автомата, соответствующего заданной грамматике?

$$G = \langle \{+, -, *\}, \{A, B, C\}, P, C \rangle.$$

$$P : A \rightarrow + \mid A +$$

$$B \rightarrow - \mid B - \mid A - \mid A +$$

$$C \rightarrow * \mid C * \mid B * \mid B - \mid A +$$

Тесты 8:

1. Конечный автомат переходит в то или иное состояние согласно:

- А) таблице переходов в памяти автомата;
- В) по данному заданию;
- С) рисункам;
- Д) направлениям;
- Е) содержанию.

2. Какой автомат называется детерминированным??

А) если для каждой допустимой конфигурации распознавателя, которая возникла на некотором шаге его работы, существует две конфигурации, в одну из которых распознаватель перейдет на следующем шаге работы;

В) если для каждой допустимой конфигурации распознавателя, которая возникла на некотором шаге его работы, существует единственно возможная конфигурация, в которую распознаватель перейдет на следующем шаге работы;

С) если распознаватель может иметь такую допустимую конфигурацию, для которой существует некоторое конечное множество конфигураций, возможных на следующем шаге работы;

Д) если распознаватель допускает чтение входных символов только в одном направлении («слева направо»);

Е) если распознаватель допускает, что считывающее устройство может перемещаться относительно цепочки входных символов в обоих направлениях: как вперед, от начала ленты к концу, так и назад, возвращаясь к уже прочитанным символам.

3. Конечный автомат – это пятерка $M = \langle Q, T, \delta, q_0, F \rangle$, где Q

А) конечное множество допустимых входных символов;

В) конечное множество состояний;

С) функцией переходов;

Д) начальное состояние;

Е) заключительное состояние.

Литература

1. Turetsky V.J. Mathematics and computer science.- Moscow: Infra-M, 2000.
2. Urginovich N. The Practical work on information technologies.- Moscow: BINOM, 2004.
3. Kuk D. The computer mathematics. - Moscow: Nauka, 1990.
4. Kozlov V.N. Mathematics and computer science.- Piter, 2004.
5. Chechkin A.V. Mathematical computer science.- Moscow: the Science, 1991.
6. Akimov O.E. The discrete mathematics: logic, groups, graphs.- Moscow: The Laboratory of base knowledge, 2001.
7. Novikov F.A. The discrete mathematics for programmers: the Textbook for high schools. –St-Petersburg: Piter, 2008.
8. Meyer B., Baudoin C. Programming methods. In two parts: Part 2. Translate from French Y.A. Pervina - Moscow: Mir, 1982.
9. Borubaev A.A., Pankov P.S. Discrete mathematics. Kyrgyz-Russian Slavic University, Bishkek, 2010.
10. <http://www.regentsprep.org/regents/math/algebra/AP1/Interval>
11. [http://en.wikipedia.org/wiki/Interval_\(mathematics\)](http://en.wikipedia.org/wiki/Interval_(mathematics))
12. Шарипбай А.А. Информатика, Алматы, Эверо, 2015, -313 с.
13. Шарипбай А.А. Теория языков и автоматов, Астана, ЕНУ им. Л.Н. Гумилева, 2015, -207 с.



Co-funded by the
Tempus Programme
of the European Union



ШАРИПБАЙ А.А.

МАТЕМАТИКА ДЛЯ КОМПЬЮТЕРНЫХ НАУК
(Учебное пособие)

Подписано к печати 29.05.2017 г. Бумага офсетная №1.
Формат 60X84 1/16 Объем 9,5 усл.печ.л.
Тираж 50 экз. Заказ 0168
Отпечатано в типографии ТОО «Мастер ПО»
010008, г. Астана, ул. Пушкина, 15-76
(ул. Кенесары, 93-76) тел.: 8 (7172) 223-418
e-mail.: masterpo08@mail.ru



ШАРИПБАЙ Алтынбен Амирулы, доктор технических наук, профессор, академик Международной академии информатизации, академик Академии педагогических наук РК, лауреат государственной премии РК в области науки, техники и образования, директор НИИ «Искусственный интеллект», профессор кафедры Информатики и информационной безопасности ЕНУ им.Л.Н.Гумилева (www.v-zerde.kz). Его научными интересами являются теоретические и прикладные проблемы информатики и информационных технологий: теория и технологии программирования, автоматизированные системы, искусственный интеллект, компьютерная лингвистика и электронное обучение. В этой области им были разработаны

методы формализации семантики продукционных и логических языков программирования, методы автоматизированной верификации программных и аппаратных средств. По результатам этих исследований он защитил докторскую диссертацию на тему «Верификация программных и аппаратных средств вычислительных машин и систем» по специальности «05.13.13 (05.13.11) - Вычислительные машины, системы и сети (Математическое, программное и техническое обеспечение)». Отдельные его научные результаты были внедрены в крупные научные центры: «Транслятор с языка имитации космических систем» в Петно-испытательном институте, г. Жуковский; «Система верификации цифровых схем» в Научно-производственном центре, г. Зеленоград; «Система параллельного программирования для многопроцессорного вычислительного комплекса» в Научно-исследовательском центре электронной вычислительной техники, г. Москва. В области информационных технологий он создал различные автоматизированные информационные системы. В частности, «Автоматизированная депозитарная система ценных бумаг» для Департамента государственного имущества и анционерных обществ г. Алматы; «Автоматизированная система учета коммунальных платежей» для Департамента коммунальных услуг и НСН в г. Алматы; «Автоматизированная система выборов»; «База данных информационного контроля» для Счетного комитета РК; «Система автоматизации создания электронных учебников» для СШ и вузов РК.

Он опубликовал более 400 научных трудов, издал 10 учебников и более 10 учебных пособий, 4 монографии и 5 терминологических и толковых словарей по информатике и вычислительной технике, получил более 30 свидетельств о государственной регистрации интеллектуальной собственности, участвовал в разработке многих государственных стандартов: 10 - в области информационных технологий, 9 - в области образования.

Под научным руководством А.Шарипбай подготовлены 5 докторов и 8 кандидатов наук, 6 докторов PhD по группе специальностей «Информатика, вычислительная техника и управление». Его научной школой создана математическая теория казахского языка, разработаны методы автоматизированного анализа и синтеза устных и письменных слов и предложений казахского языка, предложена технология создания электронных учебных изданий и др. Эти научные результаты в 1994-2014 годы были применены для создания многих электронных учебных изданий, системы дистанционного обучения казахскому языку, системы распознавания и синтеза казахской речи и других автоматизированных систем, в том числе учетных и экспертных систем, внедренных в различных государственных и негосударственных структурах и организациях РК.