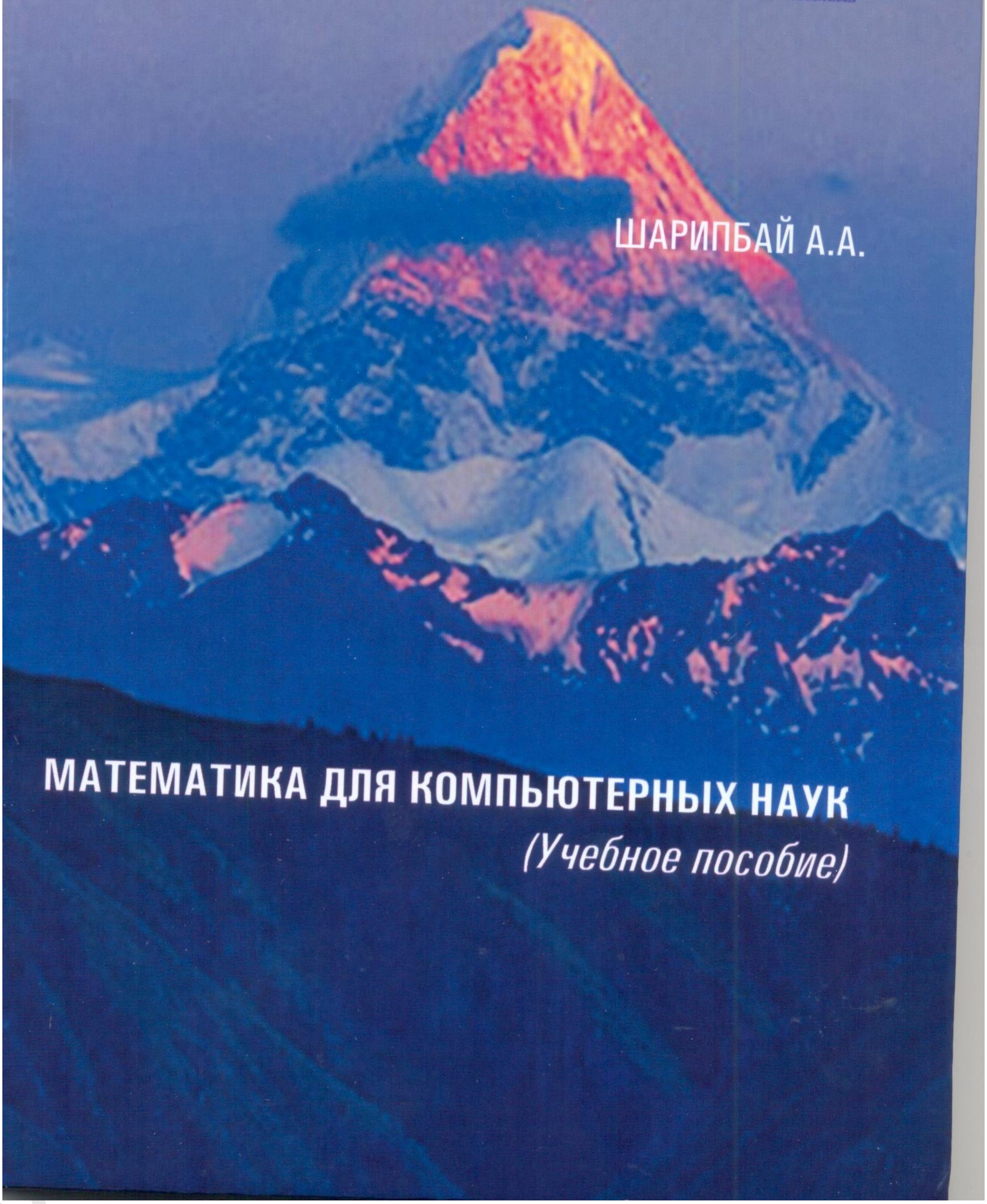
Co-funded by the
Tempus Programme
of the European Union

PROMIS



ШАРИПБАЙ А.А.

МАТЕМАТИКА ДЛЯ КОМПЬЮТЕРНЫХ НАУК

(Учебное пособие)

Co-funded by the
Tempus Programme
of the European Union

PROMIS

ШАРИПБАЙ А.А.

МАТЕМАТИКА ДЛЯ КОМПЬЮТЕРНЫХ НАУК
(Учебное пособие)

Астана 2017

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

Рецензенты:

Казиев Ғалым 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.

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

Автор:

Ш-25 Шарипбай А.А – директор НИИ «Искусственный интеллект», профессор кафедры «Информатики и Информационной безопасности» Евразийского национального университета имени Л.Н. Гумилева, доктор технических наук, лауреат Государственной премии Республики Казахстан

Шарипбай А.А.
Ш-25 Математика для компьютерных наук Учебное пособие, -Астана, 2017, - 162с.

Содержание учебного пособия «Математические основы информатики» 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, и все те, кто самостоятельно желает изучить математическую основу информатики.

Учебное пособие издается на средства Международного проекта TEMPUS - 544319-TEMPUS-1-2013-1-FR-TEMPUS-JPCR (2013-4530/001/00) “Professional 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

СОДЕРЖАНИЕ

I. ИНФОРМАЦИЯ И ДАННЫЕ **4**

I.1. Нотация, виды и измерение информации **4**
I.1.1. Форма и содержание информации 4
I.1.2. Объемные единицы и количество информации 10
I.1.3. Вероятностные единицы и количество информации 14

I.2. Виды и типы данных **19**
I.2.1. Постоянные и переменные виды данных 19
I.2.2. Числовые типы данных 22
I.2.3. Числовые операции и их свойства 26
I.2.4. Символьный тип данных 29
I.2.5. Символьные операции и их свойства 31
I.2.6. Логический тип данных 35
I.2.7. Логические операции и их свойства 37

I.3. Кодирование информации **43**
I.3.1. Кодирование логической и символьной информации 43
I.3.2. Системы счисления чисел 51

II. МНОЖЕСТВА И ОТНОШЕНИЯ **60**

II.1. Виды множеств и операции над ними **60**
II.1.1. Понятие множества и операции над ними 60
II.1.2. Отношения и способы их представления 69
II.1.3. Числовые множества и интервалы 79

III. ОСНОВЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ **87**

III.1. Логические исчисления **87**
III.1.1. Исчисление высказывания 87

III.2. Логические законы и выводы **100**
III.2.1. Тавтологии и свойства эквивалентности 100
III.2.2. Логические выводы 105

IV.1. Понятие и виды графа **117**
IV.1.1. Элементы и способы представления графов 117
IV.1.2. Деревья 126

V. ЯЗЫКИ, ГРАММАТИКИ И АВТОМАТЫ **133**

V.1. Механизмы порождения языка **133**
V.1.1. Основные понятия формальных языков 133
V.1.2. Формальные грамматики 138
V.1.3. Регулярные множества и регулярные выражения 145

V.2. Механизмы распознавания языка **151**
V.2.1. Конечные автоматы 151

ЛИТЕРАТУРА **167**

I. ИНФОРМАЦИЯ И ДАННЫЕ

I.1. Нотация, виды и измерение информации

I.1.1. Форма и содержание информации

Информация является основным объектом информатики и происходит от латинского слово «*Information*», которое означает передать сведения, сообщить о чём–либо (лице, предмете, факте, событии, явлении, процессе), независимо от формы его представления.

Понятие «Информация» не имеет точного (математического) определения. Если мы попытаемся ему дать точное определение, то опять придем к другому неопределенному понятию и это может продолжаться бесконечно. Однако мы, не зная его точного определения, можем принимать, понимать, сохранять, обрабатывать и, если надо передавать другому субъекту. С учетом сказанных можно дать следующее интуитивное (не точное) определение понятия «Информация».

Информация – это отражение свойств и отношений материальных и нематериальных объектов и субъектов окружающего мира. Это означает, что каждая информация характеризуется своей *формой* (*сообщением*) и своим *содержанием* (*значением*).

Вообще, когда мы говорим о сообщении нам надо помнить о существовании его *передатчике* и *приемнике*. Ими могут быть живые организмы (люди, животные, птицы, рыбы, насекомые и др.) и технические устройства (телефонные аппараты, радио станции и радио приемники и др.). Например, если приемником является человек, то он принимает сообщение своими органами чувств.

Сообщение от передатчика к приемнику передается посредством специальной среды, называемой *каналом связи*. Например, в качестве такой среды для звукового сообщения можно взять воздух, в котором могут распространяться звуковые волны, а каналом связи для письменного сообщения может быть бумага, на которой можно написать текст.

Схема передачи информации показана на рисунке I.1.1.А.

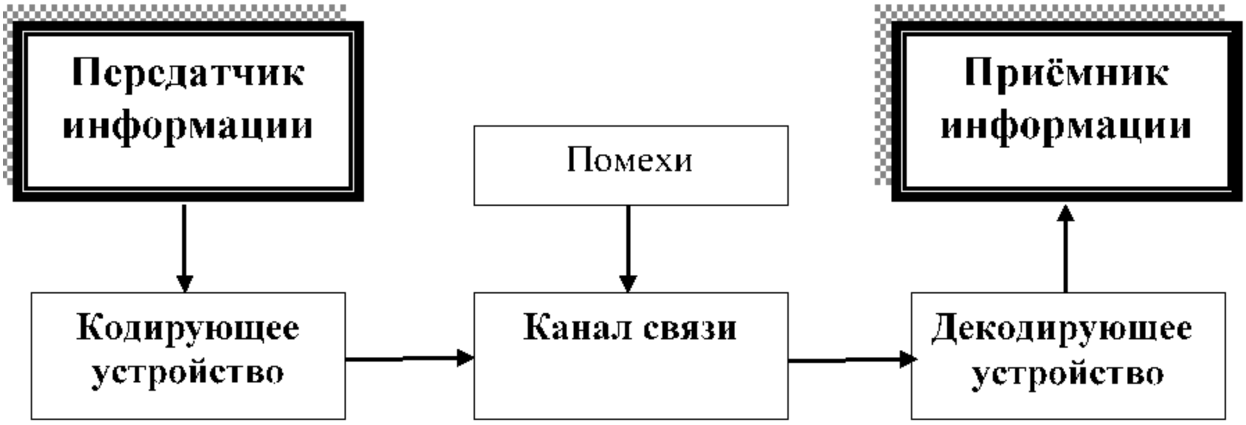


Рисунок I.1.1.1. Схема передачи информации.

Сообщение является носителем значения информации, т.е. значение информации определяется вместе с её сообщением.

Поэтому между передатчиком и приемником должно быть предварительное соглашение о виде (представлении) и значения сообщения. Такое соглашение называется *правилом интерпретации*.

***Замечание I.1.1*:**

1. Разноинтерпретированное одно и тоже сообщение является носителем разного значения. Например, читатели в зависимости от своих точек зрения могут воспринять одну и ту же журнальную статью по разному.
2. Одно и тоже значение может быть передано разным сообщением. Например, доклад одного и того человека на одну и ту же тему представленный, на разных языках.

Интерпретация заданного сообщения берется из общего правила, которое может применяться к множеству сообщений, построенных по единому закону. Например, если сообщение задано в виде предложения на известном естественном языке, то правило интерпретации такого предложения можно взять из общего правила интерпретации, применяемого ко всем предложениям этого языка. Итак к общему правилу интерпретации относится и правило определения значений чисел по их форме записи. Также часто используются в качестве сообщения утверждения, имеющие значения «истино» или «ложь» и вопросы, ответы которых могут быть «*да*» или «*нет*». Например, «*Снег имеет белый цвет*», «*Человек никогда не умирает*», «*Интересно ли изучать информатику*?», «*Понятно ли сказанное*?». Интерпретация таких сообщений зависит наряду от вышесказанных, также от жизненного опыта и сознания приемника.

При передаче (приеме) сообщения состояние передатчика (приемника) изменяется в зависимости от времени. Поэтому можно рассматривать описание материально–энергетического состояния передатчика (приемника) как функцию $\boldsymbol{x}(\boldsymbol{t})$, зависящую от времени $\boldsymbol{t}$. Эта функция $\boldsymbol{x}(\boldsymbol{t})$ может быть и непрерывной, и дискретной (разрывной). В зависимости от этого сообщение информация может быть *непрерывным* (*аналоговым*) и *дискретным* (*цифровым*). Например, к непрерывным сообщениям информации можно отнести график непрерывной функции температуры среды, изменяющейся во времени, а к дискретной информации – сообщение информации с помощью знаков некоторого языка общения.

Любую непрерывную информацию можно превратить в дискретную информацию, это называется *дискретизацией*. Для дискретизации функции среди её бесконечно многих значений берется ограниченное число значений, которые могут характеризовать остальные значения:

1) ось абциссы графика (область определения) функции разбивается на равные отрезки с помощью конечного числа точек $\boldsymbol{t_1}$, $\boldsymbol{t_2}$,..., $\boldsymbol{t_n}$ и считается, что в каждом отрезке значение функции постоянно, например, считается равным среднему значению в этом отрезке;

2) Спроектировав значение функции в каждом отрезке на ординатную ось графика (область изменения) функции, нужно найти точки $\boldsymbol{x_1}$, $\boldsymbol{x_2}$,..., $\boldsymbol{x_n}$.

Найденные таким образом точки $\boldsymbol{x_1}$, $\boldsymbol{x_2}$,..., $\boldsymbol{x_n}$ будут считаться дискретно приближенным представлением непрерывной функции $\boldsymbol{x}(\boldsymbol{t})$. Его точность можно неограниченно улучшать, уменьшая отрезки в области определения функции, до удовлетворения необходимого требования.

Графики непрерывной функции $\boldsymbol{x}(\boldsymbol{t})$ и её дискретизации показаны на рисунке I.1.1.2.

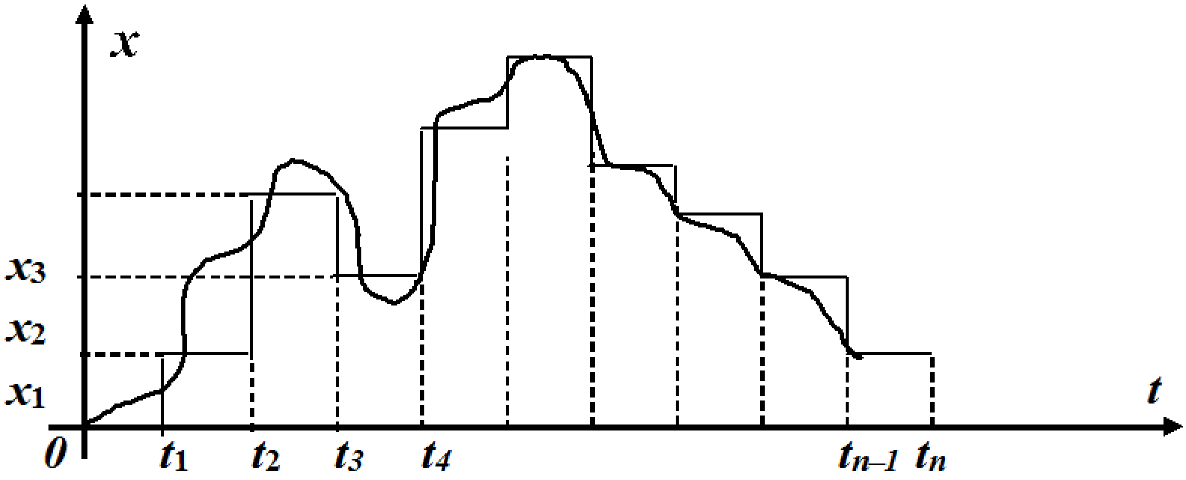


Рисунок I.1.1.2. Дискретизация непрерывной функции

Возможность дискретизации непрерывной информации позволяет представлять сообщение этой информации с помощью символов алфавита некоторого языка общения. Это очень важно для *приема*, *представления*, *хранения*, *обработки и передачи* информации с помощью компьютера.

Сообщение дается в материально–энергетическом виде (например: света, звука, движения, символа и т.п.). Иначе говоря, форма есть выражение некоторого языка. К таким языкам относятся:

- естественные языки (казахский язык, русский язык, английский язык и др.);
- математический язык (множество выражений, содержащих условные обозначения свойств и отношений математических объектов);
- музыкальный язык (ноты – множество выражений, содержащих условные обозначения свойств и отношений звуков в звуковом ряде);
- язык глухонемых (множество выражений, содержащих условные движения лица и рук);

• искусственные языки (языки программирования, языки спецификации, языки проектирования и др.);

Поскольку нас интересуют способы представления и обработки информации только на цифровых (дискретных) компьютерах, то мы будем рассматривать только виды дискретной информации и соответствующие *нотации* (искусственные языки), которые представляют собой систему условных символов (обозначений), используемых для представления понятий и их взаимоотношений, а также правила их применения в какой-либо области знаний или деятельности.

Нотация имеет свой алфавит, состоящий из упорядоченных в определенном смысле символов (букв), которые будут использоваться для построения слов – последовательности букв, имеющая определенный смысл. Построение слов осуществлялся в соответствии с синтаксическими правилами.

Примеры I.1.1.

Чтобы заметить связи между сообщением и значением рассмотрим несколько примеров. Они приведены в таблице I.2.1.

Таблица I.1.1. Примеры сообщения.

№	1–ое сообщение	2–ое сообщение
1.	Мен жақсы оқимын.	Я учусь хорошо.
2.	Из пламени получается лед.	1>3
3.	Ты меня понял?	X=0?
4.	XXI	21
5.	Завтра пойдет снег.	Лампа погасла.
6.	☼♀♂♥	© ®
7.	I speak.	Мен сөйлеймін.

В примерах 1,2,3,4 показано, что двумя разными сообщениями можно передать одно и то же значение: в 1-м примере предложения на казахском и русском языках имеют одно и то же значение, во 2-м примере значение «ложь» имеют два сообщения. В 3-м примере приведены вопросы, требующие ответы «да» или «нет». В 4-м примере одно и то же число представлено двумя способами.

В 5-м примере показана возможность передачи нескольких значений одним и тем же сообщением: по первому сообщению понимается, что завтра похолодает или можно покататься на санях.

В 6-м примере показана возможность общаться по предварительным условным обозначениям. Значения таких сообщений доступны только тем, кто знает значение этих знаков.

В 7-м примере одно значение передано предложениями двух естественных (английского и казахского) языков.

Задание I.1.1. Какому языку относятся эти выражения:

1. Республика Казахстан независимая страна.
2. $a + (b+c) = (a+ b)+ c$.
3. .

Вопросы I.1.1.

1. Чем характеризуется информация?
2. Какие есть виды языков общения?
3. Какая связь имеется между сообщением и значением?

Тесты I.1.1.

1. К какому языку относится данное выражение (a+b+c)?

A) Математический;

B) Логический;

C) Мимика;

D) Естественный;

E) Химический.

3. Между передатчиком и получателем какое предварительное соглашение имеется о виде сообщения и о его значении?

A) Интерпретация;

B) Интеграция;

C) Интервенция;

D) Информатизация.

E) Интерполяция;

4. Какая связь между сообщениями, представленными римскими цифрами XIII и арабскими цифрами 13 и их значениями?

A) Одно и тоже значение числа написано двумя способами;

B) Многозначное сообщение;

C) Однозначное сообщение;

D) Одно значение и одно сообщение;

E) Нет никакой связи.

I.1.2. Объемные единицы и количество информации

Определение понятия «Количество информации» очень трудно. Для этого сначала нужно определить единицу измерения информации. Её можно определить двумя способами: *объемный* и *вероятностный*. Оба этих способа стали известны одновременно в 40-х годах XX века. Их создали ученые США, одни из основателей науки информатики Джон фон Нейман и Клод Шеннон.

Джон фон Нейман самый первый показал возможность построения компьютеров, это привело к определению меры информации объемным способом, а Клод Шеннон определил меры информации вероятностным способом.

Минимальная неделимая единица информации называется ***bit* – *бит***, происходит от двух английских слов ***bi****nary dig****it*** – двоичная цифра. Причиной этого стало удобство для разработчиков компьютера работать двоичными числами при сохранении и обработки информации в компьютере: физический элемент, имеющий только два устойчивых состояния, может быть реализован многочисленными устройствами, два устойчивых состояния обозначаются двоичными цифрами 0 и 1. Например, легко создать устройство, показывающее наличие или отсутствие электрического тока, измеряющее низкий или высокий уровень напряжения, обнаруживающее полярность намагничивания и др.

Объемом информации, записанной двоичными цифрами 0 и 1, называется число двоичных цифр, применяемых в этой записи, это число всегда будет целым.

Следующую объемную единицу измерения информации называют *byte* – *байт*, она состоит из 8 битов, т.е. 1 байт = 2^3 бит. В одном байте можно раздельно записать 2^8 = 256 различных друг от друга символов, т.е. емкость одного байта составляет 256 битов. Это означает, что с помощью одного байта можно представить 256 различных (не повторяющихся) сообщений.

Объемное количество информации является очень большим числом. Поэтому для удобства применения определены крупные объемные единицы измерения. Эти единицы измерения кратны двум, т.е. они должны быть степенями двойки. Список объемных единиц измерения информации приведен в таблице I.1.2:

Таблица I.1.2. Список объемных единиц измерения информации.

№	**Название**	**Символ**	**Степень**
1	*byte* – байт	Б	10^0
2	*kilobyte* – килобайт	Кб	10^3
3	*megabyte* – мегабайт	Мб	10^6
4	*gigabyte* – гигабайт	Гб	10^9
5	*terabyte* – терабайт	Тб	10^{12}
6	*petabyte* – петабайт	Пб	10^{15}
7	*exabyte* – эксабайт	Эб	10^{18}
8	*zettabyte* – зеттабайт	Зб	10^{21}
9	*yottabyte* – йоттабайт	Йб	10^{24}

Примеры I.1.2.

1. Если в книге 400 страниц, в каждой странице 50 строк, а в каждой строке 50 символов, то объем книги в байтах будет 400*50*50 = 1 000 000 Байт = 1 Мб, т.е. в диске объемом 1 Гб можно сохранить 1000 таких книг.
2. Если размер электронной книги составляет 10 мегабайт и емкость электронной библиотеки составляет 100 гигабайт, то в ней можно хранить 100*1024/10 = 102400 электронных книг.
3. Если размер электронной книги составляет 10 мегабайт и емкость электронной библиотеки составляет 100 терабайт, то в ней можно хранить 100*1024*1024/10 = 104857600 электронных книг.

Задания I.1.2.

1. Назовите способы измерения количества информации.
2. Найти объемный размер в слове «РИМ».

3. Укажите емкость одного байта.

Помощь:

1. Способы Джон фон Неймана и Клода Шеннона.

2. Сохраняемая в компьютере информация (слово, число, рисунок, компьютерная программа) записывается в двоичных цифрах.

3. Нужно указать количество битов в одном байте.

Вопросы I.1.2.

1. Как называется минимальная единица информации?

2. Кто определил объем информации?

3. Чему должны быть кратны крупные единицы измерения информации?

Тесты I.1.2.

1. Сколько битов в одном байте?

A) 2;

B) 10;

C) 8;

D) 0;

E) 17

2. Сколько мегабайтов будет один ГБ?

A) 1040;

B) 1024;

C) 10024;

D) 124;

E) 102400.

3. Сколько гигабайтов будет один ТБ?

A) 10024;

B) 1040;

C) 1024;

D) 124;

E) 102400.

I.1.3. Вероятностные единицы и количество информации

До введения и обсуждения понятия «Вероятностное количество информации», рассмотрим один опыт, относящий к теории вероятности. В качестве примера можно взять бросание *N гранной игральной кости* (самая распространенная $N = 6$). Результатом считается выпадание грани, с надписью цифр 1, 2,..., N.

Введем числовую меру для измерения неопределенности, назвав её ***энтропией*** и обозначим через H. Данные N и H будут между собой в следующем функциональном отношении:

$$H = f(N), \qquad (1.1)$$

где функция f является неотрицательной и возрастающей для наших $N = 1, 2, \ldots, 6$.

Рассмотрим более подробно бросание игральной кости:

1) подготовка к бросанию кости: её исход неизвестен, т.е. есть какая-то неопределенность, обозначим её через $H1$;

2) игральная кость брошена: получена информация об исходе опыта, количество этой информации обозначим через I ;

3) обозначим неопределенность этого опыта после его осуществления через $H2$;

В качестве количества информации в ходе осуществления опыта можно взять разницу неопределенностей, полученных до опыта и после опыта:

$$I = H1 - H2. \qquad (1.2)$$

Очевидно, что в случае получения конкретного результата, ранее полученная неопределенность исчезает, т.е. $H2 = 0$. Таким образом, количество информации после опыта будет совпадать с начальной энтропией, т.е. $I = H1$. Говоря по-другому, неопределенность в опыте совпадает с информацией об исходе этого опыта. Здесь значение $H2$ может не равняться нулю, например, в ходе опыта надпись следующей грани выпадания больше 3.

Следующим важным обстоятельством является определение вида функции f в формуле (1.1). Если через N обозначить число граней, а через M – число бросаний игральной кости, то общее число исходов, определяемое векторами длины *M* и состоящих из *N* знаков, будет равно *N* в степени *M*

$$X = N^M \qquad (1.3)$$

Например, в случае двух бросаний кости с шестью гранями имеем: $X = 6^2 = 36$. Фактически каждый исход *X* является некоторой парой (X1, X2), где *X1* и *X2* – соответственно исходы первого и второго бросаний, а *X* – общее число таких пар.

Ситуацию с бросанием *M* раз кости можно рассматривать как некую сложную систему, состоящую из независимых друг от друга подсистем - "однократных бросаний кости". Энтропия такой системы в *M* раз больше, чем энтропия одной системы (принцип аддитивности энтропии):

$$f(6^M) = M \cdot f(6)$$

Данную формулу можно распространить и на случай любого *N*:

$$f(N^M) = M \cdot f(N) \qquad (1.4)$$

Теперь прологарифмируем левую и правую части формулы (1.3):

$$\ln X = M \cdot \ln N,$$

далее *M* находится так:

$$M = \frac{\ln X}{\ln N}.$$

Подставляем полученное для *M* значение в формулу (1.4):

$$f(X) = \frac{\ln X}{\ln N} \cdot f(N).$$

Обозначив через *K* положительную константу $\frac{f(N)}{\ln N}$, получим:

$$f(X) = K \cdot \ln X,$$

или, с учетом (1.1), $H = K \cdot \ln N$.

Обычно принимают $K = \frac{1}{\ln 2}$. Отсюда получается *формула Хартли*:

$$H = log_2 N \qquad (1.5)$$

Важным при введении какой-либо данные является вопрос о том, что принимать за единицу ее измерения. Очевидно, *H=1* при *N* = 2. Иначе говоря, в качестве единицы принимается количество информации, связанное с проведением опыта, состоящего в получении одного из двух равновероятных исходов (примером такого опыта может служить бросание монеты, при котором возможны два исхода: "орел", "решка"). Такая единица количества информации называется "*Бит*".

Все N исходов рассмотренного выше опыта являются равновероятными и поэтому можно считать, что на "долю" каждого исхода приходится одна N-я часть общей неопределенности опыта:

$$\frac{\log_2 N}{N}.$$

При этом вероятность i-го исхода P_i равняется, очевидно, $1/N$. Таким образом, э**нтропия** находится по *формуле Шеннона* так:

$$H = \sum_{i=1}^{N} P_i \cdot \log_2(\frac{1}{P_i}) \qquad (1.6)$$

Та же формула (1.6) принимается за меру энтропии в случае, когда вероятности различных исходов опыта *неравно вероятны* (т.е. значения P могут быть различными).

Замечание I.1.3:

Соотношение между объемным и вероятностным количеством информации неоднозначное. Не всякий текст, записанный двоичными символами, допускает измерение объема информации в вероятностном смысле, но заведомо допускает его в объемном. Далее, если некоторое сообщение допускает измеримость количества информации в обоих смыслах, то они не обязательно совпадают, при этом вероятностное количество информации не может быть больше объемного.

Примеры I.1.3.

1. Рассмотрим алфавит, состоящий из двух знаков 0 и 1. Если считать, что со знаками 0 и 1 в двоичном алфавите связаны одинаковые вероятности их появления ($P(0) = P(1) = 0{,}5$), то по формуле (1.5) количество информации на один знак при двоичном кодировании будет равно

2.

$$H = log_2 2 = 1\text{Бит}.$$

Таким образом, количество информации (в битах), заключенное в двоичном слове, равно числу двоичных знаков в нем.

3. Определим количество информации, связанное с появлением каждого символа в сообщениях, записанных на русском языке. Будем считать, что русский алфавит состоит из 33 букв и знака "пробел" для разделения слов. По формуле (1.5)

$$H = \log 34 = 5 \text{ Бит.}$$

Однако в словах русского языка (равно как и в словах других языков) различные буквы встречаются неодинаково часто. По формуле (1.6): *H= 4,72 Бит*.

Задания I.1.3.

1. Пусть в указанном ящике имеются 16 разноцветных шаров. Нужно вычислить объем информации, используя энтропию, чтобы вытащить белый шар.

2.Определите количество информации, связанное с появлением каждого символа в сообщениях, записанных на казахском языке в 42-х буквенном алфавите.

3.Определите количество информации, связанное с появлением каждой двоичной цифры в сообщениях, записанных в одном мегабайте.

Помощь:

1. $H = \sum_{i=1}^{N} P_i \cdot \log_2 (\frac{1}{P_i})$.

2. Для определения количество информации используйте формулу (1.5).

3. Надо учесть, что в одном мегабайте находятся 10^6 битов.

Вопросы I.1.3.

1. Какими свойствами обладает функция для вычисления энтропии?

2. В чем заключается принцип аддитивности энтропии?

3. Как называется величина, измеряющая неопределенность?

4. Отношение между объемными и вероятностными единицами измерения информации однозначно?

5. Зачем нужна формула Хартли?

Тесты I.1.3.

1. Что является минимальной неделимой единицей объемного измерения информации?

A) Терабайт (tera***byte***);

B) Байт (Byte);

C) Мегабайт (mega***byte***);

D) Гегабайт (giga***byte***);

E) Бит (bit).

2. Как называется величина измерения неопределенности по Шенону?

A) Эктропия;

B) Энтропия;

C) Величина;

D) Информация;

E) Байт.

3. Указать формулу Хартли?

A) $H = log_2 N$;

B) $H = \lg N$;

C) $H = \ln N$;

D) $H = \ln (N–1)/2$;

E) $H = ln(N - 1)^2$.

I.2. Виды и типы данных

I.2.1.Постоянные и переменные виды данных

В информатике понятие «*Информация*» часто заменяют на понятие «*Данные*», при этом сообщение информации расматривается как имя данных, а значение информации – как значение данных.

Данные, в зависимости от способов придания значений своим именам, подразделяются на *постоянные данные* (*постоянные*) и *переменные данные* (*переменные*).

Если значения данных определяются во время построения общих *правил интерпретации* используемого языка общения, то такие данные будут *постоянными*. Иначе говоря, при именовании постоянной одновременно определяются её значения. Вместе со значением станет известным и тип постоянной. Например, если цепочку 135, образованную из цифр 1, 3 и 5, рассматривать как имя постоянной, то это значение будет целое число «сто тридцать пять». Если из этих же цифр построить другую цепочку 315, то она будет именем постоянной со значением «триста пятьнадцать», являющимися целым числом.

Отсюда следует следующее утверждение: *имена*, *значения и типы постоянных* не изменяются, они определяются одновременно.

Если значения данных изменяются во время их обработки, то такие данные будут *переменными*. При этом тип значения переменной не должен изменяться. В противном случае во время обработки этой переменной появятся трудности, связанные с несовместимостью типов данных, участвующих в обработке.

В информатике для именования переменной используется понятие «*Идентификатор*».

Идентификатор – цепочка с ограниченной длиной, которая начинается с буквы и состоит из букв и цифр.

Значение переменной описывается своим типом, который однозначно определяет:

а) допустимые значения, которые может иметь объект описываемого типа;

б) допустимые операции, которые могут быть применимы к объекту описываемого типа.

Вообще, значения данных подразделяются на *численные*, *символьные* и *логические* типы.

Для обработки данных необходимо применить операции к этим данным. А для того чтобы применить операции нужно знать их определения, обозначения и свойства.

Операции, в зависимости от того, что над какими типами они определены подразделяются на *числовые операции*, *символьные операции* и *логические операции*.

Следует отметить, что любая из этих операций определена и исследована в различных разделах математики. Например, все символьные операции определены в разделе «Математическая лингвистика», которая исследует состав и свойства языков, все логические операции определены в разделе «Математическая логика», а все числовые операции – в других разделах математики (арифметика, алгебра и др.).

Можно определить свойства операций, определенных на каждом типе данных. Эти свойства группируются и образуют *аксиоматику* касательно этих данных.

В предлагаемых аксиоматиках, предназначенных для данных различных типов, имеются много сходства. Они показывают эквивалентность, в том числе имеются закономерности такие как *коммутативтность*, *ассоциативность* и *дистрибутивность*.

Такие закономерности упрощают сложное выражение, сокращая количество операций, и облегчают его вычисление.

Примеры I.2.1.

1. N12B – идентификатор.
2. 7X – не идентификатор, потому что начинается с цифры 7.
3. A+B – не идентификатор, так как в нем есть знак «+».

Задания I.2.1. Будут ли идентификаторами такие данные?

1. A–B;
2. XY;
3. C6R7.

Помощь:

Идентификатор начинает с буквы и состоит из букв и цифр.

Вопросы I.2.1.

1. Какие типы имеются у данных?
2. Зачем нужен идентификатор?
3. Чем различаются постоянные и переменные данные?

Тесты I.2.1.

1. Каким понятием можно заменить информацию?

A) Данные;
B) Идентификатор;
D) Сообщение;
C) Представление;
E) Определение.

2. У каких данных не изменяются имя, значение и тип?

A) Постоянные данные;
B) Определенные данные;
D) Переменные данные;
C) Не точные данные;
E) Неопределенные данные.

3. Как подразделяются данные в зависимости от способов присваивания значения?

A) Постоянные и переменные данные;
B) Постоянные и непостоянные данные;
C) Переменные и непостоянные данные;
D) Непостоянные и точные данные;
E) Действительные и непостоянные данные.

I.2.2. *Числовые типы данных*

Числовой тип образуется из множества чисел, операции над этими числами и свойствами этих операций. Они подразделяются на три: *целые числа*, *вещественные числа и комплексные числа*.

Замечание I.2.2.

В информатике обработка числовых данных может потребовать различные системы счисления чисел. Основаниями таких систем могут быть 2,3,4,... Разница между ними только в методах обозначения значений чисел, а виды операции над числами и их свойства будут одинаковыми. Поэтому сначала рассмотрим хорошо знакомую нам запись чисел в десятичной системе счисления, операции над ними и свойства этих операций, так как все сказанное, связанное с десятичной системой счисления, подходят и к числам в другой системе счисления.

Целые числа представляются арабскими цифрами, перед их отрицательными значениями записывается знак «–», а перед положительными значениями может быть записан знак «+».

Действительные числа, в зависимости от способа представления, подразделяются на две группы: *действительные с фиксированной точкой* и *действительные с плавающей точкой*.

Представление *действительных с фиксированной точкой* состоит из целой и дробной частей. Целая часть размещается перед (в левой стороне) дробной частью и они разделяются между собой знаком «.», называемым десятичной точкой. Перед ними для указания положительности или отрицательности их значений записывается «+» или «–». Обе части представляются арабскими цифрами.

Представление действительных с плавающей точкой состоит из частей, называемых *мантиссой*, основой *системы счисления* и *порядка*. Значения мантиссы, основания системы счисления и порядка могут быть положительными или отрицательными. Для их обозначения перед значениями записываются «+» или «–». Порядок представляется как целое число, а мантисса – как действительное число с фиксированной точкой.

Если обозначить мантиссу через M, порядок через p, основу системы счиления через q, то действительные числа будут следующими:

$$M * q^p .$$

Чтобы понять сказанное в таблице I.2.2 расмотрены примеры действительных чисел с плавающей точкой.

Таблица I.2.2. Примеры действительных 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 число в форме с плавающей точкой может быть представлено многими записями. Например, одно и то же число 3.14 может иметь следующие записи:

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

Чтобы иметь единственную запись для представления действительного числа с плавающей точкой нужно нормализовать его. Для этого должно выполняться следующее условие:

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

где $|M|$ – абсолютная величина.

Например, действительные числа с плавающей точкой $13.64 * 10^2$ и $0.00617 * 10^{-5}$ в нормализованном виде будут такими:

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

Комплексные числа представляются в виде алгебраической суммы своих частей: первая (левая) слагаемая – действительная часть, второя (правая) слагаемая – комплексная часть. Действительная часть и комплексная часть комплексного числа записываются в виде действительных чисел. Чтобы их различить после комплексной части ставится строчная латинская буква «*i*» в качестве её признака.

Примеры I.2.2.

1. 3.14 – положительное действительное число с фиксированной точкой, целая часть 3, а дробная часть 14.

2. +5 – положительное целое число 5.

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.

6. 0 + 3i – положительное комплексное число, действительная часть 0, а мнимая часть 3.

7. -3.14 + 2i – отрицательное комплексное число, действительная часть -3.14,, а мнимая часть 2.

8. $1.7 * 10^{2}$ - 0.12i – положительное комплексное число, действительная часть $1.7 * 10^{2}$, а мнимая часть 0.12.

Задания I.2.2.

Определить действительные числа с плавающей точкой и комплексные числа:

1)40.23;

2)$1.21 * 10^{2} + 5i$;

3)$3.3 * 10^{-2}$.

Помощь:

1) Действительное число с фиксированной точкой состоит из целой и дробной частей.

2) Комплексное число имеет действительную часть и мнимую часть, в конце которой записывается строчная латинская буква i.

3) Действительное число с плавающей точкой имеет основание системы счисления, мантиссу и порядок.

Вопросы I.2.2.

1. Какие типы имеются у числовых данных?

2. На какие группы подразделяются действительные числа в зависимости от формы их представления?

3. Из каких частей состоят комплексные числа?

Тесты I.2.2.

1. Что нужно использовать при нормализации действительных чисел?

A) $\boldsymbol{q}^{+1} \leq |\boldsymbol{M}| < 1$;

B) $\boldsymbol{q}^{-1} = |\boldsymbol{M}| < 1$;

C) $\boldsymbol{q}^{-1} \leq |\boldsymbol{M}| < 1$;

D) $\boldsymbol{q}^{-1} > |\boldsymbol{M}| < 1$;

E) $\boldsymbol{q}^{-1} \leq |\boldsymbol{M}| < \infty$.

2. Что является для 0 + 3*i* действительной частью, а что мнимой частью?

A) Действительная часть *i*, мнимая часть 3;

B) Действительная часть 0, мнимая часть 3;

C) Действительная часть 3, мнимая часть i;

D) Действительная часть *i*, мнимая часть 0;

E) Действительная часть 0, мнимая часть *i*.

3. Какими цифрами обозначается любое целое число?

A) Латинскими цифрами;

B) Греческими цифрами;

C) Казахскими цифрами;

D) Русскими цифрами;

E) Арабскими цифрами.

I.2.3. Числовые операции и их свойства

Операции, определенные над числовыми типами, нам известны со школы как *арифметические операции*: *сложение*, *вычитание*, *умножение*, *деление*. Они обозначаются знаками «+», «-», «*», «/» соответственно. Применяя эти операции, можно построить числовые выражения. Обычно, для записи числовых выражений мы используем *инфиксную запись*, в которой знаки операций записываются между операндами (аргументами). Например, если для любых чисел a и b ставится в соответствии третье число, являющееся суммой этих чисел, то её мы запишем как $a+b$.

При вычислении значений числовых выражений нужно учитывать приоритеты (порядки выполнения) этих операций: сначала выполняются высокоприоритетные операции умножение и деление, затем – низкоприоритетные операции сложение и вычитание. Иногда для изменения порядка выполнения операций используются скобки. В одном выражении можно использовать несколько скобок и их можно записывать внутри друг друга при этом операция в самой внутренней скобке и в самой левой скобке выполняется первой.

Существуют и бесскобочные способы записи числовых выражений, называемые *префиксной записью* и *постфиксной записью*. В префиксной записи знаки операций записываются перед своими операндами, а в постфиксной записи знаки операции – после своих операндов. Например, числовое выражение в инфиксной записи $(x+3)*(y-2)$ записывается в префиксной записи как $* + x\ 3 - y\ 2$, а в постфиксной записи – как $x\ 3 + y\ 2 - *$.

Замечание I.2.3.

1. Правила (но не порядок) выполнения этих операций на различных типах будут разными, несмотря на то, что они обозначаются одинаково. Например, правило суммирования целых чисел не годится для сложения действительных чисел.

2. Префиксная запись и постфиксная запись арифметического выражения, в честь их автора польского математика Яна

Лукашевича, названы *прямой польской записью* и *обратной польской записью* соответственно.

Пусть a, b и c – любые числа и a^{-1} – обратное к a число. Тогда операции над числами «+» – *сложение* и «*» – *умножение* имеют свойства, показанные в таблице I.2.3.

Таблица I.2.3. Свойства числовых операций.

№	Аксиома	Описание
1	$a+b=b+a$	Закон коммутативности
2	$a*b=b*a$	
3	$a+(b+c)=(a+b)+c$	Закон ассоциативности
4	$a*(b*c)=(a*b)*c$	
5	$a*(b+c)=a*b+a*c$	Закон дистрибутивности
6	$(a+b)*c=a*c+b*c$	
7	$a+0=0+a=a$	Свойства сложения
8	$a+(-a)=(-a)+a=0$	
9	$a*1=1*a=a$	Свойства умножения
10	$a\neq 0, a*a^{-1}=a^{-1}*a=1?$	

Аксиома 8 гласит, что для каждого числа a существует противоположное к нему число $-a$, а аксиома 9 – для каждого ненулевого числа a найдется обратное ему число $a^{-1}=1/a$.

Примеры I.2.3.

1. 5* $(b+c)$ = 5*b +5* c;
2. 3+ (–3) = (–3) + 3= 0;
3. (4 + 8)/2+(5+ 3)*2 = 12/2+8*2= 6+16 = 22.

Задания I.2.3.

1. Найдите обратное число к числу 6;
2. Вычислить 2*a+3*b при a=5 и b=8;
3. Вычислить $(a+5)*(3*b)$ при $a=10$ и b=2.

Помощь:

1. $a * a^{-1} = a^{-1} * a = 1$;

2. Учитывать порядок выпонения операций;

3. Учитывать порядок выпонения операций и скобки.

Вопросы I.2.3.

1. Что такое префиксная запись?
2. Что такое инфиксная запись?
3. Что такое постфиксная запись?

Тесты I.2.3.

1. Найти закон коммутативности?

A) $a + b = b + a$;

B) $a + (b+c) = (a+b) + c$;

C) $a + 0 = 0 + a = a$;

D) $a*1 = 1*a = a$;

E) $a * (b+c) = a*b + a*c$.

2. Найти закон дистрибутивности?

A) $a + (b+c) = (a+b) + c$;

B) $a * (b+c) = a*b + a*c$;

C) $a + 0 = 0 + a = a$;

D) $a*1 = 1*a = a$;

E) $a + b = b + a$.

3. Найти закон ассоциативности?

A) $a * (b+c) = a*b + a*c$; B) $a + (b+c) = (a+b) + c$;

C) $a + (-a) = (-a) + a = 0$;

D) $a*1 = 1*a = a$;

E) $a + b = b + a$.

I.2.4. Символьный тип данных

Символьный тип образуется из множества цепочек знаков алфавита заданного языка общения. В информатике алфавиты состоят из группы: *букв*, *цифр* и *специальных знаков*.

Среди букв могут быть строчные (маленькие) и прописные (заглавные) буквы. Иногда в алфавит одного языка могут добавляться буквы алфавита другого языка. Например, в алфавите современного казахского языка дополнительно имеются буквы русского языка *ё*, *э*, *ю*, *я*, *ц*, *ч*, *щ*, *ь*, *ъ*.

В качестве цифр берутся не римские, а арабские цифры *0*,*1*,*2*,*3*,*4*,*5*,*6*,*7*,*8*,*9*.

К специальным знакам относятся *знаки используемых операций*, *знаки препинания*, *знаки группировки* (*скобки*), *знак пробела* и т.п.

Значения символьных данных заключаются в символьных скобках. В качестве символьной скобки применяется знак апострофа « ' ». Например, 'ABC', '2015', 'X+1 > 0', '(3,14)'.

Итак, значение символьных данных представляются цепочками букв, цифр или специальных знаков, заключенных в символьные скобки. Можно определить длину каждой цепочки: она равна количеству символов в этой цепочке. Длину обозначают между двух вертикальных линий « | ». Например, длины выше приведенных символьных величин представляются так:

|ABC| = 3, |2015| = 4, | X+1 > 0| = 5, (3,14)| = 6.

В множество символьных данных включается абстрактная величина «*пустая цепочка*». В составе пустой цепочки нет ни одного символа. Обычно пустая цепочка обозначается знаком «ε». Длина пустой цепочки равна нулю, т.е. | ε | = 0.

Замечание I.2.4.

Пробел не пустая цепочка, он считается реальным символом. Длина пробела равна единице, т.е. | | = 1.

Примеры I.2.4.

1) ‘ABCDE’ – цепочка, образованная из начальных букв латинского алфавита.

2) ‘X+Y = 100’ – цепочка, образованная из смешанных знаков.

3) ‘A1’– цепочка, образованная из латинской буквы и цифры.

Задания I.2.4.

Определить тип следующих цепочек:

1. ‘ABC’;
2. ‘2013’ ;
3. ‘X+Y > 1’.

Вопросы I.2.4.

1. Что означает длина символьных данных?
2. Чему равна длина символьной цепочки ‘ABCD’?
3. Чему равна длина пустой цепочки?

Тесты I.2.4.

1. Из каких групп состоят алфавиты языков общений, используемых в информатике?

A) Состоит из группы букв, цифр и специальных знаков.

B) Не состоит из группы букв, цифр и специальных знаков.

C) Состоит из группы цифр и специальных знаков.

D) Состоит только из группы специальных знаков.

E) Состоит только из группы букв и цифр.

2. Какая цепочка состоит из смешанных знаков?

A) ‘210’ ;

B) ‘ABC’ ;

C) ‘X+1 > 0’;

D) ‘( )’;

E) ‘96’.

3. Чему равна длина цепочки |ABCDFHTRY| ?

A) 0;

B) 1;

C) 10;

D) 9;

E) 11.

I.2.5. Символьные операции и их свойства

Все операции над символьными данными в основном подразделяются на две группы:

1) Операции *конструирования*, позволяющие из заданных двух символьных цепочек получить одну символьную цепочку;

2) Операции *деления*, позволяющие из заданной символьной цепочки удалить символ или часть этой цепочки в зависимости от определенного условия.

Самой простой операцией конструирования является операция *конкатенация* (*сцепление*), она обозначается знаком «·». Для цепочек X и У она определяется так: после сцепления справа к значению X значения У получится цепочка Z и записывается в виде X•Y=Z. Например, если значение X есть ‘ҚАЗАҚ’, а значение Y есть ‘СТАН’, то значение Z будет ‘ҚАЗАҚСТАН’. В некоторых языках общения знак этой операции не записывается.

К операции конструирования относится операция *дизъюнкция* (*выбор*), обозначаеиая знаком «|». Например, если A=‘АЛМАТЫ’, B=‘АСТАНА’, то C = A|B = ‘АЛМАТЫ’ | ‘АСТАНА’.

Еще одной операцией конструирования является операция *итерация*, она обозначается знаком «*». Эта операция производная (сложная), которая определяется через операции *конкатенация* и *дизъюнкция*. Например, для любых символьных данных *X* можно определить:

$$X^* = \varepsilon \mid X \mid X^2 \mid X^3 \mid \ldots \mid X^n \mid \ldots,$$

где $X^n = \underbrace{X \cdot X \cdot \ldots \cdot X}_{n}$.

Для правильного построения символьного выражения с помощью операций конструирования нужно знать порядок их выполнения. Установлен следующий порядок: первая – операция итерация «*», вторая – операция конкатенация «·», третья – операция дюзъюнкция «|». Иногда для указания порядка выполнения операций используются скобки. Например, у этих двух выражений *0|10** и (*0*|(*1*(*0**))) значения одниковы.

Самой простой операцией деления является операция удаление из заданной цепочки определенного количества символов, начиная с указанного места. Если имя операции деление обозначим через ***ДЕЛ***, то можно записать удаление символов из цепочки Х в количестве М, начиная с позиции К в виде следующей функции:

ДЕЛ (Х, К, М) = У,

где зачение цепочки У получается после удаления М символов из значения Х, начиная с позиции К, то можно писать

Теперь рассмотрим свойства символьных операций.

Пусть α, β и γ – произвольные непустые символьные данные и ε – пустая цепочка. Тогда свойства операций конструирования, определенных над символьными данными, «•» – *конкатенация*, «|» – *дизъюнкция* и «*» – *итерация* показываются в таблице I.2.5.

Таблица I.2.5. Свойства символьных операциии.

№	Аксиома	Описание
1	$\alpha\cdot\varepsilon = \varepsilon\cdot\alpha = \alpha$	Закон коммутативности для пустых цепочек
2	$\alpha \mid \beta = \beta \mid \alpha$	Закон коммутативности для непустых цепочек
3	$\alpha\cdot\beta \neq \beta\cdot\alpha$	Закон некоммутативности для непустых цепочек
4	$\alpha\cdot(\beta\cdot\gamma) = (\alpha\cdot\beta)\cdot\gamma$	Закон ассоциативности
5	$\alpha \mid(\beta \mid \gamma)=(\alpha \mid\beta) \mid \gamma$	Закон ассоциативности
6	$\alpha\cdot(\beta \mid \gamma)=\alpha\cdot\beta \mid \alpha\cdot\gamma$	Закон дистрибутивности
7	$(\alpha \mid\beta)\cdot\gamma=\alpha\cdot\gamma \mid \beta\cdot\gamma$	Закон дистрибутивности
8	$\alpha\mid\alpha = \alpha$	Закон редукции
9	$(\alpha^*)^*=\alpha^*$	Закон редукции
10	$\alpha^*= \varepsilon\mid\alpha^2\mid\alpha^3\|\ldots\mid\alpha^n\mid\ldots$	Закон итерации
11	$\alpha^+ =\alpha \mid\alpha^*$	Закон итерации

1. Цепочка α является *префиксом* (*началом*) цепочки β, если найдется цепочка ξ такая, что выполняется равенство $\beta = \alpha\xi$. Если через $\alpha \sqsubset \beta$ обозначить отношение «цепочка α является префиксом цепочки β», то его формальное определение можно записать как

$$\alpha \sqsubset \beta \rightleftharpoons \exists\xi(\beta = \alpha\xi).$$

2. Цепочка α является *суфиксом* (*концом*) цепочки β, если найдется цепочка ζ такая, что выполняется равенство $\beta = \zeta\alpha$. Если через $\beta \sqsupset \alpha$ обозначить отношение «цепочка α является суфиксом цепочки β», то его формальное определение можно записать как

$$\alpha \sqsupset \beta \rightleftharpoons \exists\zeta(\beta = \zeta\alpha).$$

3. Цепочка α является *подцепочкой* (*частью*) цепочки β, если найдутся цепочки ζ и ξ такие, что выполняется равенство $\beta = \zeta\alpha\xi$. Если через $\alpha \sqsubseteq \beta$ обозначить отношение «цепочка α является *подцепочкой* цепочки β», то его формальное определение можно записать как

$$\alpha \sqsubseteq \beta \rightleftharpoons \exists\zeta\exists\xi(\beta = \zeta\alpha\xi).$$

Примеры I.2.5.

1. Если значение X есть ‘АЛТЫНБЕК’, K=1, а M=5, то

ДЕЛ (X, K,M) = ***ДЕЛ*** (‘АЛТЫНБЕК’, 1,5) = ‘БЕК’

2. Если значение X есть ‘АЛА’, а значение Y есть ‘ТАУ’, то X•Y = ‘АЛАТАУ’ ≠ Y• X = ‘ТАУАЛА’.

3. Если значение X есть ‘ТАУГҮЛ’, K=4, а M=3, то

ДЕЛ (X, K,M) = ***ДЕЛ*** (‘ТАУГҮЛ’, 4,3) = ‘ТАУ’.

4. $\varepsilon \sqsubset abcd,\ a \sqsubset abcd,\ ab \sqsubset abcd,\ abc \sqsubset abcd,\ abcd \sqsubset abcd.$

5. $abcd \sqsupset \varepsilon,\ abcd \sqsupset d,\ abcd \sqsupset cd,\ abcd \sqsupset bcd,\ abcd \sqsupset abcd.$

Задания I.2.5.

1. Если A=’АЛА’, B= ’ТАУ’, то A·B=?,

2. Если X= ‘ЖЕРОРТА’, K=4 и M=4, то ***ДЕЛ*** (X,K,M)=?

3. Если X= ’БУРАБАЙ’, Y =’БУРА’, то X $\sqsupset$Y или Y $\sqsubset$X?

Помощь:

1. Нужно знать операции конструирования;

2. Нужно знать операции деления;

3. Нужно знать отношения над цепочками.

Вопросы I.2.5.

1. Сколько групп операци определены над символьными данными?

2. С помощью каких операций определяется операция итерация над символьными данными?

3. Что такая пустая цепочка?

Тесты I.2.5.

1. Если A=’ШЫМ’, B= ’КЕНТ’, то A·B=?

A) ШЫМ;

B) ШЫМКЕНТ;

C) КЕНТ;

D) КЕНТШЫМ;

E) ШЫН.

2. Если X= ‘ЖЕЗКАЗГАН’, K=4 ; и M= 6, то ДЕЛ (X,K,M)=?

A) ГАН;

B) КАЗ;

C) ЖЕЗКАЗ;

D) ЖЕЗ;

E) КАЗГАН.

3. Какому виду аксиомы относится $\alpha \mid \beta = \beta \mid \alpha$?

A) Некоммутативность для непустых цепочек;

B) Коммутативность для пустых цепочек;

C) Закон дистрибутивности для пустых цепочек;

D) Закон ассоциативности для пустых цепочек;

E) Коммутативность для непустых цепочек.

I.2.6. Логический тип данных

Логический тип состоит из логических значений. К логическим значениям относятся «*истина*» и «*ложь*».

Логические значения задаются в следующем виде:

1) «*истина*» или «*да*» или «*1*» или «+»;

2) «*ложь*» или «*нет*» или «*0*» или «–».

Эти значения являются результатами определенных условий. К таким условиям относятся утверждения, принимающие значения «*истина*» или «*ложь*», и вопросы, требующие ответы «да» или «нет». Примерами логических значений могут служить значения сообщений из строк 1, 2 и 3 в таблице I.1.2.

В различных литературных источниках вместо логического значения (логической константы) «***ложь***» и «***нет***» используют «***false***», «***non***» или «**0**», а вместо логического значения (логической константы) «***истина***» и «***да***» – «***true***», «***yes***» или «**1**». Поэтому в дальнейшем для удобства в качестве логических значений можно использовать только **0**, **1**. Тогда будем считать, что любые логические переменные принимают значения из множества {**0**, **1**}.

Самое простое условие образуется с помощью операции сравнения (отношения): «<» – «*меньше*»; «=» – «*равно*»; «>» – «*больше*»; «≤» – «*меньше или равно*»; «≥» – «*больше или равно*».

Для построения сложного условия нужно применять логические операции, определенные над логическими значениями. Виды таких операций и их свойства будут рассматриваться ниже в разделе «Логические операции и их свойства».

Примеры I.2.6.

1. Простое условие «2 *меньше* 5» имеет значение «истина».
2. Простое условие «1 равно 3» имеет значение «ложь».
3. Простое условие «А меньше или равно 7» будет иметь значение «истина» или «ложь» в зависимости от значения числовой переменной А.

Задания I.2.6.

Определить значение «*истина*» или «*ложь*»:

1. 15=20;

2. 2>12;

3. 7<14.

Помощь:

Логическим значением является результат определенного утверждения, принимающего значения «*истина*» или «*ложь*», или вопросы, требующие ответы «да» или «нет».

Вопросы I.2.6.

1. Значения, определяющие логический тип?
2. Как задаются логические значения?
3. Какое значение будет иметь утверждение «Неверно, что компьютер умнее человека»?

Тесты I.2.6.

1. Определите логическое значение 45=60?

A) Ложь;
B) Истина;
C) Приблизительно;
D) Равно;
E) Мнимо.

2. Определите логическое значение 17<48?

A) Истина;
B) Ложь;
C) Приблизительно;
D) Равно;
E) Мнимо.

3. Определите логическое значение -12≥0?

A) Истина;
B) Ложь;
C) Приблизительно;
D) Равно;
E) Мнимо.

I.2.7. Логические операции и их свойства

Логические операции разнообразны. Мы среди них рассмотрим самые простые. Это операция *нет*, *и*, *или*. Используя эти операции можно определить любую сложную логическую операцию, т.е. они позволяют построить любое сложное логическое выражение (условие).

В зависимости от языка общения *логическое значение*, *операции сравнения* (*отношения*) и *обозначения логических операций* могут быть разными. Например,

1) *Операции сравнения* (*отношения*):

«<» – «*меньше*»;

«=» – «*равно*»;

«>» – «*больше*»;

2) *Логические операции*:

«¬» – «*инверсия*» или «*нет*»;

«&» или «∧» – «*конъюнкция*» или «*и*»;

«|» или «∨» – «*дизъюнкция*» или «*или*».

В любом логическом выражении логические отношения должны выполняться раньше логических операций, а среди логических операций самой первой выполняется операция «*инверсия*», затем операция «*конъюнкция*», а в конце операция «*дизъюнкция*».

Значения (модели) логических операций можно определить с помощью таблиц истинности. Например, если заданы логические переменные A и B, то значения логических операций можно определить следующим образом:

1. *Инверсия* (*нет*):

A	**¬*A***
0	1
1	0

Если значение логической переменной ***A*** есть «*ложь*», то его отрицание будет «*истина*» и наоборот.

2. *Конъюнкция* (*и*):

A	***B***	***A∧B***
0	0	0
1	0	0
0	1	0
1	1	1

Значение результата будет «*истина*» тогда и только тогда, когда обе переменные А и В принимают значение «*истина*», в противном случае результат имеет значение «*ложь*».

3. *Дизъюнкция* (*или*):

A	***B***	***A∨B***
0	0	0
0	1	1
1	0	1
1	1	1

Значение результата будет «*ложь*» тогда и только тогда, когда обе переменные А и В принимают значение «*ложь*», в противном случае результат имеет значение «*истина*».

Перед обсуждением свойств логических операций рассмотрим операции сравнения, которые определяются над любыми типами данных. Операцию сравнения можно определить между количествами или измерениями данных и их различаемыми друг от друга характеристиками.

Операции сравнения имеют много видов. Но, несмотря на это, результаты всех операций сравнения можно представить с помощью логических значений «истина» и «ложь». Кроме того, у всех операций сравнения имеются общие свойства.

Пусть задана операция сравнения ***R***, определенная над данными *a*, *b* и *c*. Тогда операция ***R*** имеет следующие свойства:

1) *рефлексивность*, если для каждого a выполняется $a\boldsymbol{R}a$;

2) *транзитивность*, если для каждого a, b и c от выполнения $a\boldsymbol{R}b$ и $b\boldsymbol{R}c$ следует выполнение $a\boldsymbol{R}c$;

3) *симметричность*, если для каждого a и b от выполнения $a\boldsymbol{R}b$ следует выполнение $b\boldsymbol{R}a$.

Теперь вместо $\boldsymbol{R}$ будем использовать хорошо нам известные обозначения «<» – меньше, «=» – равно и «>» – больше и запишем следующие свойства:

1. Для любых данных a и b операция $a < b$, $a = b$ или $a > b$ будет только в одном сравнении:

2. Если $a>b$ и $b>c$, то $a>c$.

3. Если $a= b$ и $b= c$, то $a=c$.

4. Если $a< b$ и $b<c$, то $a<c$.

Пусть p, q, r – произвольные логические данные и *0* – ложь, *1* – истина. Свойства логических операций приведены в таблице I.2.7.

Таблица I.2.7. Свойства логических операций.

№	**Аксиома**	**Описание**
1	$p\wedge q \sim q\wedge p$	Закон коммутативности
2	$p\vee q \sim q\vee p$	Закон коммутативности
3	$p\wedge(q\wedge r) \sim (p\wedge q)\wedge r$	Закон ассоциативности
4	$p\vee(q\vee r) \sim (p\vee q)\vee r$	Закон ассоциативности
5	$p\wedge(q\vee r) \sim (p\wedge q)\vee(p\wedge r)$	Закон дистрибутивности
6	$p\vee(q\wedge r) \sim (p\vee q)\wedge(p\vee r)$	Закон дистрибутивности
7	$\neg(p\vee q) \sim \neg p\wedge\neg q$	Закон Де Моргана
8	$\neg(p\wedge q) \sim \neg p\vee\neg q$	Закон Де Моргана
9	$\neg(\neg p) \sim p$	Закон двойного отрицания
10	$p \sim p$	Закон идентичности

11	$p \vee \neg p \sim 1$	Закон исключения третьего
12	$p \wedge \neg p \sim 0$	Закон противоречия
13	$p \wedge p \sim p$	Свойство конъюнкции
14	$p \wedge 1 \sim p$	
15	$p \wedge 0 \sim 0$	
16	$p \wedge (p \vee q) \sim p$	
17	$p \vee p \sim p$	Свойство дизъюнкции
18	$p \vee 1 \sim 1$	
19	$p \vee 0 \sim p$	
20	$p \vee (p \wedge q) \sim p$	

Примеры I.2.7.

1. В выражении $A \vee \neg B \wedge C$ выполняется сначала $\neg B$, затем $\neg B \wedge C$, а в конце $A \vee \neg B \wedge C$.
2. В выражении $A = 0 \vee B>1$ выполняется сначала операция сравнения $A = 0$ и $B>1$, затем логическая операция $\vee$.
3. В выражении $\neg(A=0) \wedge B=1$ выполняется сначала операция сравнения $A=0$ и $B=1$, затем логические операции $\neg$, $\wedge$.

Задания I.2.7.

В этом выражении выполнить операции и определить его логическое значение.

1) $(1 < 2) \wedge 1=3 \vee 2 <5$;

2) $1>3 \vee 1<2$;

3) $1<3 \vee 5=7$;

Помощь:

Среди логических операций самой первой выполняется операция «инверсия», затем операция «конъюнкция», а в самом конце операция «дизъюнкция».

Вопросы I.2.7.

1. Какую логическую операцию обозначают знаки «&», «∧»?
2. Какие значения имеют пустые места в этой таблице?

A	B	A∨B
0	0	
0	1	
1	0	
1	1	

Тесты I.2.7.

1. Какое логическое значение (0 или 1) будет иметь логическое 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 $\neg G \wedge D$, а в конце $D \vee\neg F \wedge G$.

3. Какой из них закон Де Моргана?

A) $\neg(\neg p) \sim p$;
B) $p \sim p$;
C) $\neg(p\vee q) \sim \neg p\wedge\neg q$;
D) $p\wedge\neg p \sim 0$;
E) $p\vee\neg p \sim 1$.

I.3. Кодирование информации

I.3.1. Кодирование логической и символьной информации

В параграфе I.1.1 было показано представление сообщения дискретной информации через символы алфавита некоторого языка (нотации), что очень важно для приема, представления, хранения, обработки и передачи информации с помощью компьютера.

Следует отметить, что человек может различить символы по их начертаниям, а компьютер – только 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, логической, 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.

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.

Логическая информация, которая имеет только два значения «Ложь» и «Истинна», не зависимо от их нотации, в компьютере представляется как 0 и 1, соответственно.

Из параграфа 1.3.8 показано, что используя логические операции $\neg$, & и $\vee$, обозначающие слова "*не*", "*и*" и "*или*"", можно построить любое сложное логическое выражение.

В современных компьютерах все три логические операции $\neg$,

& и ∨ аппаратно реализованы с помощью базовых логических элементов компьютера.

Базовые логические элементы компьютера, с указанием их входа и выхода, показаны на рисунке 1.3.1.1:

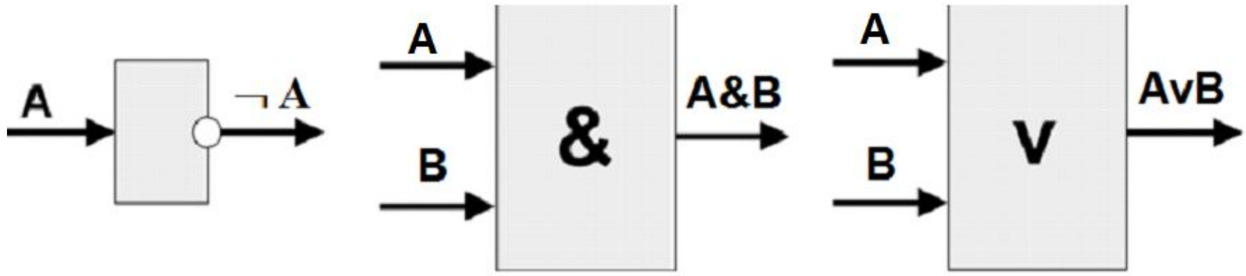


Рисунок 1.3.1.1. Базовые логические элементы.

При вводе в компьютер символьной информации происходит ее двоичное кодирование, код символа хранится в оперативной памяти, а при выводе символа на принтер или на экран компьютера происходит декодирование, т.е. преобразование кода символа в его изображение.

Код каждого символа задается значениями индексов в кодовой таблице 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. 2^8 = 256 символов или соответствующие им двоичный код от 00000000 до 11111111. Значения от 0 до 127 постоянны и формируют основную часть таблицы, куда входят десятичные цифры, буквы латинского алфавита (заглавные и строчные), знаки препинания (точка, запятая, скобки и др.), а также пробел и различные служебные символы (табуляция, перевод строки и др.). Значения от 128 до 255 формируют дополнительную часть таблицы, где принято кодировать символы национальных алфавитов. Чтобы обойти это ограничение, Международная организация по стандартизации (ISO - International Standards Organization) создала новый стандарт кодировки символов, названный Latin-1, который содержал символы европейских языков, не вошедшие в набор ASCII. Microsoft расширила Latin-1 и назвала этот стандарт ANSI. Но ANSI по-прежнему осталась 8-битной кодировкой. Многие языки имеют тысячи символов, особенно такие языки, как китайский, корейский и японский.

В Казахстане в 2002 году принят государственный стандарт кодировки букв казахского алфавита в 8-битовой кодовой таблице В таблице 1 показана 2-ая часть 8-битовой таблицы кодировки, где размещены коды букв казахского алфавита для Windows.

Таблица 1.3.1. 8-битовая таблица кодировки букв казахского алфавита для Windows.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8			‚		„	…	†			‰		<		Қ	Һ	
	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9		‘	’	“	”	•	–	—		™		>		қ	h	
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A		Ұ	ұ	Ә	¤	Ө		§		©	Ғ				®	Ү
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B		±	І	і	ө	µ	¶			№	ғ		ә	Њ	њ	ү
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	р	С	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Здесь в каждой ячейке указаны её десятичный номер и начертание символа, если он имеет 8-битовую кодировку, а в покрашенных ячейках размещены прописные и строчные буквы, которые обозначают специфические буквы казахского языка.

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 в результате в начале 90-х годов XX века был разработан стандарт кодирования символов, получивший название Unicode.

В Unicode для кодирования символов предоставляется 31 бит (4 байта за вычетом одного бита). Количество возможных комбинаций дает запредельное число: 2^{31} = 2 147 483 684 (т.е. более двух миллиардов). Поэтому Unicode описывает алфавиты всех известных языков, даже «мертвых» и выдуманных, включает многие математические и иные специальные символы, даже архаические символы, такие как древнетюркские руники, санскрит и египетские иероглифы, т.е. Unicode позволяет использовать в тексте почти любые языки и символы. В его таблице кодирования имеются коды применяющихся в различных отраслях науки знаков и различных декоративных знаков. Однако информационная емкость 31-битового Unicode все равно остается слишком большой. Поэтому чаще используется сокращенная 16-битовая версия (2^{16} = 65 536 значений), где первые 128 кодов совпадают с таблицей ASCII.

Первая 16-битовая 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.

Теперь рассмотрим алфавитный подход к измерению информации, в котором сообщением информации считается любая последовательность символов. Для определения количества такой информации подсчитывают длину её сообщения, без учета его значения (содержания).

Определение 1.3.1. Информационным объёмом сообщения называется количество двоичных цифр, которое используется для кодирования этого сообщения.

Пусть М – количество символов исходного алфавита, в котором записано сообщение, N – количество символов в записи сообщения. Тогда информационный объём сообщения вычисляется по формуле:

$$I = N \cdot log_2 M \qquad (1)$$

Если $log_2 M$ не является целым числом, то его нужно округлить в большую сторону или найти значение $log_2 \widetilde{M}$, где $\widetilde{M}$ – ближайшая целая степень 2 и $\widetilde{M} > M$.

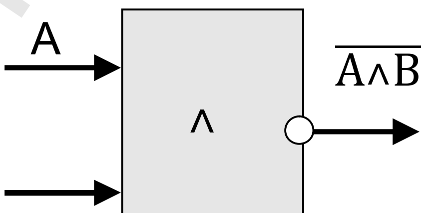
Информационный объём сообщения, выраженный в битах, и минимальное количество разрядов, необходимое для записи сообщения в двоичном алфавите, совпадают.

С помощью n двоичных разрядов можно закодировать двоичным кодом все элементы множества мощностью 2^n. Информационный объём одного символа алфавита, обозначающего элемент данного множества, равен n.

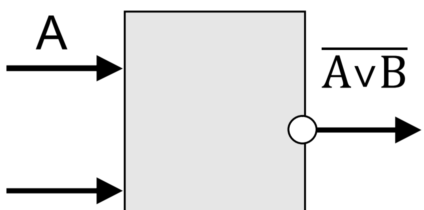
Пример 1.3.1.

1. С помощью этих базовых логических элементов **НЕ**, **И**, **ИЛИ** можно построить новые логические элементы:

И-НЕ



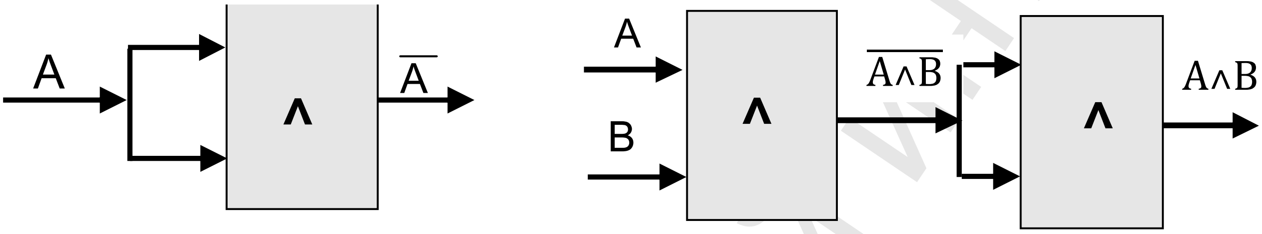
ИЛИ-НЕ



В В V

2. Любое логическое выражение можно реализовать на элементах **НЕ –И:**

НЕ: $\bar{A} = \bar{A} \vee \bar{A} = \overline{A \wedge A}$ И: $A \wedge B = \overline{\overline{A \wedge B}}$



3. Определить информационный объём слово «АСТАНА», если считать, что алфавит сообщения состоит из 10 букв.

Решение. Длина данного сообщения равна N = 6, мощность его алфавита равна 10, т.е. $|M|$ =10. По формуле (1) находим $I = 6 \cdot log_2 10$. Поскольку число 10 не равно целой степени числа 2, то значение $log_2 10$ округлим в большую сторону или найдем значение $log_2 \widetilde{M}$, где $\widetilde{M}$ – ближайшая целая степень числа 2 и $\widetilde{M} > 0$, что $\widetilde{M} = 16$. Тогда $I = 6 \cdot log_2 16 = 6 \cdot 4 = 24$. Ответ: 24.

Вопросы 1.3.1.

1. Как кодируется логическая информация?
2. Что понимается под логическим элементом?
3. Какие стандарты имеются для кодировки символов?
4. Как вычисляется информационный объем сообщения информации?

Задания 1.3.1.

1. Построить из логических элементов **НЕ –И** новую схему для реализации выражения $X = \bar{A} \& B \vee A \& \bar{B} \& \bar{C}$.

Помощь: реализация операции ∨ осуществляется параллельным соединением с одинаковыми входами, но разными выходами.

2. Сколько информации несёт в себе сообщение: «АСТАНА - СТОЛИЦА КАЗАХСТАНА!».

3. Определить максимальное количество страниц, содержащего по 80 символов в каждой строке и 64 строки на странице, которое может содержать файл объёмом 10 Кбайт в 8-битовой кодировке.

4. Два текста содержат одинаковое количество символов. Первый текст составлен в 8 символьном алфавите, а второй – 16. Определить во сколько раз отличается объём информации в них.

Помощь.

Обозначим за x количество символов в обоих текстах, тогда объем информации в первом тексте – $8x$, а во втором $16x$.

Тесты 1.3.1.

1.Определить минимальное количество битов для кодирования 16 слов в 6 символьном алфавите, если использовать 8-битовую кодировку.

A) 768;
B) *800*;
C) 1024;
D) 516;
E) 1540.

2. Определить размер алфавита, если объем составленного на нём сообщения составляет 1024 символов и занимает 1 Кбайт.

A) 256;
B) *512*;
C) 1024;
D) 2048;
E) 128.

3. Определить во сколько раз отличается объем информации в двух текстах, если они содержат одинаковое количество символов, но первый записан в 16-битовой, а второй – в 4-битовой кодировке.

A) 4;

B) *8*;

C) 24;

D) 32;

E) 2.

I.3.2. Системы счисления чисел

Вообще, *системой счисления* называется способ записи чисел с помощью цифр и множества правил. Есть несколько способов записи чисел с помощью цифр.

Любая система счисления должна удовлетворять следующие правила:

– наличие возможности записи значений чисел в заданном диапазоне (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 места (позиции) в записи числа. Если в записи числа одна и та же цифра встречается несколько раз, то она определяет разное значение. Например, запись числа арабскими цифрами: в трехзначном числе 333 самая левая цифра 3 определят 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 – единицу.

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

Возможности всех позиционных систем счисления одинаковы. Однако среди них десятичная система счисления является наиболее распространенной. Тому причина – в обеих руках человека имеются десять пальцев, что позволяет легко и удобно посчитать до десяти. Досчитав до десяти и исчерпав все возможности «вычислительного средства», разумно для следующей позиции (второго разряда) взять число 10 (десять) как новую единицу. Далее десятичное число десять будет единицей следующей позиции (третьего разряда) и так, продолжая вычисления, появилась десятичная система счисления.

Однако десятичная система счисления не сразу заняла преимущественное место в вычислениях. В разных исторических периодах многие народы использовали не десятичные системы счиления. Например, у древних тюрков основа системы счисления равно семи (1 неделя = 7 дней, 1 обхват = 7 пяди), у древних вавилонцев – шестьдесят (1 минута = 60 секунд, 1 час = 60 минут, 1 градус = 60 минут), у англичан – двенадцать (1 год = 12 месяцев, 1 фут = 12 дюйм, 1 шиллинг = 12 пенс).

В любой позиционной системе счисления с основой $\boldsymbol{q}$ заданное число $\boldsymbol{A}$ можно представить следующим образом:

$$A_{(q)} = a_{n-1}q^{n-1} + \cdots + a_1 q^1 + a_0 q^0 + a_{-1}q^{-1} + \cdots + a_{-m}q^{-m} \quad (1)$$

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

Среди этих систем счисления нам нужны *десятичная система счиления* и *двоичная система счисления*.

В таблице I.3.2 для каждой десятичной цифры дается двоичный эквивалент.

Таблица I.3.2. Двоичный эквивалент десятичных цифр.

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 можно заметить, что при записи одного и того же числового значения в различной системе счисления потребуется различное число знаков (разрядов). Например, двухзначное число 16 в десятичной системе счисления в двоичной системе счисления будет 10000, т.е. потребует пять знаков.

Для перевода заданного целого числа с основой ***p*** на основу ***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 равно 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.

При переводе смешанных чисел, необходимо в новую систему перевести отдельно целую и дробную части по правилам перевода целых чисел и правильных дробей, а затем оба результата объединить в одно смешанное число в новой системе счисления.

Перевод двоичных, восьмеричных и шестнадцатеричных чисел в десятичную систему счисления производится по правилу:

Для перевода числа Р-ичной системы в десятичную необходимо использовать следующую формулу разложения:

$$a_na_{n-1}\dots a_1a_0 = a_nP^n + a_{n-1}P^{n-1} + \dots + a_1P^1 + a_0P^0$$

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему счисления и обратно производится по правилу:

Для перевода числа из восьмеричной системы счисления в двоичную необходимо каждую цифру этого числа записать трехразрядным двоичным числом (триадой). При этом незначащие нули слева для целых чисел и справа для дробей не записываются.

Для обратного перевода двоичного числа в восьмеричную систему счисления, необходимо исходное число разбить на триады влево и вправо от запятой и представить каждую группу цифрой в восьмеричной системе счисления. Крайние неполные триады дополняют нулями.

Для перевода числа из шестнадцатеричной системы счисления в двоичную необходимо каждую цифру этого числа записать четырехразрядным двоичным числом (тетрадой). Примечание: незначащие нули слева для целых чисел и справа для дробей не записываются.

Для обратного перевода двоичного числа в шестнадцатеричную систему счисления, необходимо исходное число разбить на тетрады влево и вправо от запятой и представить каждую группу цифрой в шестнадцатеричной системе счисления. Крайние неполные триады дополняют нулями.

Замечания I.3.2.

1)Число, являющееся основой самой маленькой системы счисления равно двум, она называется двоичной системой счисления. Число в двоичной системе счисления состоит только из цифр 0 и 1.

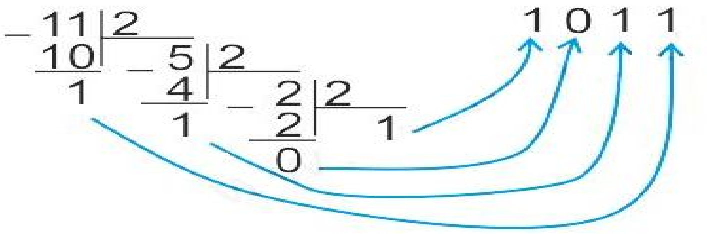
2)Самой распространенной системы счисления является десятичная система счисления, которая сотоит из десяти арабских цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

3)Правила выполнения операций в двоичной системе счисления и в десятичной системе счисления похожи.

4)Свойства операций над двоичными числами одинаковы со свойствами операций над десятичными числами.

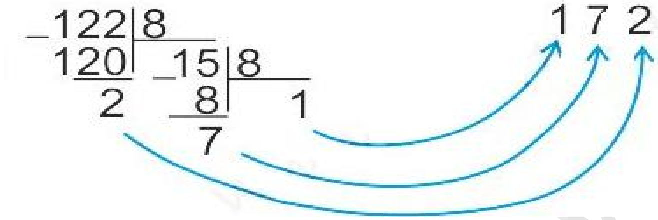
Примеры I.3.2:

1. Перевести число $11_{(10)}$ в двоичную систему счисления.



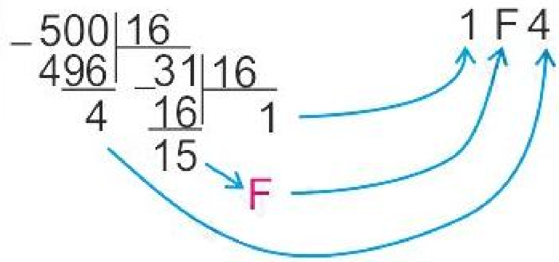
Ответ: $11_{(10)} = 1011_{(2)}$.

2. Перевести число $122_{(10)}$ в восьмеричную систему счисления



Ответ: $122_{(10)} = 172_{(8)}$.

3. Перевести число $500_{(10)}$ в шестнадцатеричную систему счисления.



Ответ: $500_{(10)} = 1F4_{(16)}$.

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

0,	625
	* 2
1,	250
	* 2
0,	500
	* 2
1,	000
	* 2
0,	000

Теперь, собирая сверху вниз, полученные единицы перехода на старшие разряды от результатов выполнения цепочки операций умножения дробной части заданного числа на основание переводимой системы счисления (в нашем случае 2), получим 1010, что означает $0.625_{(10)} = 0.1010_{(2)}$.

5. Четырехзначное десятичное число ***1952*** выражается так:

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

6. Десятичное число с трехзначной целой частью и трехзначной дробной частью $\mathbf{596.174_{(10)}}$ выражается так:

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

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

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

8. Перевести число $57{,}24_{(8)}$ в десятичную систему счисления.

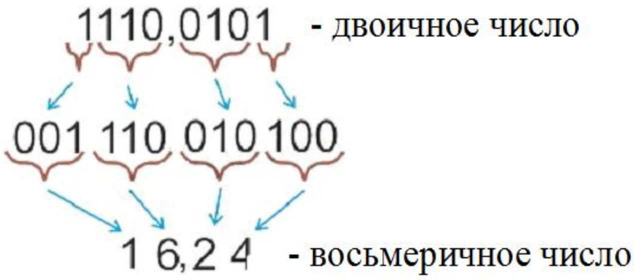
$$\overset{1}{5}\,\overset{0}{7},\overset{-1}{2}\,\overset{-2}{4}_{(8)} \to_{(10)} = 5*8^1+7*8^0+2*8^{-1}+4*8^{-2}=47{,}3125_{(10)}$$

9. Перевести число $7A{,}84_{(16)}$ в десятичную систему счисления.

$$\overset{1}{7}\,\overset{0}{A},\overset{-1}{8}\,\overset{-2}{4}_{(16)\to(10)} = 7*16^1+10*16^0+8*16^{-1}+4*16^{-2}=122{,}515625_{(10)}$$

10. Записать число $16{,}24_{(8)}$ в двоичной системе счисления.

11. Записать число $1110{,}0101_{(2)}$ в восьмеричной системе счисления.



12. Записать число $7A,7E_{(16)}$ в двоичной системе счисления.

13. Записать число $1111010,0111111_{(2)}$ в шестнадцатеричной системе.

Задания I.3.2.

1. Переведите заданное число 10000001 двоичной системы в десятичную счисления.
2. Переведите заданное число 129 десятичной системы в восьмеричную систему счисления.
3. Переведите заданное число 7A0 шестнадцатеричной системы в десятичную систему счисления.

Помощь:

1) Число $\boldsymbol{A}$ в позиционной системе с любой основой $\boldsymbol{q}$ можно выразить как:

$$A_{(q)} = a_{n-1}q^{n-1} + \cdots + a_1 q^1 + a_0 q^0 + a_{-1}q^{-1} + \cdots + a_{-m}q^{-m}$$

2) Для перевода заданного целого числа с основой ***p*** на основу ***q*** нужно делить несколько раз на ***q***, пока остаток не станет меньше, чем ***q***. Частное – старший разряд, остатки – остальные разряды.

3) Можно также использовать формулу

$$A_{(q)} = a_{n-1}q^{n-1} + \cdots + a_1 q^1 + a_0 q^0 + a_{-1}q^{-1} + \cdots + a_{-m}q^{-m}$$

Вопросы I.3.2.

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?

Тесты I.3.2.

1. Какая зависимость имеется сежду цифрами в позицонной системе счисления?

A) Значение цифр зависит от их записи;

B) Значение букв зависит от их места в записи;

C) Значение числа зависит от записи;

D) Место цифр зависит от их значения;

E) Значение цифр зависит от их места в записи.

2. Какая зависимость имеется в непозицонной системе счисления?

A) Значение цифр не зависит от их места в записи;

Б) Значение букв не зависит от их места в записи;

C) Значение числа не зависит от записи;

D) Не зависит от записи заданного числа;

E) Значение цифр зависит от их места в записи.

3. Что является основой позиционной двоичной системы счисления?

A) 2;0
B) 0 и 1;
C) 1 и 2;
D) 10;
E) 0.

II. МНОЖЕСТВА И ОТНОШЕНИЯ

II.1. Виды множеств и операции над ними

II.1.1. Понятие множества и операции над ними

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.

Когда мы говорим о множестве, то понимаем, что множество – это набор или совокупность элементов, объединенных каким–либо общим признаком (свойством). Элементами могут быть реальные физические или абстрактные математические объекты. На основании этого можно дать следующее интуитивное определение понятия множества:

Множеством называется объединение отдельных (дискретных) элементов, выбранных по некоторому признаку (критерию, типу).

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. При этом его элементы указываются в фигурных скобках { и }. 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}.

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

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

2) {1,2,3,4, ...} 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". $\{x \in Z: 2 \leq x \leq 6\}$ есть множество чисел от 2 до 6 включительно, Z – множество целых чисел;

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

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

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

Имена множеств обозначаются прописными (заглавными) латинскими буквами, а их элементы − строчными (малыми) латинскими буквами или арабскими цифрами. В обоих случаях можно использовать индексы, в том числе многоуровневые.

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

Среди всех множеств выделяют два особых множества:

1. $\varnothing$ – *пустое множество*, не содержащее ни одного элемента.

2. $\boldsymbol{U}$ – *универсальное множество* (*универсум*), содержащее все элементы рассматриваемого типа (предметной области).

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.

Мощность множества $\boldsymbol{A}$ равна числу его элементов и обозначается через $|\boldsymbol{A}|$.

Теперь можно определить операции над множествами:

Пусть заданы два множества $\boldsymbol{A}$ и $\boldsymbol{B}$, тогда над ними можно определить следующие операции:

1. *Объединение* множеств $\boldsymbol{A}$ и $\boldsymbol{B}$ состоит из элементов $\boldsymbol{A}$ или $\boldsymbol{B}$:

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

2. *Пересечение* множеств $\boldsymbol{A}$ и $\boldsymbol{B}$ состоит из элементов $\boldsymbol{A}$ и $\boldsymbol{B}$:

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

3. *Дополнение* к множеству $\boldsymbol{A}$ состоит из элементов универсума $\boldsymbol{U}$ и не включает элементов $\boldsymbol{A}$:

$$\overline{\boldsymbol{A}} = \{\ \boldsymbol{x}\ |\quad \boldsymbol{x} \in \boldsymbol{U}\ \ \&\ \ \boldsymbol{x} \notin\ \boldsymbol{A}\ \}.$$

4. *Разность* множеств $\boldsymbol{A}$ и $\boldsymbol{B}$ состоит из элементов множества $\boldsymbol{A}$ и не включает элементов $\boldsymbol{B}$:

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

5. *Симметрическая разность* множеств $\boldsymbol{A}$ и $\boldsymbol{B}$ состоит только из элементов $\boldsymbol{A}$ или только из элементов $\boldsymbol{B}$:

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

6. *Декартовое* (*прямое*) *произведение* множеств $\boldsymbol{A}$ и $\boldsymbol{B}$ состоит из всевозможных упорядоченных пар элементов $\boldsymbol{A}$ и $\boldsymbol{B}$:

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

Операции (1) – (3) могут быть представлены с помощью диаграммы Эйлера-Венна (Рис. I.2.1), в которой универсум $\boldsymbol{U}$

изображается прямоугольником, а множества $\boldsymbol{A}$ и $\boldsymbol{B}$ – окружностями. Для выделения результата применяется штриховка.

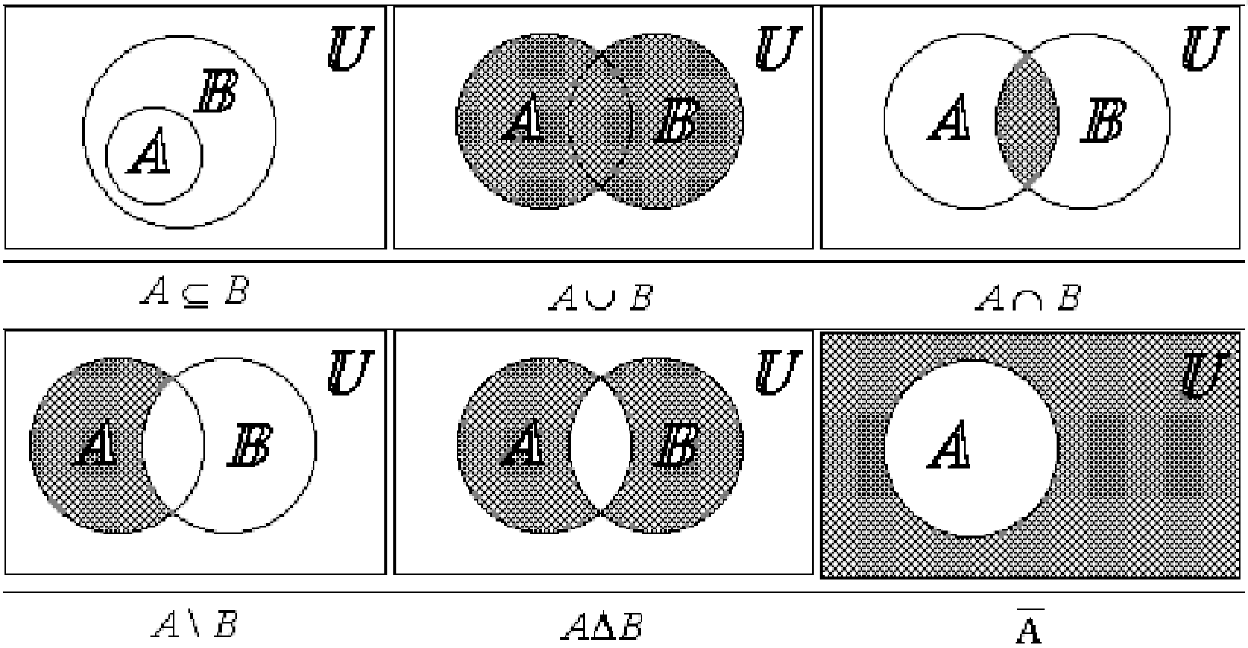


Рисунок I.2.1. Диаграммы Эйлера-Венна.

Здесь показано, что множества $\boldsymbol{A}$ и $\boldsymbol{B}$ являются подмножествами $\boldsymbol{U}$, и они записываются как $\boldsymbol{A} \subseteq \boldsymbol{U}$ и $\boldsymbol{B} \subseteq \boldsymbol{U}$ (см. *I*.2.2.).

Операции (1) – (3) можно определить не только над двумя множествами, но и над *n* множествами $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n$, где $n \in \boldsymbol{N}$ & $n>2$

Объединение над множествами $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n$ определяется как:

$$\boldsymbol{A}_1 \cup \boldsymbol{A}_2 \cup \ldots \cup \boldsymbol{A}_n = \bigcup_{i=1}^{n} \boldsymbol{A}_i .$$

Пересечение над множествами $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n$ определяется как:

$$\boldsymbol{A}_1 \cap \boldsymbol{A}_2 \cap \ldots \cap \boldsymbol{A}_n = \bigcap_{i=1}^{n} \boldsymbol{A}_i .$$

Прямое произведение над множествами $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n$ определяется как множество кортежей вида $(a_1, a_2, \ldots, a_n)$, $a_1 \in \boldsymbol{A}_1, a_2 \in \boldsymbol{A}_2, \ldots, a_n \in \boldsymbol{A}_n$, т.е.

$$\boldsymbol{A}_1 \times \boldsymbol{A}_2 \times \ldots \times \boldsymbol{A}_n = \{(a_1, a_2, \ldots, a_n) \colon a_1 \in \boldsymbol{A}_1 \,\&\, a_2 \in \boldsymbol{A}_2 \,\&\ldots\&\, a_n \in \boldsymbol{A}_n\}.$$

Здесь, если $\boldsymbol{A}_1 = \boldsymbol{A}_2 = \ldots = \boldsymbol{A}_n = \boldsymbol{A}$, то $\underbrace{\boldsymbol{A} \times \boldsymbol{A} \times \ldots \times \boldsymbol{A}}_{n} = \boldsymbol{A}^n$ – степень.

Теперь можно показать табличный способ задания множеств и операции над ними. Пусть заданы $\boldsymbol{U}$, $\boldsymbol{A} \subseteq \boldsymbol{U}$ и $x \in \boldsymbol{U}$.

Индикаторной (*характеристической*) функцией для множества $\boldsymbol{A}$ называется функция $\boldsymbol{I_A}(x)$, заданная как:

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

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

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

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

$$\boldsymbol{I_A(x)\le I_B(x)\Leftrightarrow A\subseteq B};$$

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

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

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

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

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

Индикаторы удобно задавать с помощью таблицы II.1.1.

Таблица II.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

Операции над множествами обладают следующими свойствами:

I. Объединение*, *пересечение и разность:

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

7) $\boldsymbol{A}\cap\boldsymbol{A} = \boldsymbol{A}$ – идемпотентность;

8) $\boldsymbol{A}\cap\boldsymbol{B} = \boldsymbol{A}$, если $\boldsymbol{A}$ все элементы $\boldsymbol{A}$ содержатся в $\boldsymbol{B}$;

9) $\boldsymbol{A}\cap\boldsymbol{B} = \boldsymbol{B}\cap\boldsymbol{A}$– коммутативность;

10) $(\boldsymbol{A}\cap\boldsymbol{B})\cap\boldsymbol{C} = \boldsymbol{A}\cap(\boldsymbol{B}\cap\boldsymbol{C}) = \boldsymbol{A}\cap\boldsymbol{B}\cap\boldsymbol{C}$ – ассоциативность;

11) $\boldsymbol{A}\cup(\boldsymbol{B}\cap\boldsymbol{C}) = (\boldsymbol{A}\cup\boldsymbol{B})\cap(\boldsymbol{A}\cup\boldsymbol{C})$ – дистрибутивность;

12) $\boldsymbol{A}\cap(\boldsymbol{B}\cup\boldsymbol{C}) = (\boldsymbol{A}\cap\boldsymbol{B}) \cup (\boldsymbol{A}\cap\boldsymbol{C})$ – дистрибутивность;

13) $\boldsymbol{A}\cap (\boldsymbol{B}\setminus\boldsymbol{C}) = (\boldsymbol{A}\cap\boldsymbol{B}) \setminus (\boldsymbol{A}\cap\boldsymbol{C})$ – дистрибутивность;

14) $\boldsymbol{A}\cup\overline{A} = \boldsymbol{U}$ – свойство дополнения;

15) $\boldsymbol{A}\cap\overline{A} = \varnothing$ – свойство дополнения;

16) $\overline{A \cup B} = \overline{A} \cap \overline{B}$ – закон де Моргана;

17) $\overline{A \cap B} = \overline{A} \cup \overline{B}$ – закон де Моргана;

18) $\overline{\overline{\mathbf{A}}} = \mathbf{A}$ – инволютивность;

19) $\boldsymbol{A}\setminus \varnothing = \boldsymbol{A}$ – свойство разности;

20) $\boldsymbol{A}\setminus\boldsymbol{A} = \varnothing$ – свойство разности;

21) $\boldsymbol{A}\setminus\boldsymbol{B} = \boldsymbol{A}\cap\overline{A} = \varnothing$ – свойство разности;

22) $\boldsymbol{B}\setminus\boldsymbol{A} = \boldsymbol{B}\cap\overline{A} = \boldsymbol{B}\setminus (\boldsymbol{B}\cap\boldsymbol{A})$ – свойство разности;

***II*. *Симметрическая разность и прямое произведение*:**

1) $\boldsymbol{A}\triangle\varnothing = \boldsymbol{A}$ – свойство нуля;

2) $\boldsymbol{A}\triangle\boldsymbol{A} = \varnothing$ – идемпотентность;

3) $\boldsymbol{A}\triangle\boldsymbol{B} = (\boldsymbol{A}\cup\boldsymbol{B}) \setminus (\boldsymbol{A}\cap\boldsymbol{B})$ – свойство симметрической разности;

4) $\boldsymbol{A}\triangle\boldsymbol{B} = \boldsymbol{B}\triangle\boldsymbol{A}$ – коммутативность;

5) $(\boldsymbol{A}\triangle\boldsymbol{B})\triangle\boldsymbol{C} = \boldsymbol{A}\triangle(\boldsymbol{B}\triangle\boldsymbol{C}) = \boldsymbol{A}\triangle\boldsymbol{B}\triangle\boldsymbol{C}$ – ассоциативность;

6) $(\boldsymbol{A}\cup\boldsymbol{B})\times\boldsymbol{C} = (\boldsymbol{A}\times\boldsymbol{C}) \cup (\boldsymbol{B}\times\boldsymbol{C})$ – дистрибутивность;

7) $\boldsymbol{A}\times(\boldsymbol{B}\cup\boldsymbol{C}) = (\boldsymbol{A}\times\boldsymbol{B}) \cup (\boldsymbol{A}\times\boldsymbol{C})$ – дистрибутивность;

8) $(\boldsymbol{A}\cap\boldsymbol{B})\times\boldsymbol{C} = (\boldsymbol{A}\times\boldsymbol{C}) \cap (\boldsymbol{B}\times\boldsymbol{C})$ – дистрибутивность;

9) $\boldsymbol{A}\times(\boldsymbol{B}\cap\boldsymbol{C}) = (\boldsymbol{A}\times\boldsymbol{B}) \cap (\boldsymbol{A}\times\boldsymbol{C})$ – дистрибутивность;

10) $(\boldsymbol{A}\setminus\boldsymbol{B})\times\boldsymbol{C} = (\boldsymbol{A}\times\boldsymbol{C}) \setminus (\boldsymbol{B}\times\boldsymbol{C})$ – дистрибутивность;

11) $\boldsymbol{A}\times(\boldsymbol{B}\setminus\boldsymbol{C}) = (\boldsymbol{A}\times\boldsymbol{B}) \setminus (\boldsymbol{A}\times\boldsymbol{C})$ – дистрибутивность;

Примеры II.1.1.

1. Все неотрицательные целые числа образуют множество натуральных чисел.

2. $\boldsymbol{D} = \{0,\ 1\}$, где элементами множества $\boldsymbol{D}$ являются только перечисленные постоянные величины 0, 1.

3. $\boldsymbol{X} = \{x\text{:}\ \ x>0\}$, где элементами множества $\boldsymbol{X}$ являются только положительные переменные величины x.

4. $|\varnothing| = 0$.

5. Если $\boldsymbol{A} = \{a, b, \text{c}, d, e\}$, то $|\boldsymbol{A}| = 5$.

6. Пусть $\boldsymbol{A} = \{a, b, c, d, e, f\}$, $\boldsymbol{B} = \{c, d\}$, то:
   - $\boldsymbol{B} \times \boldsymbol{B} = \{(c, c), (c, d), (d, c), (d, d)\}$;
   - $\boldsymbol{A} \setminus \boldsymbol{B} = \{a, b, e, f\}$;
   - $\boldsymbol{A} \triangle \boldsymbol{B} = \{a, b, e, f\}$;
   - $\overline{\boldsymbol{A}}$ зависит от того, какой будет универсум $\boldsymbol{U}$. Допустим, если $\boldsymbol{U} = \{a, b, c, d, e, f, h\}$, то $\overline{\boldsymbol{A}} = \{h\}$.

7. Пусть $\boldsymbol{A} = \{1,\ 2\}$, $\boldsymbol{B} = \{a,\ b\}$, $\boldsymbol{C} = \{+,\ -\}$. Тогда используя дистрибутивность $(\boldsymbol{A} \cup \boldsymbol{B}) \cup \boldsymbol{C} = \boldsymbol{A} \cup (\boldsymbol{B} \cup \boldsymbol{C}) = \boldsymbol{A} \cup \boldsymbol{B} \cup \boldsymbol{C}$ можно получить $(\{1,2\} \cup \{a,b\}) \cup \{+,-\} = \{1,2\} \cup (\{a,b\}) \cup \{+,-\}) = \{1,2\} \cup \{a,b\} \cup \{+,-\}$.

Задания II.1.1. Пусть $\boldsymbol{A} = \{1, 2, 4\}$, $\boldsymbol{B} = \{3, 4, 5, 6\}$. Тогда выполните следующие операции:

1) $\mathrm{A} \cup \mathrm{t}$;
2) $\mathrm{A} \cup \mathrm{A}$;
3) $\mathrm{A} \cap \mathrm{B}$;
4) $\mathrm{A} \times \mathrm{B}$;
5) $\mathrm{A} \setminus \mathrm{B}$;
6) $\mathrm{A} \cup$;
7) $\mathrm{A} \triangle \mathrm{B}$;
8) $(\mathrm{A} \cup \mathrm{B}) \times \mathrm{C}$;
9) $(\mathrm{A} \setminus \mathrm{B}) \times \mathrm{C}$.

Вопросы II.1.1.

1. Как задаются множества?
2. Что такое универсальное множество?
3. Как определяется подмножества?
4. Что такое диаграмма Эйлера-Венна?
5. Как определяется прямое произведение множеств?

6. Как определяется разность множеств?
7. Как определяется симметричная разность множеств?
8. Как определяется индикаторная функция множества?
9. Выяснить, какие из следующих дистрибутивных законов справедливы для любых множеств A, B, C:

1) $A\backslash(B\cup C) = (A\backslash B) \cup (A\backslash C)$;
2) $A\backslash(B\cap C) = (A\backslash B) \cap (A\backslash C)$;
3) $A\triangle(B\cup C) = (A\triangle B) \cup (A\triangle C)$;
4) $A\triangle(B\cup A) = (A\triangle B) \cup (A\triangle C)$;
5) $A\backslash(B\triangle A) = (A\backslash B) \triangle (A\backslash C)$;
6) $A\cup BC = (A\cup B)(A\cup C)$;
7) $A\cup(B\backslash C) = (A\cup B) \backslash (A\cup C)$;
8) $A(B\backslash C) = AB \backslash AC$;
9) $A\cup(B\triangle C) = (A\cup B) \triangle (A\cup C)$;
10) $A (B\triangle C) = AB \triangle AC$;
11) $A\triangle(B\backslash C) = (A\triangle B) \backslash (A\triangle C)$?

Тесты II.1.1.

1. Если $\boldsymbol{D} = \{d \mid d$ – целое число и $0 \leq d \leq 9$, то чему равна $|\boldsymbol{D}|$?

A) 10
B) 9
C) 10
D) 0
E) 1

2. Какой результат получится после выполнения операции $\boldsymbol{A}\cup\boldsymbol{B}$ для заданных множеств $\boldsymbol{A} = \{a, b, c\}$ и $\boldsymbol{B} = \{b, d, e, f\}$?

A) $\{a, b, c, d, e, f\}$
B) $\{a, b, c\}$
C) $\{b, d, e, f\}$
D) Ø
E) $\{c, b, d, e\}$

3. Какой результат получится после выполнения операции $\boldsymbol{A}\backslash\boldsymbol{B}$

для заданных множеств ***A*** ={а, б, в, г, д, е, ж} и ***B*** = {г, е, ж}?

A) {а, б, в, д}

B) {а, б, в, г, д, е}

C) {г, е, ж}

D) { г, е}

E) Ø

II.1.2. Отношения и способы их представления

Понятие «Отношение» играет очень важную роль во многих отраслях науки, особенно в информатике, так как оно используется при построении любого алгоритма, с помощью которого решается та или иная задача, для организации ветвления или повторения.

Определение II.1.2.1. Если заданы два множества $\boldsymbol{A}$ и $\boldsymbol{B}$ одного и того же типа, то можно вести следующие отношения:

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

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

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

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

Нетрудно заметить, что выше введенные отношения $=$, $\subseteq$ и $\subset$ являются подмножествами прямого произведения $\boldsymbol{A}\times\boldsymbol{B}$, т.е. можно считать, что любое отношение – это некоторое подмножество в прямом произведении, выделяемое определенным законом.

Заметим, что пустое множество $\varnothing$ является собственным подмножеством любого конечного множества.

Примеры II.1.2.1.

1) если $\boldsymbol{A} = \{a,b,c\}$, $\boldsymbol{B} = \{b,a,c\}$, то $\boldsymbol{A} = \boldsymbol{B}$;

2) если $\boldsymbol{A} = \{1,2,3,4\}$, $\boldsymbol{B} = \{3,1,4,2\}$, то $\boldsymbol{A} \subseteq \boldsymbol{B}$;

3) если $\boldsymbol{A} = \{1,2,3\}$, $\boldsymbol{B} = \{3,1,4,2\}$, то $\boldsymbol{A} \subset \boldsymbol{B}$

Определение II.1.2.2. Пусть $\boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_n$ − произвольные множества, не обязательно различные. Тогда *n-арным отношением* на множествах $\boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_n$ является подмножество

$$\boldsymbol{R}^n \subseteq \boldsymbol{A}_1 \times \boldsymbol{A}_2 \times \dots \times \boldsymbol{A}_n,$$

где $n \geq 1$, $\boldsymbol{R}^1$ – унарное отношение на $\boldsymbol{A}_1$, $\boldsymbol{R}^2$ – бинарное отношение на

$\boldsymbol{A}_1, \boldsymbol{A}_2$, $\boldsymbol{R}^3$ – тернарное отношение на $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3$ и т.д.

Всякое *унарное отношение* на множестве $\boldsymbol{A}$ является характеристическим свойством некоторого его подмножества. Множество всех унарных отношений на $\boldsymbol{A}$ совпадает с *множеством* всех подмножеств множества $\boldsymbol{A}$.

Примеры II.1.2.2.

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 II.1.2.3. *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 участников 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 II.1.2.3.

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

$\boldsymbol{R}ab$ – префиксная запись;

$a\boldsymbol{R}b$ – инфиксная запись;

$ab\boldsymbol{R}$ – постфиксная запись.

2. Если множества $\boldsymbol{A} = \{a_1, \ldots, a_r\}$ и $\boldsymbol{B} = \{b_1, \ldots, b_s\}$ находятся в бинарном отношении $\boldsymbol{R}$, то его можно представить с помощью таблицы I.2.2, в которой элементы a_i представляются строками, элементы b_j, – столбцами, а отношения $a_i\boldsymbol{R}b_j$ отмечается "1":

В таблице II.1.2 представлено двоичное (бинарное) отношение, определенное между двумя заданными множествами $\boldsymbol{A} = \{a_1, \ldots, a_r\}$ и $\boldsymbol{B} = \{b_1, \ldots, b_s\}$.

Таблица II.1.2. Двоичное отношение.

$\boldsymbol{R}$	b_1	b_2	…	b_{s-1}	b_s
a_1	1		…	1	
a_2		1	…		
…	…	…	…	…	…
a_{r-1}	1		…	1	
a_r			…		1

Определение II.1.2.4. Говорят, что бинарное отношение $\boldsymbol{R}$ на множестве $\boldsymbol{S}$:

4) *рефлексивно*, если для каждого $s \in \boldsymbol{S}$ имеет место $s\boldsymbol{R}s$;

5) *транзитивно*, если для любых $s,t,u \in \mathbf{S}$ из $s\boldsymbol{R}t$ и $t\boldsymbol{R}u$ следует $s\boldsymbol{R}u$;

6) *симметрично*, если для любых $s, t \in \boldsymbol{S}$ из $s\boldsymbol{R}t$ следует $t\boldsymbol{R}s$;

7) *антисимметрично*, если из $a\boldsymbol{R}b$ и $b\boldsymbol{R}a$ следует $a=b$.

Примеры II.1.2.4.

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

Определение II.1.2.5. Бинарное отношение $\boldsymbol{R}$ называется отношением *эквивалентности*, если оно удовлетворяет свойствам рефлексивности, транзитивности и симметричности.

Каждому отношению эквивалентности на множестве $\boldsymbol{S}$ соответствует единственное разбиение данного множества на смежные классы.

Примеры II.1.2.5. Отношение = в любом числовом множестве будет отношением эквивалентности, т.е. для любых k, m, $n \in N$:

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

Определение II.1.2.6. Бинарное отношение $\boldsymbol{R}$ на некотором множестве $\boldsymbol{S}$, удовлетворяющее свойствам рефлексивности, транзитивности и антисимметричности, называется отношением *частичного порядка*.

Примеры II.1.2.6. 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.

Определения II.1.2.7:

1. *Частично упорядоченным множеством* называется множество $\boldsymbol{A}$ с определенным на нем отношением частичного порядка. Точнее говоря, частично упорядоченным множеством называется пара $<\boldsymbol{A}, \boldsymbol{R}>$, где $\boldsymbol{A}$ – множество, а $\boldsymbol{R}$ – отношение частичного порядка на $\boldsymbol{A}$.

2. Элементы $\boldsymbol{a}$ и $\boldsymbol{b}$ частично упорядоченного множества $\boldsymbol{A}$ называются *сравнимыми* относительно частичного порядка $\boldsymbol{R}$ на этом множестве, если имеет место $\boldsymbol{aRb}$ или $\boldsymbol{bRa}$.

3. Частичный порядок на множестве **А** называется *линейным порядком*, если любые два элемента $\boldsymbol{a}$ и $\boldsymbol{b}$ из $\boldsymbol{A}$ сравнимы относительно *частичного* порядка $\boldsymbol{R}$.

4. *Линейно-упорядоченным множеством* или *цепью* называется частично-упорядоченное множество, в котором элементы каждой пары сравнимы.

5. Элементов $\boldsymbol{a}$ и $\boldsymbol{b}$ частично упорядоченного множества $\boldsymbol{A}$ называют *несравнимыми*, если между ними не выполнено ни одно отношение частичного порядка. Возможность существования

несравнимых элементов объясняет смысл термина *«частично упорядоченное множество»*.

Примеры 1.2.2.7.

1. Множество натуральных чисел $\boldsymbol{N}$ с отношением «$\leq$» является частично-упорядоченным, все натуральные числа будут сравнимыми относительно отношения «$\leq$», и $\boldsymbol{N}$ есть цепь.

2. Множество всех действительных чисел с отношением «=» является линейно упорядоченным множеством, если все действительные числа будут сравнимыми относительно отношения «=».

3. Пусть $\boldsymbol{A}$ – множество действительнозначных функций на отрезке $[0,1]$ с определенным на нем отношением частичного порядка $<, =, >$, то элементы $\boldsymbol{f}(x)=x$ и $\boldsymbol{g}(x)=1-x$ будут несравнимы.

Определения II.1.2.8. Пусть $\boldsymbol{A}$ – частично-упорядоченное множество, $\boldsymbol{B}$ – его подмножество, т.е. $\boldsymbol{A} \supseteq \boldsymbol{B}$. Тогда:

1) *нижней гранью* (*верхней гранью*) множества $\boldsymbol{B}$ во множестве $\boldsymbol{A}$ называется элемент $a\in\boldsymbol{A}$, такой, что $a\leq b$ ($b \leq a$) для любого $b\in\boldsymbol{B}$;

2) элемент $a\in\boldsymbol{A}$ называется *наименьшим* (*наибольшим*) во множестве $\boldsymbol{A}$, если a есть нижняя (верхняя) грань самого $\boldsymbol{A}$;

3) элемент $a\in\boldsymbol{A}$ называется *минимальным* (*максимальным*) во множестве $\boldsymbol{A}$, если не существует $b\in\boldsymbol{A}$, такого, что $b <a$ ($a <b$).

Наименьший (наибольший) элемент множества $\boldsymbol{A}$ является его единственным минимальным (максимальным) элементом.

Примеры II.1.2.8.

1. Множество всех подмножеств множества $\boldsymbol{A}$ имеет наименьший элемент $\varnothing$ и наибольший элемент само $\boldsymbol{A}$.

2. Множество $\boldsymbol{N}$ натуральных чисел имеет наименьший элемент 1 и не имеет наибольшего элемента.

3. Множество $\boldsymbol{Z}$ всех целых чисел не имеет наименьшего, наибольшего, минимального и максимального элемента.

Элементы множеств $\boldsymbol{A}$ и $\boldsymbol{B}$ находятся во *взаимно-однозначном соответствии*, если каждому элементу $a\in\boldsymbol{A}$ по некоторому закону сопоставлен один и тот же элемент $b\in\boldsymbol{B}$, причем каждый $b\in\boldsymbol{B}$

оказывается сопоставленным одному и тому же $a\in \boldsymbol{A}$.

Множества $\boldsymbol{A}$ и $\boldsymbol{B}$ являются *эквивалентными* (*равномощными*), если можно установить взаимно-однозначное соответствие между их элементами.

Замечание II.1.2.2. Бинарное отношение может задаться тройкой множеств $<\boldsymbol{R}, \boldsymbol{A}, \mathbf{B}>$, где $\boldsymbol{R}\subseteq \boldsymbol{A}\times\boldsymbol{B}$ – график отношения и записываться $(a, \underline{b}) \in \boldsymbol{R}$ или $a\boldsymbol{R}b$. Тогда можно определить:

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

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

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

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

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

Определение II.1.2.9. Бинарное отношение $\boldsymbol{f}\subseteq \boldsymbol{X}\times\boldsymbol{Y}$ называется *функцией* из $\boldsymbol{X}$ в $\boldsymbol{Y}$, если $\mathbf{Dom}\,\boldsymbol{R} = \boldsymbol{X}$ и $(x, y)\in \boldsymbol{f}$, $(x, z)\in \boldsymbol{f} \Rightarrow y=z$.

Функция $\boldsymbol{f}$: $\boldsymbol{X}\to\boldsymbol{Y}$ называется:

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

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

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

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

Примеры II.1.2.9. Пусть x, y, $z\in \boldsymbol{N}$ – натуральные числа и задана бинарная функция $\boldsymbol{f}(x,y)=z$, тогда:

1) если x есть 3, y есть 5, z есть 8, а $\boldsymbol{f}$ есть операция сложение +, то вместо записи $\boldsymbol{f}(x,y)=z$ можно писать 3+5=8 и с ней связать тернарное отношение $\boldsymbol{Add}(3, 5, 8)$, для которого справедливо 8 = 3+5.

2) если x есть 8, y есть 2, z есть 4, а $\boldsymbol{f}$ есть операция деление «:», то вместо записи $\boldsymbol{f}(x,y)=z$ можно писать 8:2 = 4 и с ней связать

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

Определение II.1.2.10. *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 $a\boldsymbol{R}b$, значит существуют x_1, x_2, …, x_n такие, что $a\boldsymbol{R}x_1$, $x_1\boldsymbol{R}x_2$, …, $x_n\boldsymbol{R}b$. Но из симметричности отношения ***R*** следует $b\boldsymbol{R}x_n$, $x_n\boldsymbol{R}x_{n-1}$, …, $x_1\boldsymbol{R}a$, следовательно, $b\boldsymbol{R}a$.

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

Примеры II.1.2.10.

1. Для любых a, b, $c \in \boldsymbol{N}$ из справедливости отношений $a<b$ и $b<c$ следует отношение $a<c$.

2. Для любых x, $y \in \boldsymbol{N}$ из справедливости отношения $x=y$ следует отношение $y= x$.

3. Если $\boldsymbol{A}$– множество городов, и на них задано отношение $x\boldsymbol{R}y$, означающее "существует автобусный маршрут из x в y", то транзитивным замыканием этого отношения будет отношение "существует возможность добраться автобусом из x в y".

Задания II.1.2. При $\boldsymbol{A}=\{a,b,c,d\}$, $\boldsymbol{B}=\{b,d\}$, $\boldsymbol{C}=\{c\}$ определить:

1. $\boldsymbol{A} \triangle \boldsymbol{B}$;
2. $\boldsymbol{A} \times (\boldsymbol{B} \cup \boldsymbol{C})$;
3. $\boldsymbol{A} \times (\boldsymbol{B} \cap \boldsymbol{C})$;
4. $\boldsymbol{A} \cup (\boldsymbol{B} \cap \boldsymbol{C})$.
5. $\boldsymbol{A} \setminus (\boldsymbol{B} \cup \boldsymbol{C})$;
6. $(\boldsymbol{A} \setminus \boldsymbol{B}) \cap \boldsymbol{C}$.
7. $(\boldsymbol{A} \triangle \boldsymbol{B}) \cap \boldsymbol{C}$;

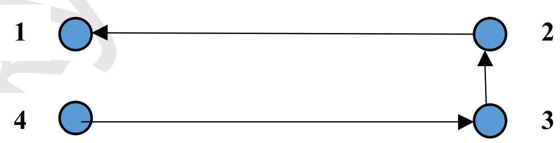
8. Выпишите упорядоченные пары, принадлежащие бинарным отношениям на множествах $A = \{1,3,5, 7\}$ и $B = \{2, 4, 6\}$:

(а) $U = \{\{x, y\}: x + y = 9\}$;

(б) $V = \{(x, y): x< \text{y}\}$.

9. Найдите все упорядоченные пары, принадлежащие отношению, которое определяется на множестве $A = \{1, 2, 3, 4, 5, 6\}$ следующим множеством $R = \{(x, y): x$ — делитель $y\}$.

10. Перечислите упорядоченные пары, принадлежащие R, если отношение R будет представлено как на нижнем рисунке.



10. Определите, какие из следующих отношений на множестве людей рефлексивны, симметричны или транзитивны:

(a) «...имеет тех же родителей, что и...»;

(b) «...является братом...»;

(c) «... старше или младше, чем...»;

(d) «... не выше, чем...».

(d) «... старше, чем сын...».

Вопросы II.1.2.

1. В чем заключается частичная упорядоченность множества?
2. Как определяется функция?
3. Как определяется композиция отношений?
4. Из чего состоит мощность объединения двух множеств?
5. Какие множества являются равномощными?
6. Какие элементы являются несравнимыми?
7. Что такое эквивалентность двух множеств?
8. Что такое транзитивное замыкание отношения?
9. Как устанавливается рефлексивность бинарного отношения?
10. Как устанавливается транзитивность бинарного отношения?
11. Как устанавливается симметричность бинарного отношения?
12. Как устанавливается антисимметричность бинарного отношения?

Тесты II.1.2.

1. Сколько множеств участвуют в бинарном отношении?

A) 2
B) 3
C) 4
D) 0
E) 1

2. Как соотносятся между собой отношение и прямое произведение?

A) Отношение является подмножеством прямого произведения.
B) Отношение включает прямое произведение.
C) Отношение и прямое произведение пересекаются.
D) Отношение является элементом прямого произведения.
E) Отношение описывает прямое произведение.

3. Какое бинарное отношение является отношением эквивалентности?

A) Если оно рефлексивное, транзитивное и симметричное;

B) Если оно рефлексивное и симметричное;

C) Если транзитивное и симметричное;

D) Если оно рефлексивное и транзитивное;

E) Если оно рефлексивное, не транзитивное и не симметричное;

II.1.3. Числовые множества и интервалы

Если элементами множества являются только числовые значения, то такие множества называются *числовыми множествами*. В зависимости от типа значений элементов мы будем различать числовые множества: *множество натуральных чисел*, *множество целых чисел*, *множество простых чисел*, *множество рациональных чисел*, *множество иррациональных чисел*, *множество действительных* (*вещественных*) *чисел*:

1. *Натуральные числа* являются самыми первыми числами, которыми люди начали пользоваться. Натуральные числа можно применить для счёта предметов и использовать в качестве их номеров. Множество натуральных чисел ограничено снизу и не ограничено сверху. Наименьшим натуральным числом является 1 - единица. Множество натуральных чисел обозначается буквой N, т.е. $N = \{1, 2, 3, ...\}$.

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

$$Z = \{..., -3, -2, -1, 0, 1, 2, 3, ...\} = \{0, \pm1, \pm2, \pm3, ...\}.$$

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. $Z^+ = N$.

3. *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. Простые числа широко используются в криптографии.

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:

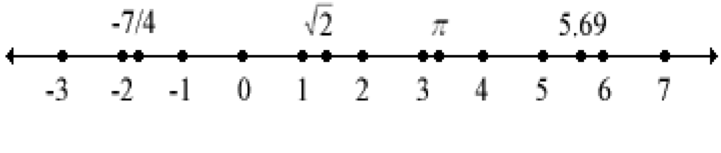
$$\pi = 3.1415926535\ \ ...,\ \sqrt{2} = 1.4142135623\ \ ...,\ \sqrt{7} = 2.6457513110\ \ ...$$

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*



Для каждого множества M можно построить новое множество, элементами которого являются все подмножества M и только они. Тогда множество M называют универсумом I, а множество всех его подмножеств – булеаном.

Если мощность универсума равна m, то мощность его булеана

$$\left|B(I)\right| = 2^m$$

Пример II.1.3.

1. Вложенность множества натуральных чисел N во множество целых чисел Z и вложенность последнего во множество действительных чисел R показаны на рисунке II.1.3.

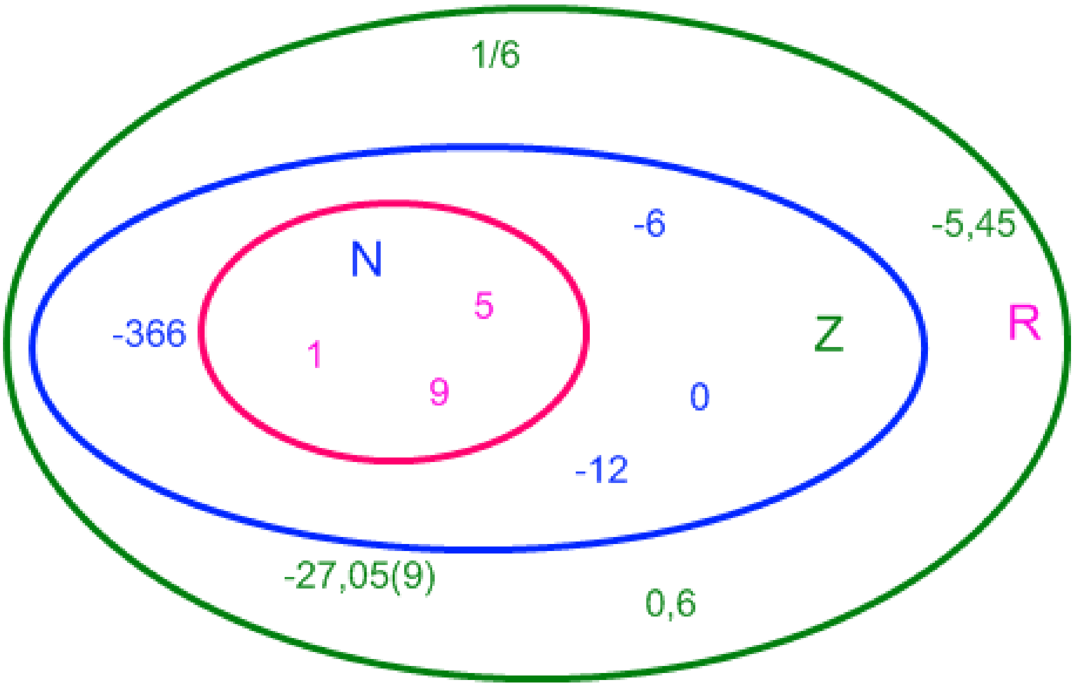


Рисунок II.1.3. Вложенность числовых множеств в друг друга.

2. Возьмем в качестве универсума I множество натуральных чисел на отрезке [1, 3], I={1, 2, 3}, тогда булеан

$B(I)$={ ,{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}}

Если элементы числового множества состоит из всех действительных чисел между данной парой чисел, то такое множество называется *интервалом*. Интервал может считаться 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, 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 в интервальной нотации как:

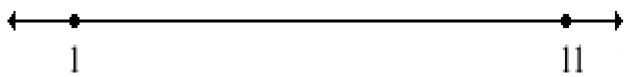
$$(3, 8) = \{x: 3<x<8\}.$$

В сегменте реальной числовой прямой, этот интервал был бы представлен линией так:

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 в интервальной нотации как:

$$[1, 11] = \{x: 1\leq x\leq 11\}.$$

В сегменте вещественно–числовой прямой, этот интервал представляется так:



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.

Таблица 7.1. Числовые интервалы.

Название	**Старое обозначение**	**Новое обозначение**	**Нотация в конструкторе множества**
Открытый интервал	(a,b)	$]a,b[$	$\{x\mid a<x<b\}$
Полу открытый (закрытый) интервал	$[a,b)$	$[a,b[$	$\{x\mid a\le x<b\}$
	$(a,b]$	$]a,b]$	$\{x\mid a<x\le b\}$
Закрытый интервал	$[a,b]$	$[a,b]$	$\{x\mid a\le x\le b\}$
Открытый числовой луч	$(a,+\infty)$	$]a,+\infty[$	$\{x\mid x>a\}$
	$(-\infty,b)$	$]-\infty,b[$	$\{x\mid x<b\}$
Закрытый числовой луч	$[a,+\infty)$	$[a,+\infty[$	$\{x\mid x\ge a\}$
	$(-\infty,b]$	$]-\infty,b]$	$\{x\mid x\le b\}$
Числовая ось	$(-\infty,+\infty)$	$]-\infty,+\infty[$	$\boldsymbol{R}$

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 $0 \le x \le 1$ 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:

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

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

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

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 стилей 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 неограниченная 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 –∞ и + ∞. В этой интерпретации, обозначения $[-\infty, b]$, $[-\infty, b)$, $[a, +\infty]$ и $(a, +\infty]$ все значимые и различны. В частности, (–∞, + ∞) обозначает множество всех простых действительных чисел, в то время как [–∞, + ∞] обозначает расширенные действительные.

Задание II.3.1.

1. Пусть U = {1, 2, 3, 4, 5}; A = {1, 3, 5}; B = {2, 4}; C = {2, 3, 4}; D ={5}. Вычислите $\overline{\mathrm{A} \cap \mathrm{B}}$; $(\mathrm{B} \setminus \mathrm{D}) \setminus (\mathrm{A} \cup \mathrm{C})$; $(\mathrm{U} \setminus \mathrm{B}) \cup \mathrm{D}$;

2. Пусть U = {1, 3, 4, 5, 7, 9}; A = {1, 3, 9}; B = {5, 7, 9}; C = {4, 5}; D ={9}. Вычислите $(\mathrm{U} \setminus D) \setminus \mathrm{C}$; $A \cap D$; $\overline{A \cap \overline{\overline{B}}}$; $\overline{\overline{D}} \cap \overline{\overline{B}}$;

3. Пусть U = {2, 4, 6, 8, 10}; A = {2, 4}; B = {4, 6, 8}; C={2, 6,

10}; D ={4}. Вычислите $A \cap \overline{D}$; $(A \setminus B) \cap (U \setminus D)$; $\overline{\overline{B} \cup C}$.

4. Пусть $A=\{1,2,4\}$, $B=\{3,4,5,6\}$, $C=\{5,7\}$ Вычислите $A\times C$; $A\cup\overline{A}$; $A\triangle B$; $(A\cap B)\times C$; $(A\setminus B)\times C$.

Вопросы II.3.1.

1. Если $A=\{1,2,3\}$, $B=\{3,4\}$, то что означает $\{1,2,3,4\}$?
2. Если $A=\{1,2,4\}$, $B=\{4,3,2\}$, то что означает $\{2,4\}$?
3. Если $A=\{1,2,3,4\}$, $B=\{3,4,5,6\}$, то что означает $\{2,4\}$?
4. Сколько элементов в каждом из множеств: $\{\varnothing,\{\varnothing\},\{\varnothing,\{\varnothing\}\}\}$.

Тесты *II.3.1*.

1.Определить $|D|$, если D состоит только из арабских цифр,

A) 10

B) 26

C) 28

D) 0

E) 30

2. Какой результат получится после выполнения операции A∩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) Ø

3. Какой результат получится после выполнения операции A∪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) Ø

E) {1, 3, 8}

III. ОСНОВЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ

III.1. Логические исчисления

III.1.1. Логика высказываний

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.

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, оно истинно – 1 или ложно - 0.

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.

Высказывания могут связываться с друг другом с помощью логических связей, которые приведены в таблице III.1.1.1.

Таблица III.1.1.1. Логические связки.

Логические связки	**Логические операции**	**Обозначение**	**Местность**
не верно, что	инверсия	$\neg$	унарная
или	дизъюнкция	$\vee$	бинарная
и	конъюнкция	$\wedge$	бинарная
если …, то …	импликация	$\to$	бинарная
тогда и только тогда, когда	эквивалентность	$\sim$	бинарная

Математическую логику мы будем изучать с помощью метаязыка, который отличается от предметного языка изучаемой логики. Если в качестве предметного языка можно использовать какой-нибудь естественный язык с сочетанием языка математики, то для определения метаязыка - языка логики высказываний нужно определить алфавит, синтаксис и семантику:

Алфавит языка логики высказываний.

Алфавит языка логики высказываний состоит из логических констант и счетного множества высказываний, обозначаемых строчными латинскими (пропозициональными) буквами с индексами или без них, а также из обозначения пяти логических операций, приведенных в таблице III.1.1.1 и круглых скобок для указания приоритета этих операций:

(1) 0, 1 – логические константы;

(2) $p, q, r, ...,$ – строчные латинские буквы с индексами или без них (пропозициональные буквы) служат для обозначения атомарных высказываний;

(2) $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\leftrightarrow$ – знаки логических операций для обозначения логических связок высказываний;

(3) (,) – круглые скобки для обозначения приоритета логических операций.

Синтаксис языка логики высказываний.

Синтаксис языка высказываний – правила, которые позволяют строить сложные высказывания из элементов алфавита, и индуктивно определяют понятия «формула» следующим образом:

1) ***Базис индукции:*** всякая пропозициональная буква, обозначающая некоторое атомарное высказывание, является формулой;

2) ***Индукционный шаг:*** A и B – формулы, то:

$\neg A$ – формула;

$A \wedge B$ – формула;

$A \vee B$ – формула;

$A \rightarrow B$ – формула;

$A \sim B$ – формула.

3) ***Индукционное ограничение:*** формула получается только с помощью правил, описанных в 1) – базис индукции и 2) – индукционный шаг.

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

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

Подформула – это часть формулы, сама являющаяся формулой.

Семантика языка логики высказываний.

Семантика языка высказываний – правила интерпретации формул, придающие определенные логические значения формулам.

Значение истинности формулы зависит только от структуры этой формулы и от значений истинности её составляющих, т.е. значение формулы есть функция от значений её составляющих.

Сначала в таблице I.2.3.2. зададим пропозициональные буквы и логические операции в области из двух элементов $\{0, 1\}$:

Таблица I.2.3.2. Таблица истинности для логических операций.

p	q	$\neg p$	$p\wedge q$	$p\vee q$	$p\to q$	$p\leftrightarrow q$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

Интерпретация формулы – это отображение ι, сопоставляющее каждому атомарному высказыванию этой формулы некоторое значение истинности. Интерпретацию ι, заданную на множестве атомарных высказываний, можно продолжить (распространить) на множестве формул (сложных высказываний) посредством индукции.

Значение формулы $F[p_1, ..., p_n]$ при данной интерпретации входящих в нее пропозициональных букв ι: $\{p_1, ..., p_n\} \Rightarrow \{0,1\}$ можно определить индукцией по построению формулы:

$F = p$: $F[\iota] = \iota(p)$;

$F = A{\wedge}B$: $F[\iota] = (A{\wedge}B)[\iota] = A[\iota] \wedge B[\iota]$;

$F = \neg A$: $F[\iota] = \neg A[\iota]$;

$F = A{\vee}B$: $F[\iota] = (A{\vee}B)[\iota] = A\,[\gamma] \vee B[\iota]$;

$F = A{\rightarrow}B$: $F[\iota] = (A{\rightarrow}B)[\iota] = A\,[\iota]{\rightarrow} B[\iota]$;

$F = A{\leftrightarrow}B$: $F[\iota] = (A{\leftrightarrow}B)[\iota] = A\,[\iota]{\leftrightarrow} B[\iota]$.

Интерпретация, при которой значение истинности формулы есть 1, называется моделью этой формулы.

Таким образом, логика высказываний выражает определенные функционально-истинностные связи между высказываниями, которые задаются таблицей истинности. Таблица истинности для ***n*** операндов имеет 2^n вариантов значений операндов, представленные строками таблицы. С каждой из этих строк связано свое значение истинности.

Пример III.1.1.

1. Пусть задана формула $(p{\wedge}q){\vee}r \rightarrow \neg(p{\vee}q)$. Нужно определить семантику этой формулы. Для этого для неё будет строиться таблица истинности, используя правила интерпретации.

Таблица истинностная формулы $(p{\wedge}q){\vee}r \rightarrow \neg(p{\vee}q)$: будет составлена из строк, задающих значения всех её подформул. Ниже приведена таблица истинности формулы $(p{\wedge}q){\vee}r \rightarrow \neg(p{\vee}q)$:

p	q	r	$p{\wedge}q$	$(p{\wedge}q){\vee}r$	$p{\vee}q$	$\neg(p{\vee}q)$	$(p{\wedge}q){\vee}r \rightarrow \neg(p{\vee}q)$
1	1	1	1	1	1	0	0
1	1	0	1	1	1	0	0
1	0	1	0	1	1	0	0
1	0	0	0	0	1	0	1
1	1	1	0	1	1	0	0
0	1	0	0	0	1	0	1
0	0	1	0	1	0	1	1
0	0	0	0	0	0	1	1

Примеры III.1.1. Примерами высказываний являются следующие повествовательные предложения;

1) Число три меньше, чем пять;

2) Не верно, что компьютер умнее человека;

3) Программист не работал или компьютер был сломан;

4) Он посещал занятия и успешно сдал экзамен;

5) Он говорит тогда и только тогда, когда услышит.

Задания III.1.1.

1. Сравните истинностные таблицы формул $\neg A \vee B$ и $A \to B$.
2. Найдите тавтологии среди следующих формул:
   a) $(A \& \neg A) \to (B \vee C \to (C \to \neg A))$;
   b) $(A \to \neg A) \sim \neg A$;
3. Вычислив только одну строку таблицы истинности, найдите формулы, не являющиеся тавтологиями:
   a) $A \vee B \to A \& B$;
   b) $(A \to B) \to (B \to A)$.
4. Найти значение формулы $(\neg B \to A)\ [\gamma]$, если имеются $(\mathrm{A} \vee B)\ [\gamma]=1$ и $(A \to B)\ [\gamma]=1$.

Вопросы III.1.1.

1. Будет ли тавтологией формула $(\neg A \vee B) \& (C \to (A \leftrightarrow B))$?
2. Существуют ли такие высказывания A, B и C, чтобы для них одновременно выполнялись условия в некоторой интерпретации γ $(A \to B)[\gamma] =$ И; $(B \vee C)[\gamma] =$ Л; $(B \leftrightarrow (A \& \neg C))[\gamma] =$ Л?
3. Эквивалентны ли следующие высказывания:
   $A \supset B$ и если A, то B?
   $A \supset B$ и коль скоро A, то B?
   $A \supset B$ и в случае A имеет место B?
   $A \supset B$ и A влечет B?
4. Из чего состоит алфавит логики высказаваний?
5. Как определяется подформула?
6. Как вычисляется значение формулы при данной интерпретации входящих в нее пропозициональных букв?
7. Как определяется семантика языка высказываний?

Тесты III.1.1.

1. 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.

2. Какой из них закон Де Моргана?

A) $\neg(\neg p) \sim p$

B) $p \sim p$

C) $\neg(p \vee q) \sim \neg p \wedge \neg q$

D) $p \wedge \neg p \sim 0$

E) $p \vee \neg p \sim 1$

3.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 $D \vee \neg F \wedge G$.

4. Что такое формула с тесными отрицаниями?

A) Формула, в которой $\neg$ 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.

III.1.2. Логика предикатов

Предикаты – это языковое выражение, обозначающее какое-то свойство или отношение.

Исчислением предикатов называют формальное исчисление, допускающее высказывание относительно переменных, фиксированных логических функций и предикатов.

Прежде чем рассмотреть метаязык исчисления предикатов, дадим определения понятия терма, логической функции и предиката.

Определим понятие логической функции как n-местную операцию на множестве $\{0,1\}$.

Алфавит:

(1) $x, y, \dots, x_1, x_2, \dots$ – 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, $t_1, t_2, \dots, t_n$ – термы, то $f^{(n)}(t_1, t_2, \dots, t_n)$ – терм.

Значение терма:

(1) если t – предметная переменная x, то $\mathrm{Val}\ t = \gamma(x)$;

(2) если $t = f^{(n)}(t_1, t_2, \dots, t_n)$, то $Val\, t = f^{(n)}(t_1, t_2, \dots, t_n)$

Функция:

$t = f^{(n)}(t_1, t_2, \dots, t_n)$ представима термом

$t = f^{(n)}(v_1, v_2, \dots, v_n)$, если

$\{v_1, v_2, \dots, v_n\} \subseteq \{x_1, x_2, \dots, x_n\}$ и $t[\gamma] = f^{(n)}[\gamma]$ для всех интерпретаций

$\gamma: \{x_1, x_2, \dots, x_n\} \Rightarrow \{0,1\}$

Предикатами будут логические функции $J^{(n)}(x_1, x_2, \dots, x_n)$, *которые заданы на непустой области D и принимают значения* 0, 1.

Предикат $J^{(n)}(x_1, x_2, \dots, x_n)$ становится высказыванием после означивания входящих в него переменных на элементах множества *D*.

***Алфавит*:**

(1) $x, y, \dots, x_1, x_2, \dots$ – предметные переменные;

(2) $P^{(n)}(x_1, x_2, \dots, x_n), \dots$ – предикатные буквы (*n*=0,1, ...);

(3) &, ∨, ¬, →, ↔, ∀, ∃ – логические связки и кванторы;

(4) (,) – вспомогательные символы.

***Формулы*:**

(1) $P^{(n)}(x_1, x_2, \dots, x_n)$, …– элементарные формулы или атомы;

(2) если A, B – формулы, то A&B, A∨B, ¬A, A→B, A↔B – формулы;

(3) если A(x) – формула со свободной переменной x, то ∀xA(x) и ∃xA(x) – формулы.

***Свободные и связанные переменные*.**

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

***Интерпретация формулы*.**

Значение формулы $E[P_1, P_2, \dots, P_m;\ x_1, x_2, \dots, x_n]$ при интерпретации предикатных букв $\tau: P^{(n)} \Rightarrow J^{(n)}$ и означивании $\gamma: \{x_1, x_2, \dots, x_n\} \Rightarrow D$ ($D \neq \varnothing$) предметных переменных, обозначается E[τ,γ], определим индукцией по построению формулы E :

1) $E = P^n(x_1, x_2, \dots, x_n)$, то E[τ,γ] = P[γ];

2) E = (A&B) $[P_1, P_2, \dots, P_m;\ x_1, x_2, \dots, x_n]$, то

$$E[\tau,\gamma] = A[\tau,\gamma]\ \&\ B[\tau,\gamma].$$

Аналогично для остальных логических связок.

3) $E = \forall x_1 A[P_1, P_2, \dots, P_m;\ x_1, x_2, \dots, x_n]$, то

$$E[\tau, \gamma] = \forall x_1 A[\tau, x_1, \gamma] = И],$$

где $\gamma: \{x_1, x_2, \dots, x_n\} \Rightarrow D$, если $A\ [\tau, a, \gamma] = И$ для любого $a \in D$.

4) $E = \exists x_1 A[P_1, P_2, \dots, P_m;\ x_1, x_2, \dots, x_n]$, то

$$E[\tau, \gamma] = \exists x_1 A[\tau, x_1, \gamma] = И],$$

где $\gamma: \{x_1, x_2, \dots, x_n\} \Rightarrow D$, если $A\ [\tau, a, \gamma] = 1$ для некоторого $a \in D$.

Формула $E[P_1, P_2, \dots, P_m;\ x_1, x_2, \dots, x_n]$ называется *общезначимой* или *тавтологией*, если для любой области $D \neq \varnothing$, для любых интерпретаций τ предикатных букв и любом означивании γ предметных переменных в области D, E[τ,γ] = И.

Примеры III.1.2.

1. 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.

2. Покажем, что формула P(*x*,*y*) → Q(*x*) не 1–общезначима, следовательно, не общезначима.

Помощь. 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_1	J_1	J_1
1	1	И	Л	И	Л

Истинностная таблица формулы P(*x*, *y*)→Q(*x*):

x	*y*	P(*x*,*y*)	Q(*x*)	P(*x*,*y*)→Q(*x*)
1	1	И	И	И
1	1	И	Л	Л
1	1	Л	И	И
1	1	Л	Л	И

3. Покажем, что формула ∀*x*(∃*x* P(*x*)→P(*x*)) не 2–общезначима.

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

x	J_1	J_2	J_3	J_4
1	И	И	Л	Л

2	И	Л	И	Л

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

x	P(x)	$\exists x$ P(x)	$\exists x$ P(x)→P(x)	$\forall x$($\exists x$P(x)→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	

4. Покажем, что формула $\forall x\ \exists y$ P(x, y)→$\exists y\ \forall x$ 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: $\exists y J_7(1,y)$~И, при x=2: $\exists y J_7(2,y)$~И, тогда $\forall x\ \exists y J_7(x,y)$=И.

При y=1: $\forall x\ J_7(x,1)$=Л, при y=2:$\forall x J_7(x,2)$=Л, тогда $\exists y \forall x J_7(x,y)$=Л.

Откуда $\forall x\ \exists y J_7(x,y) \rightarrow \exists y \forall x J_7(x,y)$ = Л.

Задания III.1.2. Будут ли следующие выражения формулами, и если это формулы, то какие переменные в них являются свободными, а какие связанными:

1) $\forall x_1 \exists x_2 \forall x_3 P(x_1, x_2, x_3, x_4)$;

2) $\forall x_1 P(x_1, x_2) \supset \exists x_2 P(x_1, x_2)$;

3) $\exists x_1 \exists x_2 P(x_1, x_2) \wedge Q(x_1, x_2)$.

2. Пусть $\boldsymbol{M} = < Z^+, f >$, где Z^+ – множество неотрицательный целых чисел, f – соответствие, которое для предикатных символов $S^{(3)}(x, y, z)$, $P^{(3)}(x, y, z)$ определяет следующие предикаты:

$S^{(3)}(x, y, z) = 1 \Leftrightarrow x + y = z$; $P^{(3)}(x, y, z) = 1 \Leftrightarrow x \cdot y = z$.

Записать в модели $\boldsymbol{M}$ формулы, выражающие следующие утверждения: 1) $x = 0$; 2) $x = 1$; 3) $x = 2$; 4) x – четное число; 5) x – нечетное число; 6) $x = y$; 7) $x \leq y$; 8) $x < y$; 9) x делит y; 10) коммутативность сложения; 11) ассоциативность сложения; 12) коммутативность умножения; 13) ассоциативность умножения; 14) дистрибутивность сложения относительно умножения;

К достоинствам исчисления предикатов можно отнести:

- методы достаточно хорошо изучены и формально обоснованы;

- в базах знаний можно хранить лишь множество аксиом, а все остальные знания получать из них по правилам вывода.

К недостаткам исчисления предикатов относятся:

- требуется адекватное описание предметной области;

- невозможность применения в качестве термов предикатов других предикатов.

Вопросы III.1.2.

1. Чем сходны и чем различаются классические исчисления логики предикатов и логики высказываний?

2. В какой интерпретации утверждение примет значение истинно?

   1) Любое простое число больше нуля;

   2) Существует планета, которая вращается вокруг земли;

   3) Существует время года, которое холоднее, чем лето.

3. Как с помощью предикатов написать предложения?

1) Счастливые часов не наблюдают;

2) Только дурак может быть счастлив;

3) Есть на свете счастливые люди;

4) Какой-то дурак не наблюдает часов.

4. Записать с помощью предикатов и кванторов условие «Все животные смертны. Все люди есть животные». Вывод: Все люди смертны» и доказать оно справедливо.

Тесты III.1.2.

1. Предикат - это:

A) языковое выражение, обозначающее какое-то свойство или отношение.

B) термин, используемый в математической логике по отношению к той или иной логической системе, для указания того, что для данной логики.

C) основной раздел современной логики, в котором описываются выводы, учитывающие внутреннюю структуру высказываний.

D) результат некоторой реконструкции естественного языка.

E) Высказывание, которое принимает значение 0 или 1.

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

A) У всех кошек есть хвост;

B) Найдется целое число x, удовлетворяющее условию $x^2 = 2$;

C) Всякое высказывание есть отрицание самого себя;

D) Всякое высказывание есть логическое следствие самого себя;

E) То есть всякое высказывание есть соединение самого себя.

3. Исчислением предикатов является:

A) формальное исчисление, допускающее высказывание относительно переменных, фиксированных функций и предикатов.

B) формальная теория, основным объектом которой служит понятие логического высказывания.

C) термин, используемый в математической логике по отношению к той или иной логической системе, для указания того, что для данной логики справедливы все законы исчисления высказываний.

D) правило вывода в исчислении высказываний.

E) аксиоматическая система с использованием высказываний и предикатов.

III.2. Логические законы и выводы

III.2.1. Тавтологии и эквивалентность

Прежде чем, приступить к логическим рассуждениям мы должны освоить логические законы, позволяющие сравнить двух высказываний на эквивалентность, для того, чтобы, при необходимости, заменять одно другим. Так же, нам потребуется техника для обнаружения тавтологий, более мощная, чем таблица истинности. И, наконец, мы рассмотрим методы рассуждений, которые могут быть полезны для разрешения логических проблем, сформулированных на естественном языке.

Определение III.2.1.

1. Два высказывания называются *эквивалентными*, если они на одних и тех же состояниях своих переменных принимают одни и те же значения.
2. *Тавтология* – высказывание, значение которого истинно при всех интерпретациях входящих в нее пропозициональных букв и обозначается знаком $\models$.
3. *Противоречие* – высказывание, значение которого ложно при всех интерпретациях входящих в нее пропозициональных букв.
4. Два высказывания называются эквивалентными, если они на одних и тех же состояниях своих переменных принимают одни и те же значения.

Таким образом, один способ установить эквивалентность двух высказываний – вычислить их таблицы истинности и сравнить. Мы, однако, воспользуемся другим способом.

Два высказывания A и B – эквивалентны тогда и только тогда, когда $A \sim B$ – тавтология (общезначимо).

Ниже приводится список основных тавтологий в логике высказываний:

1а. $\models A\to(B\to A)$

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

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

3а. $\models A\&B\to A$

$3б.\ \models A\&B \to B$

$4а.\ \models A \to A \vee B$

$4б.\ \models B \to A \vee B$

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

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

$7.\ \models \neg\neg A \to A$

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

$9а.\ \models (A \leftrightarrow B) \to (A \to B)$

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

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

Для упрощения выражений, мы определим, что означает для двух выражений быть эквивалентными и заменим более сложное высказывание на менее сложное.

Основные, часто используемые свойства эквивалентности:

***I*. Коммутативность**

1. $A \wedge B \sim B \wedge A$
2. $A \vee B \sim B \vee A$

***II*. Ассоциативность**

1. $A \wedge (B \wedge C) \sim (A \wedge B) \wedge C$
2. $A \vee (B \vee C) \sim (A \vee B) \vee C$

***III*. Дистрибутивность**

1. $A \wedge (B \vee C) \sim (A \wedge B) \vee (A \wedge C)$
2. $A \vee (B \wedge C) \sim (A \vee B) \wedge (A \vee C)$

***IV*. Закон Де Моргана**

1. $\neg(A \vee B) \sim \neg A \wedge \neg B$
2. $\neg(A \wedge B) \sim \neg A \vee \neg B$

***V*. Закон импликации**

1. $A \to B \sim \neg A \vee B$

***VI*. Закон прямого и обратного условий**

1. $A \sim B \sim (A \to B) \wedge (B \to A)$

***VII*. Свойство отрицания**

1. $\neg(\neg A) \sim A$

***VIII*. Закон идентичности**

$1.\ A \sim A$

***IX*. Закон исключения третьего**

$1.\ A \vee \neg A \sim И$

***X*. Закон противоречия**

$1.\ A \wedge \neg A \sim Л$

***XI*. Свойства дизъюнкции**

$1.\ A \vee A \sim A$

$2.\ A \vee T \sim И$

$3.\ A \vee Л \sim A$

$4.\ A \vee (A \wedge B) \sim A$

***XII*. Свойства конъюнкции**

$1.\ A \wedge A \sim A$

$2.\ A \wedge T \sim A$

$3.\ A \wedge Л \sim Л$

$4.\ A \wedge (A \vee B) \sim A$

***XIII*. Законы поглощения**

$1.\ A \& A \sim A,\ A \vee A \sim A$

$2. A \& (A \vee B) \sim A, A \vee (A \& B) \sim A$

$3.\ A \& И \sim A, A \& Л \sim Л, A \vee 1 \sim 1, A \vee Л \sim A.$

$4.\ A \rightarrow (B \rightarrow C) \sim A \& B \rightarrow C.$

$5.\ (A \rightarrow B) \& (B \rightarrow A) \sim (A \leftrightarrow B)$

***XIV*. Закон контрапозиции**

$1.\ (A \rightarrow B) \sim (\neg B \rightarrow \neg A)$

***XV*. Закон тождества**

$1. A \rightarrow A$

Мы будем использовать эти свойства в разных целях. Коммутативность, на.пример, позволяет нам менять местами элементы высказывания, в целях его упрощения. Ассоциативность позволяет снимать скобки. Дистрибутивность позволяет собирать подобные члены, подобно тому, как мы это делаем в арифметическом выражении. Закон импликации позволяет уходить от операции →, используя только операции ¬, ∨, ∧. Для того, чтобы убедиться в правильности этих свойств, достаточно построить их таблицы истинности.

Используя свойства эквивалентности, мы можем упростить высказывания.

Пример III.2.1.

1. Упростить выражение $(p \vee \neg q) \wedge r \wedge (\neg p \vee q)$. Применяем правила:

   I.1 получим $(p \vee \neg q) \wedge (\neg p \vee q) \wedge r;$

   I.2 получим $(\neg q \vee p) \wedge (\neg p \vee q) \wedge r;$

   V.1 получим $(q \rightarrow p) \wedge (p \rightarrow q) \wedge r;$

   VI.1 получим $(p \sim q) \wedge r;$

Окончательно получим $(p \vee \neg q) \wedge r \wedge (\neg p \vee q) \sim (p \sim q) \wedge r.$

2. Упростить выражение $p \vee (\neg q \rightarrow p) \vee \neg q$. Применяем правила:

   V.1 получим $p \vee (\neg(\neg q \vee p) \vee \neg q;$

   VII.1 получим $p \vee (q \vee p) \vee \neg q;$

   I.2 получим $p \vee (q \vee p) \vee \neg q;$

   I.2 получим $(p \vee p) \vee (q \vee \neg q)$;

   XI.1 получим $p \vee (q \vee \neg q)$;

   IX.1 получим $p \vee 1$;

   XI.2 получим 1.

Итого мы доказали, что $p \vee (\neg q \rightarrow p) \vee \neg q \sim 1$ – тавтология.

3. Упростить выражение $((p \rightarrow q) \rightarrow p) \rightarrow p$.

   V.1 получим $(\neg(p \rightarrow q) \vee p) \rightarrow p;$

   V.1 получим $(\neg(\neg p \vee q) \vee p) \rightarrow p;$

   IV.1 получим $((\neg(\neg p) \wedge \neg q) \vee p) \rightarrow p;$

   VII.1 получим $((p \wedge \neg q) \vee p) \rightarrow p;$

   I.2 получим $(p \vee (p \wedge \neg q)) \rightarrow p;$

   XI.4 получим $p \rightarrow p;$

   V.1 получим $\neg p \vee p;$

   I.2 получим $p \vee \neg p;$

   IX.1 получим 1.

В результате доказано, что $((p \rightarrow q) \rightarrow p) \rightarrow p$ – тавтология.

Задание III.2.1.

Пусть задано составное высказывание $(p \vee q) \wedge (p \vee \neg q)$. Доказать, что это составное высказывание эквивалентно с высказыванием p.

Помощь.

Нужно заполнить значениями в таблице истинности столбец для высказывания $(p \vee q) \wedge (p \vee \neg q)$.

Затем полученный столбец значений сравнить со столбцом для p. Если они будут совпадать, то можно заключить, что выражение $(p \vee q) \wedge (p \vee \neg q)$ эквивалентно p и обозначить это как $(p \vee q) \wedge (p \vee \neg q) \sim p$. Это означает, что везде, где мы встретим выражение $(p \vee q) \wedge (p \vee \neg q)$, можем его заменить на p.

Тесты III.2.1.

1. Будут ли эквивалентными следующие высказывания
   A) $A \wedge$B и A и B?
   B) $A \wedge$B и не только A, но и B?
   C) $A \wedge$B и B, хотя и A?
   D) $A \wedge$B и B, несмотря на A?
   E) $A \wedge$B и как A, так и B?

2. Будут ли эквивалентными следующие высказывания
   A) $A \vee$B и A или B?
   B) $A \vee$B и A или B?
   C) $A \vee$B и A, если не B?
   D) $A \vee$B и A и B?
   E) $A \vee$B и A или B?

3. Будут ли эквивалентными следующие высказывания
   A) $A \sim$B и A, если и только если B?
   B) $A \sim$B и если A, то B, и обратно?
   C) $A \sim$B и A, если B, и B, если A?
   D) $A \sim$B и A равносильно B?
   E) $A \sim$B и A тогда и только тогда, когда B?

III.2.2. Логические выводы

Доказательство на основе правил вывода.

Основной целью всякого рассуждения является установление истины в форме некоторого общезначимого утверждения, т.е. тавтологии. Для простых случаев, у нас есть метод таблиц истинности. Однако, он становится громоздким при числе переменных больше четырех.

Есть другой метод, называемый доказательством, который представляет собой последовательность логических выводов, правильность каждого из которых строго логически обоснован. Таким образом, рассуждение в этом методе принимает форму последовательности логических выводов.

Процесс доказательства в исчислении высказываний есть по существу последовательность преобразований высказывания p с целью показать, что p общезначимо и является развитием метода упрощения высказываний. Однако, доказательство включает важный дополнительный компонент: *вывод из предположения*. Вывод в доказательстве основан на небольшом числе *правил вывода*. Каждый шаг в доказательстве есть либо уже доказанное высказывание, либо высказывание, истинное по предположению и вводимое для последующих шагов. Каждый шаг, который является предположением, заключается в скобки [,]. Эти правила устанавливают, что одни высказывания могут следовать из других (аксиом), истинность которых либо уже была установлена, либо считаются таковыми по предположению. Все другие шаги должны быть доказаны. Последним шагом в доказательстве должно быть само высказывание p:

Докажем высказывание $p \Rightarrow p$

1. $[p]$
2. p
3. $p \Rightarrow p$ Правило I

Первым шагом мы делаем предположение, что p – общезначима. Тогда второй шаг непосредственно следует из первого. Раз мы предположили общезначимость p на первом шаге, то мы используем этот факт на втором. На третьем шаге мы используем правила вывода **I**, устанавливающее общезначимость высказывания $p \Rightarrow p$.

Доказательство с помощью правил вывода гибче, чем доказательство с помощью таблицы истинности. В первом случае мы можем проанализировать каждый шаг в цепочке доказательства. В то же время, неограниченный рост таблицы истинности не позволят нам этого сделать.

Присмотревшись внимательно к правилам вывода, можно увидеть, что они хорошо согласуются с нашей интуицией. Например, возьмём правило VIII. Если на предыдущих шагах была доказана общезначимость высказываний p и q, то очевидно, что высказывание $p \wedge q$ тоже будет общезначимым.

Итак, в дальнейшем при доказательстве мы будем использовать либо правила эквивалентности (в этом случае каждый шаг будет замещением правого вхождения в высказывании на левую часть правила) либо правила вывода.

I. Введение $\rightarrow$

$[\boldsymbol{p}] \qquad \dfrac{q}{p \Rightarrow q}$

II. Введение ~

$\boldsymbol{p \rightarrow q} \qquad \dfrac{q \Rightarrow p}{p \Leftrightarrow q}$

III. Удаление $\rightarrow$

1. $\boldsymbol{p} \qquad \dfrac{p \Rightarrow q}{q}$ (*Modus ponens*)

2. $\neg \boldsymbol{q} \qquad \dfrac{p \Rightarrow q}{\neg p}$ (*Modus tollens*)

IV. Удаление ~

1. $\dfrac{p \Leftrightarrow q}{p \Rightarrow q}$

2. $\dfrac{p \Leftrightarrow q}{q \Rightarrow p}$

V. Введение $\neg$

$[\boldsymbol{p}] \quad \dfrac{F}{\neg p}$

VI. Удаление $\neg$

1. $\boldsymbol{p} \qquad \dfrac{\neg p}{F}$

2. $\dfrac{F}{p}$

VII. Введение $\wedge$

1. $\boldsymbol{p} \quad \dfrac{q}{p \wedge q}$

VIII. Введение $\vee$

1. $\dfrac{p}{p \vee q}$

2. $\dfrac{q}{p \vee q}$

IX. Удаление $\wedge$

1. $\dfrac{p \wedge q}{p}$

2. $\dfrac{p \wedge q}{q}$

X. Удаление $\vee$

$[\boldsymbol{p}]\ [\boldsymbol{q}] \qquad \dfrac{p \vee q \quad r \quad r}{\boldsymbol{r}}$

Дедуктивный вывод.

Дедуктивный вывод является методом рассуждения, в котором частное заключение выводится из общего. Цепочка рассуждений, где высказывания связаны между собой логическими выводами.

Началом (посылками) дедукции являются аксиомы или просто гипотезы, имеющие характер общих утверждений («общее»), а концом — следствия из посылок, теоремы («частное»). Если посылки дедукции истинны, то истинны и её следствия. Дедукция противоположно индукции.

Используя дедуктивный вывод доказать $p \Rightarrow (q \Rightarrow p)$

1. $[\boldsymbol{p}]$ – Предположение

2. $[\boldsymbol{q}]$ – Предположение

3. p – 1.

4. $q \Rightarrow p$ – I, 2, 3

5. $p \Rightarrow (q \Rightarrow p)$ – I, 1, 4

Мы предположили общезначимость утверждений p, q и воспользовавшись правилом I. введение $\Rightarrow$.

Использование правила Modus Ponens.

Это правило хорошо работает, когда надо доказать высказывания типа “Если на выставке показывают картину “Джоконда - Gioconda”, то я куплю билеты.” Если кто-то сделал это утверждение, и вы увидели, что на выставке выставляется “Джоконда - Gioconda”, то вы можете заключить, что этот человек купил билеты.

Доказать $((p \Rightarrow q) \wedge (r \Rightarrow p) \wedge r) \Rightarrow q$

1. $[(p \Rightarrow q) \wedge (r \Rightarrow p) \wedge r]$ – Предположение
2. $r \Rightarrow p$ – IX. Удаление $\wedge$, 1
3. r – IX. Удаление $\wedge$, 1
4. p – III. Modus Ponens, 2, 3
5. $p \Rightarrow q$ – IX. Удаление $\wedge$, 1
6. q – III. Modus Ponens. 4, 5
7. $((p \Rightarrow q) \wedge (r \Rightarrow p) \wedge r) \Rightarrow q$ – I. Введение $\Rightarrow$, 1, 6

Использование Modus Tollens.

Доказать $((p \Rightarrow q) \wedge \neg q) \Rightarrow \neg p$

1. $[(p \Rightarrow q) \wedge \neg q]$ – Предположение
2. $p \Rightarrow q$ – IX. Удаление $\wedge$, 1
3. $\neg q$ – IX. Удаление $\wedge$, 1
4. $\neg p$ – III.2. Modus Tollens, 2, 3
5. $((p \Rightarrow q) \wedge \neg q) \Rightarrow \neg p$ – I. Введение $\Rightarrow$, 1, 4

Использование Введения $\neg$ и *Удаления* $\neg$.

Докажем $((p \Rightarrow q) \wedge \neg q) \Rightarrow \neg p$

1. $[(p \Rightarrow q) \wedge \neg q]$ – Предположение

2. $p \Rightarrow q$ – IX. Удаление $\wedge$, 1
3. $\neg q$ – IX. Удаление $\wedge$, 1
4. $[p]$ – Предположение
5. q – III. Modus Ponens, 4, 2
6. $\neg q$ – 3
7. 0 – VI. Удаление $\wedge$, 5, 6
8. $\neg p$ – V. Введение $\neg$ 4, 7
9. $((p \Rightarrow q) \wedge \neg q) \Rightarrow \neg p$ – I. Введение $\neg$ 1, 8

Доказательство от противного.

На использовании правила V. Введение $\neg$ основан часто используемый прием доказательства – доказательство от противного. Мы его уже использовали несколько раз. Его идея состоит в следующем.

Пусть мы хотим доказать общезначимость высказывания Q :"Треугольник со сторонами 2, 3, 4 – не прямоугольный."

Предположим, что $\neg Q$ – общезначимо, т.е. треугольник со сторонами 2, 3, 4 – прямоугольный. Тогда, используя теорему Пифагора, мы можем утверждать, что 4+9=16 , но $4+9 \neq 16$. Отсюда, используя правило VI.1 Удаление $\neg$, получаем 0. Имея 0 и предположение об общезначимости $\neg Q$, с помощью правила V, получаем общезначимость $\neg(\neg Q)$. Откуда, с использованием правила VII из параграфа III.2.1, получаем общезначимость Q.

Доказать $((p \Rightarrow q) \wedge p) \Rightarrow q$

1. $[\neg(((p \Rightarrow q) \wedge p) \Rightarrow q)]$ – Предположение
2. $\neg(\neg((p \Rightarrow q) \wedge p) \vee q)$ – Закон импликации V.1, 1
3. $\neg(\neg((p \Rightarrow q) \wedge p)) \wedge \neg q$ – Закон Де Моргана IV.1
4. $\neg \boldsymbol{q}$ – IX.2. Удаление $\wedge$, 3
5. $((p \Rightarrow q) \wedge p)$ – IX.1. Удаление $\wedge$, 3
6. $p \Rightarrow q$ – IX.1. Удаление $\wedge$, 5
7. p – IX.2. Удаление $\wedge$, 6
8. q – III.1. Modus Ponens, 6, 7

9. 0 – V.1 Удаление $\neg$, 4, 8

10. $((p \Rightarrow q) \wedge p) \Rightarrow q$ – V. Введение $\neg$, 1, 9

Принцип двойственности.

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

a) Если $\models \neg E$, то $\models E'$. b) Если $\models E$, то $\models \neg E'$.

c) Если $\models E \sim F$, то $\models E' \sim F'$. d) Если $\models E \to F$, то $\models F' \to E'$.

Логическое следствие.

Пусть заданы формулы $A_1, A_2, \dots, A_n$ и B. Если из одновременной истинности формул $A_1, A_2, \dots, A_n$ следует истинность формулы B, то формула B является логическим следствием формул $A_1, A_2, \dots, A_n$ и обозначается $A_1, A_2, \dots, A_n \models$ B, ($n \geq 1$), здесь $A_1, A_2, \dots, A_n$ – 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, $A \to$B (посылок) к формуле B (заключению):

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

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

Рассуждения с помощью предикатов.

Рассуждения с помощью предикатов аналогичны рассуждениям с помощью высказываний. Правила вывода, которые мы ввели в III.2.2 для высказываний, справедливы и для предикатов. Заменив все переменные их текущими значениями и произведя необходимые вычисления, мы получим высказывание, к которому применимы известные правила вывода. Однако, есть два особых случая, для которых у нас нет соответствующих правил вывода: это использование кванторов. В таблице III.2.2 собраны правила вывода для случая предикатов с кванторами.

Таблица III.2.2. Правила вывода для предикатов с кванторами.

Введение $\forall$	*Удаление* $\forall$
$\dfrac{R \Rightarrow P}{\forall i : R(i) : P(i)}$	$\dfrac{\forall\, i : R(i) : P(i)}{R(i_0) \Rightarrow P(i_0)}$
Введение $\exists$	*Удаление* $\exists$
$\dfrac{\forall i : R(i) : P(i)}{\neg\, \exists i : R(i) : \neg P(i)}$	$\dfrac{\exists i : R(i) : P(i)}{\neg\, \forall i : R(i) : \neg P(i)}$

Рассмотрим для примера правило введения $\forall$, которое гласит, что если $R \Rightarrow P$ выполняется на всех рассматриваемых состояниях, то мы можем утверждать, что $\forall i : R(i) : P(i)$ для всех значений i, где выполняется $R(i)$. Второе правило удаления $\forall$ такое, что если предикат $\forall i : R(i) : P(i)$ выполняется, то для произвольного i_0 выполняется $R(i_0) \Rightarrow P(i_0)$.

В качестве примера, демонстрирующего использование этих правил, рассмотрим обоснование корректности метода доказательства по индукции. Этот метод мы уже не раз использовали при доказательстве свойств алгоритмов. С ним мы еще много раз встретимся при обосновании правильности программ.

Для доказательства выполнимости предиката

$\forall i : R(i) : P(i)$, где $R(i)$ имеет форму $i \in \{1, 2, \ldots\}$

Рассмотрим два случая:

Начальный шаг: Докажем $P(1)$

Шаг индукции: Докажем $P(i) \Rightarrow P(i+1)$ при любом $i \geq 1$.

Если мы показали выполнимость этих шагов, то имеем

$$P(1)$$

$$P(1) \Rightarrow P(2)$$

$$P(2) \Rightarrow P(3) \text{ и т.д.}$$

Тогда из первых двух строк и правила *Modus Ponens* получаем выполнимость $P(2)$. Из $P(2)$ из третьей строки, применяя опять *Modus Ponens*, получаем $P(3)$ и т.д. до бесконечности.

Таким образом. $\forall i \in \{1, 2, ...\} : P(i)$.

Пример III.2.2.

В среду, когда случилось ограбление, либо Питс был в операционном зале банка, либо Елен в бухгалтерии банка. Питса никогда не видели в операционном зале без Ирвина. Ирвин покидал банк в среду только тогда, когда он с Елен ездил на встречу с клиентами. Если в ограблении участвовал Корн, Ирвина не было бы в банке. Ограбление произошло в среду. Мог ли Корн быть грабителем?

Обозначим:

p = Питс был в операционном зале;

q = Елен была в бухгалтерии;

s = Ирвин был в операционном зале;

h = Корн участвовал в ограблении;

u = Ограбление случилось в среду.

Тогда исходные утверждения можно записать так:

1. $u \rightarrow (p \vee q)$
2. $p \rightarrow s$
3. $\neg s \rightarrow \neg q$
4. $h \rightarrow \neg s$
5. u
6. Из 1, 5 п. Modus ponens получаем $p \vee q$
7. Предположим $[q]$
8. Из 3, 7 п. Modus Tollens получаем s
9. Из 7, 8 и “введение →“ получаем $q \rightarrow s$
10. Из 4, 10 п. Modus Tollens $\neg h$

Итак, Корн не мог участвовать в ограблении.

Задания III.2.2.

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

1) Я бы заплатил за работу по ремонту компьютера, только если бы он стал работать. Он же не работает. Поэтому я платить за ремонт 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: Ирвин не сделает эту работу, если ее сделает Питс. Питс и Сидоров сделают эту работу в том и только том случае, если ее сделает Ирвин. Сидоров, эту работу сделает, а Ирвин нет. Следовательно, Питс не сделает эту работу.

3. Пусть *M* – множество точек, прямых и плоскостей трехмерного пространства. Рассмотрим модель *D* = <*M*, *f*>, где *f* – соответствие, которое для предикатных символов $Q(x,y), R(x,y)$, $P_1(x), P_2(x), P_3(x)$ определяет предикаты:

$P_1(x) = 1 \Leftrightarrow$ «x– точка»,

$P_2(x) = 1 \Leftrightarrow$ «x– прямая»,

$P_3(x) = 1 \Leftrightarrow$ «x–плоскость»,

$Q(x,y) = 1 \Leftrightarrow$ «x лежит на y»,

$R(x,y) = 1 \Leftrightarrow$ «x совпадает с y».

Используя вышеопределенные предикаты напишите в этой модели следующие утверждения:

1) Через каждые две точки можно провести прямую и притом единственную, если эти две точки различны;

2) Через каждые три точки, не лежащие на одной прямой, можно провести единственную плоскость;

3) Определение параллельных прямых;

Вопросы III.2.2.

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

2. Выясните, какие из высказываний каждой пары являются отрицаниями друг друга?

A) в книге более 100 страниц;

B) в книге не более 100 страниц;

C) эта гвоздика красная;

D) эта гвоздика розовая;

E) это слово – существительное;

F) это слово – прилагательное.

3. Какие из следующих предложений являются составными высказываниями?

A) в 1 м 100 см или 10 дм;

B) 27 кратно 3 и меньше 3;

C) Неверно, что 45 четное число;

D) Сегодня понедельник;

E) Если треугольник равносторонний, то он равнобедренный;

4. Пусть заданы два высказывания на языке логики первого порядка. A - $\forall x\ \exists y\ (x \geq y)$ и B - $\exists y\ \forall x\ (x \geq y)$, $x \in N$, $y \in N$, N – множество натуральны чисел.

A) Как эти высказывания запишутся на естественном языке?

B) Является ли высказывание A истинным при этой интерпретации?

C) Является ли высказывание B истинным при этой интерпретации?

D) Является ли B логическим следствием A?

E) Является ли A логическим следствием B?

Тесты III.2.2.

1. Высказывания A и B являются эквивалентными, если:

A) тогда и только тогда, когда истинности высказываний A и B совпадают;

B) тогда и только тогда, когда высказывание A истинно, а B – ложно;

C) тогда и только тогда, когда высказывания A ложно и B – истинно;

D) тогда, когда конъюнкция высказываний A и B ложна;

E) тогда, когда дизъюнкция высказываний A и B истинно.

2. Выясните, в каких случаях умозаключения истинны:

A) если число натуральное, то оно целое; число 6 – целое; значит, оно натуральное;

B) если число нечетное, то оно не делится на 2; число 15 – нечетное; значит, число 15 не делится на 2;

C) если треугольник равнобедренный, то в нем имеются хотя бы две равные стороны; треугольник ABC – неравнобедренный; значит, в нем нет ни одной пары равных сторон;

D) если число делится на 3, то сумма цифр в записи этого числа делится на 3; число 32 не делится на 3; значит, сумма цифр в его записи не делится на 3;

E) если число четное, то оно делится на 3; число 13 – нечетное; значит, число 13 не делится на 2.

3. Выберите дедуктивные умозаключения:

A) все отличники I-курса спортсмены; студент II-курса Джон – отличник; следовательно, Джон – спортсмен;

B) не все отличники II-курса спортсмены; второкурсник Джек отличник; следовательно, он не спортсмен;

C) все отличники I-курса спортсмены; второкурсница Elen не отличница; следовательно, Elen не спортсменка;

D) все отличники I-курса спортсмены; второкурсник Коэн - спортсмен; следовательно, он отличник.

Е) не все отличники II-курса спортсмены; первокурсник Боб отличник; следовательно, он не спортсмен.

IV. ГРАФЫ

IV.1.Понятие и виды графа

IV.1.1. Элементы 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 $G = (V, E)$, где V – множество вершин, E – множество ребер, фактически есть отношение на множестве V, т.е. $E \subseteq V \times V$.

Две вершины, которые соединяет ребро графа, называются *граничными вершинами* этого ребра и являются *смежными*, т.е. $v_i \in V$ и $v_j \in V$ – граничные вершины $(i = \overline{1,n}; j = \overline{1,m})$, $(v_i, v_j) \in E$ – ребра.

Ребро, граничные вершины которого совпадают, называется *петлей*, т.е. $(v_i, v_i) \in E$ – петля. В петле есть единственная вершина, т.е. ребро исходит из одной вершины и напрямую входит в эту вершину обратно (см. Рисунок IV.1.1.1).

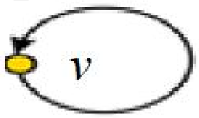


Рисунок IV.1.1.1 Петля.

Ребра с одинаковыми граничными вершинами называются *кратными*. Например, если у 3-х ребер одинаковые граничные вершины, то они будут 3-х кратными ребрами (см. Рисунок IV.1.1.2).

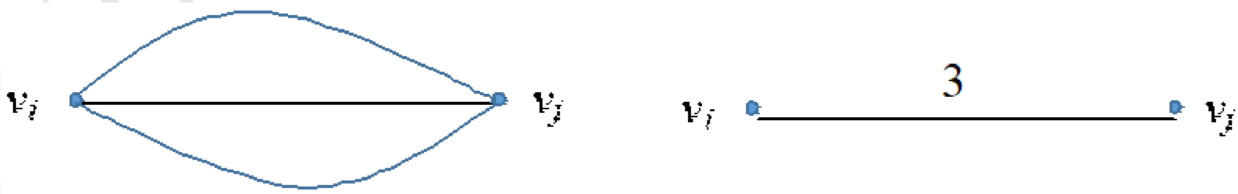


Рисунок IV.1.1.2. Кратные ребра.

Граф без петель и кратных ребер называется *простым графом*.

Вершины, которые не имеют инцидентных ребер, называются *изолированными вершинами*.

Число ребер, инцидентных вершине, называется *степенью вершины*.

Вершины, степени которых являются четными числами, называются *четными вершинами*.

Вершины с нечетной степенью называются *нечетными вершинами*.

Вершина, степень которой равна 1, называется *концевой вершиной* (степень изолированной вершины равна 0).

Сумма степеней всех его вершин равна удвоенному числу ребер, т.е. является четным числом. Количество нечетных вершин любого графа является четным.

В зависимости от свойств вершин и ребер, а также от типа отношений между ними графы подразделяются на несколько виды.

Граф, состоящий только из изолированных вершин, называется пустым или нуль-графом и обозначается через O_n, где n – число вершин графа.

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 в графе множества вершин V и ребер E конечны, то он называется *конечным графом*.

Конечный граф, содержащий n вершин и m ребер, называется $(n; m)$ – графом.

Если из каждой вершины графа исходит равное количество ребер и в каждую вершину заходят равное количество ребер, то такой граф называется *регулярным графом*.

Если в графе не содержатся петли и кратные ребра, то такой граф называется *простым графом*. Простой граф, в котором любые две вершины соединены ребром, называется *полным графом* и обозначается: U_n, где n – число вершин.

Граф без петель, но с кратными ребрами называется *мультиграфом*.

Граф, содержащий хотя бы одну петлю, называется псевдографом. В псевдографе могут быть кратные ребра.

Если для каждого ребра графа определено направление, то такой граф называется *ориентированным графом*.

Если все пары вершин графа связны в обоих направлениях, то такой граф будет *сильносвязным графом*.

Если каждое ребро графа имеет вес, то такой граф называется *взвешанным графом*, т.е. можно определить функцию $w : E \to R$, где R – множество действительных чисел, w –вес ребра и $w \geq 0$.

Если множество вершин простого графа V допускает такое разбиение на два непересекающиеся подмножества V_1 и V_2 (т.е. $V_1 \cap V_2 = \emptyset$), что не существует ребер, соединяющих вершины одного и того же подмножества, то он называется *биграфом* или *двудольным графом*.

Граф называется *плоским*, если его ребра не пересекаются в точках, отличных от вершин данного графа.

Между двумя графами можно ввести отношение, в том числе и многомерные. Например, для заданных двух графов можно определить отношение полного их совпадения (изоморфизм) или отношение включения (быть частью) одного графа в другом графе.

Два графа называются изоморфными, если они имеют одно и то же число вершин, и для любых 2-х вершин первого графа, соединенных ребром, соответствующие им вершины второго графа тоже соединены ребром и обратно.

Граф $G' = (V', E')$ называется частью графа $G = (V, E)$, если $V' \subset V$ и $E' \subset E$.

Часть графа, которая не содержит изолированные вершины, называется *подграфом*.

Часть графа, которая, наряду с некоторым подмножеством ребер графа, содержит и все вершины графа, называется *суграфом*.

В графах можно решать следующие задачи: *нахождение наикратчайшей пути из одной вершины в другую*, *нахождение* *числа замкнутных путей* и др. Для этого введем следующие определения:

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 $(v_1, v_2, v_3, \ldots, v_{m-1}, v_m)$, что пары $\{(v_1, v_2), (v_2, v_3), \ldots, (v_{\mathrm{m}-1}, v_{\mathrm{m}})\}$ станут ребрами. Этот путь может быть в обоих направлениях.

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*. Если 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*.

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

$$w(v_1, v_2, v_3, \ldots, v_{m-1}, v_m) = \sum_{i=1}^{m-1} w(v_i, v_{i+1}).$$

Граф можно представить с помощью матрицы смежности. Для этого введем отношение смежности между вершинами.

На множестве вершин V можно определить отношение смежности, представив каждое ребро как пару смежных вершин, т.е.

$$e_k = (v_i, v_j) = (v_j, v_i)$$

Множество вершин V вместе с определенным на нем отношением смежности полностью определяет граф G.

Граф G с числом вершин n можно представить *матрицей смежности* квадратной матрицей A размера $n \times n$, строки и столбцы которой соответствуют вершинам графа v_i и v_j, а ее элемент a_{ij} равен числу кратных ребер, связывающих вершины v_i и v_j, и $a_{ij} = 0$, если не существует ребро, идущее из вершины v_i в вершину v_j. Матрица смежности пустого графа, не содержащего ни одного ребра, состоит из одних нулей. Матрица смежности простого графа состоит из нулей и единиц, главная диагональ которой содержит только нули. Иногда, петля считается за два ребра, то есть значение диагонального элемента a_{ij} в этом случае равно удвоенному числу петель вокруг i–й вершины.

Граф может быть представлен *матрице инциндентности*.

Матрица инциндентности графа *G* есть матрица *A* размера $n \times m$, в которой число строк *n* соответствует количеству вершин, а число столбцов m – количеству рёбер.

Элементы этой матрицы определяются по правилу: (*i*; *j*) - элемент равен 1, если вершина v_i инцидентна ребру e_j, и равен 0, если v_i и e_j не инцидентны. Столбец, соответствующий ребру $(v_i, v_j) \in E$ содержит -1 в строке, соответствующей вершине v_j и 1, в строке, соответствующей вершине v_i. Во всех остальных случаях *0*.

Пусть даны две части графа G: G_1 и G_2, тогда можно определить:

1. Объединение графов – это граф $G = G_1 \cup G_2$, множества вершин и ребер которого определяются: $V = V_1 \cup V_2$, $E = E_1 \cup E_2$.

2. Пересечение графов – это граф $G = G_1 \cap G_2$, множества вершин и ребер которого определяются: $V = V_1 \cap V_2$, $E = E_1 \cap E_2$.

3. Если $V_1 \cap V_2 = \emptyset$ и $E_1 \cap E_2 = \emptyset$, то графы называются непересекающимися.

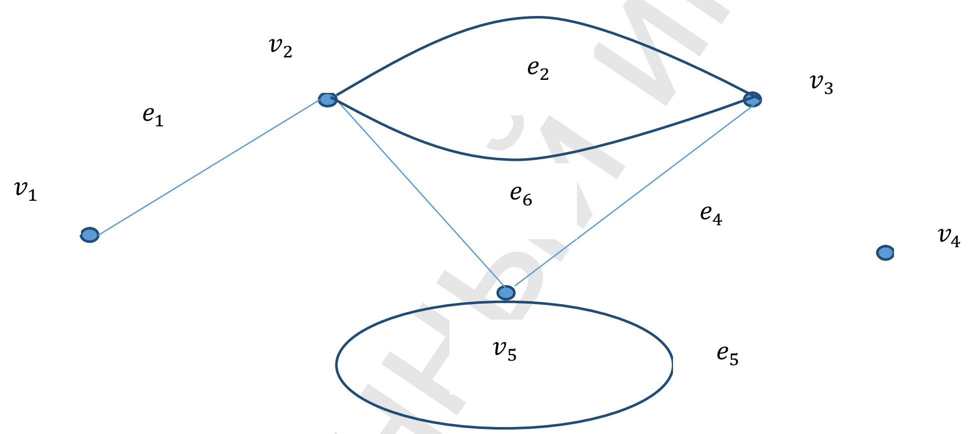
3. Сложение двух графов G_1 и G_2 по модулю 2 – это есть граф $G = G_1 \oplus G_2$, множество вершин которого $V = V_1 \cup V_2$, а

множество ребер $E = (E_1 \cup E_2) \backslash (E_1 \cap E_2)$.

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 рёбер.

Примеры IV.1.1.

1. Построим матрицу смежности для графа, представленного на рисунке ниже.



Матрица смежности этого графа имеет вид:

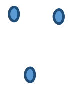
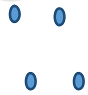
	v_1	v_2	v_3	v_4	v_5
v_1	0	1	0	0	0
v_2	1	0	2	0	1
v_3	0	2	0	0	1
v_4	0	0	0	0	0
v_5	0	1	1	0	2

Для изображенного в предыдущем примере графа матрица инцидентности имеет вид:

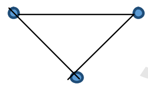
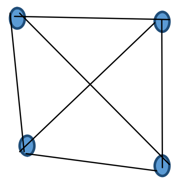
	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1	0	0	0	0	0

v_2	1	1	1	0	0	1
v_3	0	1	1	1	0	0
v_4	0	0	0	0	0	0
v_5	0	0	0	1	2	1

2. Нуль графы:

$\boldsymbol{O_1}$ - $\boldsymbol{O_2}$ - $\boldsymbol{O_3}$ -  $\boldsymbol{O_4}$ - 

3. Полные графы:

$\boldsymbol{U_2}$ - $\boldsymbol{U_3}$ -  $\boldsymbol{U_4}$ - 

Задание IV.1.1.

1. Постройте граф по данной матрице смежности.

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	1	0	1	0	0	0
v_2	0	0	0	1	1	0
v_3	0	0	1	0	0	2
v_4	2	0	1	0	0	0
v_5	0	0	0	1	0	1
v_6	0	1	0	0	1	0

2. Постройте граф, соответствующий данной матрице смежности.

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	0	1	2	1	0
v_2	0	2	0	0	1	0
v_3	1	0	0	2	0	0
v_4	2	0	2	0	1	0

v_5	1	1	0	1	2	1
v_6	0	0	0	0	1	0

Вопросы IV.1.1.

1. Какие ребра называются кратными?
2. Что такое петля?
3. Какие две вершины называются смежными?
4. Какая вершина называется изолированной?
5. Объясните, что означает инцидентность вершин и ребер?
6. Чем отличается матрица смежности от матрицы инциндентности?
7. Что такой путь в графе?
8. Чем отличается нуль граф от полного графа?
9. Что такой цикл?
10. Как определяется длина пути?

Тесты IV.1.1.

1. Как называются точки графа?

A) вершинами графа;

B) пунктами графа;

C) дугами графа;

D) узлами графа;

E) рёбрами графа.

2. Граф - это...

A) множество точек, две из которых обязательно соединяются линиями;

B) множество точек, которые никогда не соединяются линиями;

C) только две точки, которые соединяются линиями;

D) множество точек, которые могут соединяться линиями;

E) множество точек, которые могут соединяться между собой;

3.Линии, которые связывают вершины, называются...

A) рёбрами графа;

B) сторонами графа;

C) вершинами графа;

D) отрезками;

E) циклом.

IV.1.2. 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 потомком;

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 $A_1, A_2, ..., A_n$ – деревья, то можно построить новое дерево, в котором корнем является вершина v, а ребрами – исходящие из этой вершины и входящие в корни деревьев $A_1, A_2, ..., A_n$.

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

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

Определение дерева можно представить как на рисунке IV.1.2.1:

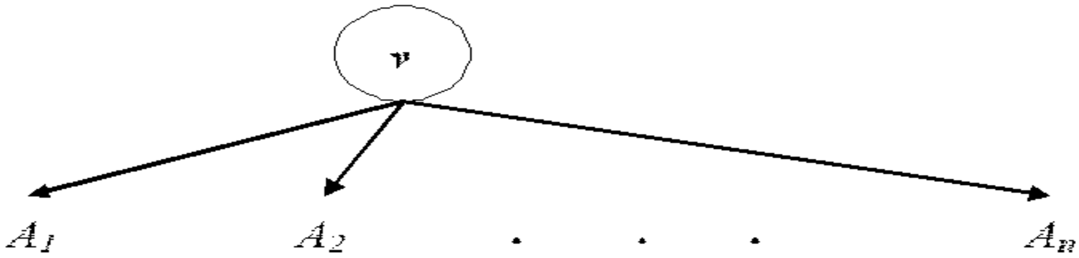


Рисунок IV.1.2.1. Определение дерева.

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 следуюшее 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 поддеревья пустые, то она есть лист. Формально бинарное дерево определяется так:

<биндерево > ::= nil | (<данные> < биндерево > < биндерево >), где nil – пусто.

В деревьях решаются следующие задачи: *обход деревьев*, *поиск дереве*, *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 поддерево и правое поддерево 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 IV.1.2.

1. Бинарное отношение может быть представлено в виде ориентированного графа как на рисунке IV.1.2.2, где 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.

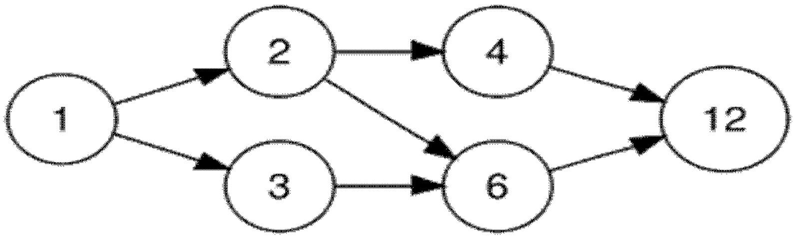


Рисунок IV.1.2.2. Представление бинарного отношения

2. На рисунке IV.1.2.3 показано бинарное дерево

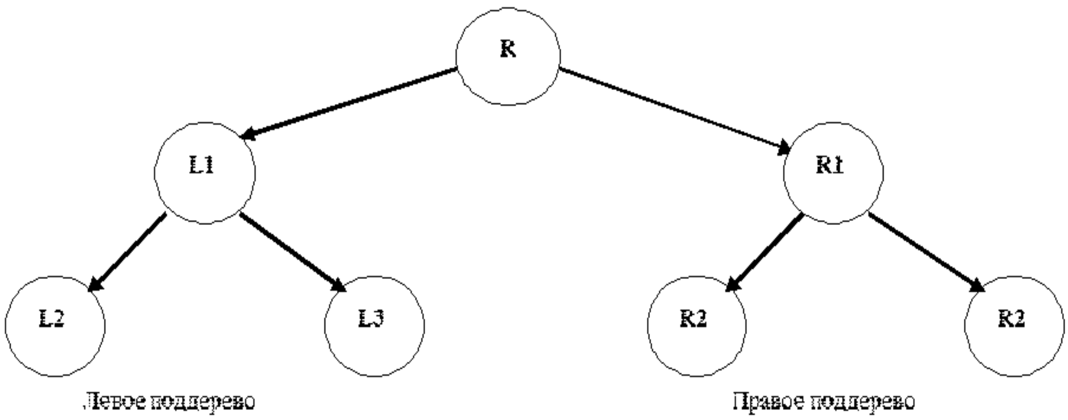


Рисунок IV.1.2.3. Бинарное дерево.

3. Обход бинарного дерева арифметического выражения ((3+1)*3/(9–5)+2–(3*(7–4)+6) свеху вниз и слева направо представлен на рисунке IV.1.2.4.

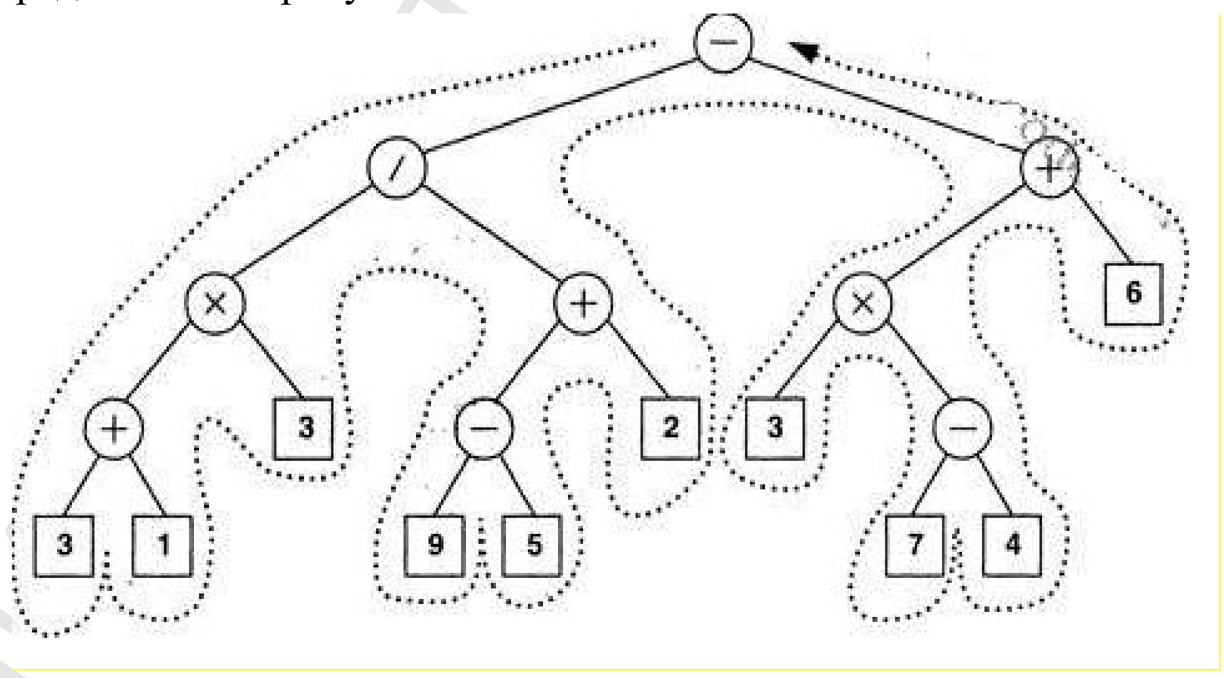


Рисунок IV.1.2.4. Обход дерева

Задания IV.1.2.

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 показан на рисунке IV.1.2.5 и содержит петлю вокруг вершины 1, в которой зависимости от конкретных приложений элемент a_{11} может считаться равным либо одному (как показано ниже), либо двум.

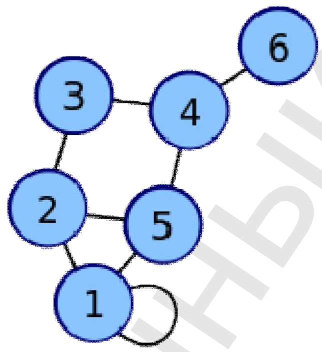


Рисунок IV.1.2.5. Неоринтированные граф.

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

Помощь:

1. Не умоляя общности, для облегчения построения требуемого графа будем расматривать 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

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

Вопросы IV.1.2.

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 IV.1.2.

1.На какие виды подразделяются графы?

A) ориентированный граф, неориентированный граф;

B) ориентированный граф, определенный граф;

C) определенный граф, неориентированный граф;

D) определенный граф, неопределенный граф;

E) неопределенный граф, неориентированный граф.

2. Что такое дерево?

A) граф без петель и без циклов;

B) граф без весов;

C) граф без сетей и циклов;

D) граф взвешанный;

E) граф ориентированный.

3.Что такое бинарное дерево?

A) дерево, в котором каждая вершина имеет не более двух потомков;

B) дерево, в котором имеются две вершин;

C) дерево, в котором нет цикла;
D) дерево, в котором нет петли;
E) дерево, в котором одна вершина не имеет прямых потомков.

V. ЯЗЫКИ, ГРАММАТИКИ И АВТОМАТЫ

V.1. Механизмы порождения языка

V.1.1. Основные понятия формальных языков

Рассмотрим основные понятия, связанные с формальными языками.

Определения V.1.1.1.

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 |ω|. 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 $\boldsymbol{T}$, то слово $\alpha\cdot\beta$ называется *конкатенацией* (*сцеплением*) слов α и β, которая получается в результате приписывания слова β в конец слова α. Обычно в записи знак · опускают и пишут просто $\alpha\beta$.

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

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

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

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

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

Замечание V.1.1.1.

Для любого непустого слово ω в алфавите $\boldsymbol{T}$ и пустого слово ε выполняются отношения ε⪽ω и ε⪾ω.

Если $\boldsymbol{T}$ – алфавит, то множество всех цепочек (слов) конечной длины в этом алфавите определяется как

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

где $\boldsymbol{T}^0 \rightleftharpoons \{\varepsilon\}$ – множество цепочек длины 0, $\boldsymbol{T}^k \rightleftharpoons \underbrace{\boldsymbol{T}\cdot\boldsymbol{T}\cdot\ldots\cdot\boldsymbol{T}}_{\boldsymbol{k}}$ – множество цепочек длины k, $k \geq 1$ – натуральные числа.

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

Очевидно, что не все цепочки из множества $\boldsymbol{T}^+$ 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 $\boldsymbol{L}$ является собственным подмножеством множества $\boldsymbol{T}^+$, т.е. $\boldsymbol{L} \subset \boldsymbol{T}^+$. Например, если в качестве элементов алфавита $\boldsymbol{T}$ взять казахских букв и различные специальные знаки, то казахский язык будет собственным подмножеством множества $\boldsymbol{T}^+$, 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 V.1.1.2. Пусть заданы два языка $\boldsymbol{L_1}$ и $\boldsymbol{L_2}$ в алфавите $\boldsymbol{T}$, т.е., $\boldsymbol{L_1}\subseteq\boldsymbol{T}^+$ и $\boldsymbol{L_2}\subseteq\boldsymbol{T}^+$, а также $\boldsymbol{U}$ – универсум, тогда можно определить:

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

где $\boldsymbol{L}^0 \rightleftharpoons \{\varepsilon\}, \boldsymbol{L}^{\boldsymbol{k}} \rightleftharpoons \underbrace{L \cdot L \cdot \ldots \cdot L}_{k}$, $k\geq 1$.

Замечания V.1.1.2.

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

Определения V.1.1.3. Пусть $\boldsymbol{L}\subseteq\ \boldsymbol{T}^*$, тогда можно ввести следующие обозначения и понятия:

1. $\boldsymbol{L}^{\mathrm{R}}$ – операция *обращение* над языком $\boldsymbol{L}$ определяется как $\boldsymbol{L}^{\mathrm{R}} = \{\tau^{\mathrm{R}}\colon\ \tau\in\boldsymbol{L}\}$.
2. Pref($\boldsymbol{L}$) – *множество префиксов* языка $\boldsymbol{L}$ определяется как Pref($\boldsymbol{L}$) $\rightleftharpoons \{\tau\colon\ \exists\xi(\xi\in\boldsymbol{L}\ \&\ \tau\Subset\xi)\}$, где $\Subset$ – отношение префикса.
3. Suf($\boldsymbol{L}$) – *множество постфиксов* языка $\boldsymbol{L}$ определяется как Suf($\boldsymbol{L}$) $\rightleftharpoons \{\tau\colon\ \exists\zeta(\zeta\in\boldsymbol{L}\ \&\ \tau\Supset\zeta)\}$, где $\Supset$ – отношение постфикса.

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

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

3. Множества $\boldsymbol{K}$ и $\boldsymbol{L}$ называются *равномощными*, если существует биекция из $\boldsymbol{K}$ в $\boldsymbol{L}$.

Примеры V.1.1. Пусть задан алфавит $\boldsymbol{T}=\{a, b, c, d\}$. Тогда:

1) $|a|=1$, $|bb|=2$, $|ccc|=3$, $|abcd|=4$.
2) Если $\alpha = ccc$ и $\beta = dddd$, то $\alpha\beta = cccdddd$.
3) $ab^2 = abb$, $(ab)^3 = ababab$.
4) Если $\alpha = ab$, $\beta = abcd$, $\xi = cd$, то $\beta = \alpha\xi$, т.е. $\alpha \Subset \beta$.
5) $\varepsilon\Subset abcd$, $a\Subset abcd$, $ab\Subset abcd$, $abc\Subset abcd$, $abcd\Subset abcd$.
6) $abcd\Supset\varepsilon$, $abcd\Supset d$, $abcd\Supset cd$, $abcd\Supset bcd$, $abcd\Supset abcd$.
7) Если $\tau = ca$ и $\xi = da$, то $\tau\Subset\omega$ и $\omega = cada$.
8) Если $\tau = bbb$, $\zeta = aaa$ и $\xi = ccc$, то $\tau\subseteq\omega$ и $\omega = aaabbbccc$.
9) Если $\omega = abcdabcd$ и $\tau = cd$, то $\tau\subseteq\omega$ и $|\omega|_\tau = 2$.

Задания V.1.1.

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

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

3. Пусть $\boldsymbol{T} = \{a, b, c\}$. $\boldsymbol{L_1} = \{\tau\in\boldsymbol{T}\text{: } |\tau| = 4\}$ и $\boldsymbol{L_2} = \{\tau\in\boldsymbol{T}^*\text{: } |\tau|_c = 1\}$. Вычислить число цепочек языка $\boldsymbol{L}_1\setminus\boldsymbol{L}_2$.

4. Пусть заданы языки $\boldsymbol{L_1}$ и $\boldsymbol{L_2}$. Выяснить равномощность языков

$\boldsymbol{L}_1\cdot(\boldsymbol{L}_2^{\boldsymbol{R}})$ и $\boldsymbol{L}_1^{\boldsymbol{R}}\cdot\boldsymbol{L}_2$.

5. Для языков $\boldsymbol{L}_1$ и $\boldsymbol{L}_2$ определите результат их *конкатенации и объединения*:

$\boldsymbol{L_1} = \{d, de, dee\}$ и $\boldsymbol{L_2} = \{\varepsilon, d, e, de, d\}$;
$\boldsymbol{L_1} = \{\varepsilon, d, de, dee\}$ и $\boldsymbol{L_2} = \{d, e, dee, d\}$;
$\boldsymbol{L_1} = \{\varepsilon, d, e, de, ded\}$ и $\boldsymbol{L_2} = \{\varepsilon, d, e, de, ed\}$;
$\boldsymbol{L_1} = \{d, e, dd, de, ee, ded\}$ и $\boldsymbol{L_2} = \{d, e, dd, de, ed\}$.
$\boldsymbol{L_1} = \{d, e, ded, ddde, eedd\}$ и $\boldsymbol{L_2} = \{d, e, ddd, ded, eee\}$.

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

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

Вопросы V.1.1.

1. Существует ли такой язык $\boldsymbol{L}$, что выполняется неравенство $\boldsymbol{L}^* \ne \{x^n: \ x \in \boldsymbol{L}, n \ge 0\}$?

2. Существует ли такой язык $\boldsymbol{L}$, что выполняется неравенство $(\boldsymbol{L}^R)^* \ne (\boldsymbol{L}^*)^R$?

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

Тесты V.1.1:

1. Пусть заданы языки $\boldsymbol{L_1} = \{aa,bb\}$, $\boldsymbol{L_2} = \{cc,dd\}$, тогда что будет конкатенация этих языков $\boldsymbol{L_1} \cdot \boldsymbol{L_2}$?

A) $\boldsymbol{L_1} \cdot \boldsymbol{L_2} = \{aacc, aadd, bbcc, bbdd\}$.
B) $\boldsymbol{L_1} \cdot \boldsymbol{L_2} = \{aaaa, dddd, bbbb, cccc\}$.
C) $\boldsymbol{L_1} \cdot \boldsymbol{L_2} = \{aabb\ aacc, bbaa, ddaa\}$.
D) $\boldsymbol{L_1} \cdot \boldsymbol{L_2} = \{aadd, bbdd, ccdd, dddd\}$.
E) $\boldsymbol{L_1} \cdot \boldsymbol{L_2} = \{aacc, aabb, aadd, aacc\}$.

2. Пусть задан язык $\boldsymbol{L} = \{a^m b a^n: \ m \le n\}$ является языком в алфавите $\{a, b\}$, тогда что содержит множество всех его подцепочек?

A) цепочки *b*, *ba*, *aba*, *baa*, *abaa*, *baaa*, *aabaa*, *abaaa* и т. д.
B) цепочки *a*, *ab*, *bab*, *abb*, *abab*, *aaaa*, *aaaab*, *abbaa* и т. д.
C) цепочки *aa*, *ab*, *ba*, ааа, *aabb*, *bdaa*, *aabba*, *abbba* и т. д.
D) цепочки *b*, *bb*, *ba*, *baba*, *abba*, *bbaa*, *abbaa*, *abbba* и т. д.
E) цепочки *b*, *aa*, *ab*, *baab*, *abba*, *baab*, *abbba*, *abbbb* и т. д.

3. Пусть задан язык $\boldsymbol{L} = \{ab^n: \ n \ge 0\}$ в алфавите $\{a, b\}$, тогда что содержит множество всех его подцепочек?

A) цепочки *a*, *ab*, *abb*, *abbb*, *abbbb*, *abbbbb* и т. д.
B) цепочки *a*, *ab*, *abb*, *abbb*, *abbbb*, *abbbbb* и т. д.
C) цепочки *a*, *ab*, *abb*, *abbb*, *abbbb*, *abbbbb* и т. д.

D) цепочки *a*, *ab*, *abb*, *abbb*, *abbbb*, *abbbbb* и т. д.

E) цепочки *a*, *ab*, *abb*, *abbb*, *abbbb*, *abbbbb* и т. д.

V.1.2. Формальные грамматики

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

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 V.1.2. Принимаются следующие соглашения:

(1) строчные латинские курсивные буквы *a*, *b*,..., *z* и арабские цифры 0, 1, ..., 9 обозначают терминалы;

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

(3) строчные греческие буквы α, β,..., ω 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 $\alpha \to \beta$;

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

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

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

$$\begin{array}{c} \alpha_1 \to \beta \\ \alpha_2 \to \beta \\ \dots \\ \alpha_m \to \beta \end{array}$$

(9) правила вида $\alpha \to \beta_1 | \beta_2 | \dots | \beta_n$ является сокращением n правил вида $\alpha \to \beta_1$, $\alpha \to \beta_2$, …, $\alpha \to \beta_n$ или:

$$\begin{array}{c} \alpha \to \beta_1 \\ \alpha \to \beta_2 \\ \dots \\ \alpha \to \beta_n \end{array}$$

(10) правило вида $\alpha_1 | \alpha_2 | \dots | \alpha_{\mathrm{m}} \to \beta_1 | \beta_2 | \dots | \beta_n$ является сокращением $\boldsymbol{m} \times \boldsymbol{n}$ правил, полученных из соглашении (6) и (7).

Определение V.1.2.1.*Формальной грамматикой* называется следующая четверка $\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$, где:

$\boldsymbol{T}$ – непустое конечное множество терминальных символов (терминалов);

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

$\boldsymbol{P}$ – непустое конечное множество правило подстановки вида $\alpha \to \beta$, где $\alpha \in (\boldsymbol{T} \cup \boldsymbol{N})^* \times \boldsymbol{N} \times (\boldsymbol{T} \cup \boldsymbol{N})^*$, $\beta \in (\boldsymbol{T} \cup \boldsymbol{N})^*$, то есть,

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

S – начальный нетерминал, $S \in \boldsymbol{N}$.

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 (порождаемой) 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.

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

Определения V.1.2.2.

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

1) S – выводимая цепочка грамматики $\boldsymbol{G}$;

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

2. Выводимая цепочка грамматики $\boldsymbol{G}$, не содержащая нетерминальных символов из $\boldsymbol{N}$, называется *терминальной цепочкой*, порождаемой грамматикой $\boldsymbol{G}$.

3. Если $\alpha = \gamma\xi\delta$, $\beta = \gamma\eta\delta$ и $\alpha\rightarrow\beta$, $\xi\rightarrow\eta$ – правила вывода грамматики $\boldsymbol{G}$, то говорят, что между цепочками α и β установлено *отношение непосредственной выводимости*, которое означает, что в грамматике $\boldsymbol{G}$ 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 $\boldsymbol{G}$ 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 $\alpha_0, \alpha_1, ..., \alpha_k$ 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 $\boldsymbol{G}$.

Если существует $i\geq1$ (или $i\geq0$) выполняется отношение $\alpha\Rightarrow^{i}\beta$, то это записывается как $\alpha\Rightarrow^{+}\beta$ (или $\alpha\Rightarrow^{*}\beta$). Здесь через $\Rightarrow^{+}$

обозначается транзитивное замыкания отношения $\Rightarrow$, а через $\Rightarrow^*$ – рефлексивное и транзитивное замыкания 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 V.1.2: $\alpha\Rightarrow^*\beta$ 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 V.1.2.3.

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

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

3. *Языком* $\boldsymbol{L}(\boldsymbol{G})$, порождаемым грамматикой $\boldsymbol{G}$, называется множество терминальных цепочек, которые выводятся из одного начального нетерминала S посредством применения правило подстановки из множества $\boldsymbol{P}$, т.е. формально записывается как

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

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

Примеры V.1.2.

1. Пусть порождается скобочное алгебраическое выражение в инфиксной записи с помощью грамматикой $\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$, где

$\boldsymbol{T} = \{+,\ -, /, *, (,), a\}$, $\boldsymbol{N} = \{S, E, T, F\}$,

$\boldsymbol{P} = \{S\rightarrow E, E\rightarrow E+T \mid E-T \mid T, T\rightarrow T*F|T/F|F, F\rightarrow a|(E)\}$.

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

3. Две грамматики для порождения алгебраичесих выражений, образованных операндами i, n и операциями +, * с одинаковыми терминальными символами $\boldsymbol{T} = \{i, n, (,), +, *\}$ и

нетерминальными символами $N = \{S, F, H\}$, но с разными правилами:
$P_1 = \{S \to S+F,\ S \to F,\ F \to F*H,\ F \to H, F \to H,\ H \to i,\ H \to n,\ H \to (S)\}$;
$P_2 = \{S \to S+F|F,\ S \to S+F|S*F|F,\ F \to F*H|H,\ H \to i|n|(S)\}$;
эквивалентны.

Задания V.1.2:

1. Построить все сентенциальные формы для грамматики с правилами:

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

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

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

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

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

4. Описать язык, порождаемый грамматикой

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

5. Описать язык, порождаемый грамматикой

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

6. Описать язык, порождаемый грамматикой

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

V.1. Описать язык, порождаемый грамматикой

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

8. Описать язык, порождаемый грамматикой

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

9. Описать язык, порождаемый грамматикой

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

Вопросы V.1.2.

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 $\boldsymbol{L_1} \setminus \boldsymbol{L_2}$?

8.Как определяется $\boldsymbol{L_1} \triangle \boldsymbol{L_2}$?

9.Как определяется $\boldsymbol{L_1} \cdot \boldsymbol{L_2}$?

10. Как определяется $\boldsymbol{L}^+$

11. Чему равно $\boldsymbol{L}^0$?

12. Чему равно $\boldsymbol{L}^k$?

13. Как строится обращение языка $\boldsymbol{L}^R$?

14. Существует ли язык $\boldsymbol{L}$, что выполняется $(\boldsymbol{L}^R)^* \neq (\boldsymbol{L}^*)^R$?

15. Как получается префикс языка Pref($\boldsymbol{L}$)?

16. Как получается постфикс языка Suf($\boldsymbol{L}$)?

17. Как определяется Substr($\boldsymbol{L}$)?

Тесты V.1.2.

1.Как правильно определяется алфавит формальных языков?

A) некоторое конечное множество символов;

B) любая конечная последовательность символов;

C) подмножество цепочек конечной длины;

D) набор токенов, которые могут представлять определенную лексему в исходной программе

E) набор всех операторов сравнения

2. Как правильно определяется цепочки в алфавите V?

A) некоторое конечное множество символов;

B) любая конечная последовательность символов алфавита V;

C) подмножество цепочек конечной длины алфавита V;

D) набор токенов, которые могут представлять определенную лексему в исходной программе;

E) набор всех операторов сравнения.

3. Что обозначается символом |α|?

A) длина строки алфавита;

B) конечное множество символов;

C) последовательность символов;

D) язык алфавита V;

E) длина цепочки.

V.1.3. Регулярные множества и регулярные выражения

Пусть $\varnothing$ – пустое множество, $\{\varepsilon\}$ – множество пустых цепочек $\boldsymbol{T}$ – конечный алфавит и t – символ из $\boldsymbol{T}$, т.е. $t \in \boldsymbol{T}$. Тогда можно дать следующее определение.

Определение V.1.3.1. *Регулярное множество* в алфавите $\boldsymbol{T}$ определяется рекурсивно следующим образом:

Базис рекурсии:

(1) $\varnothing$ – регулярное множество в алфавите $\boldsymbol{T}$;

(2) $\{\varepsilon\}$ – регулярное множество в алфавите $\boldsymbol{T}$;

(3) $\{t\}$ – регулярное множество в алфавите $\boldsymbol{T}$;

Рекурсивное расширение: Если $\boldsymbol{P}$ и $\boldsymbol{Q}$ – регулярные множества в алфавите $\boldsymbol{T}$, то регулярными являются:

(4) $\boldsymbol{P} \cdot \boldsymbol{Q}$ – конкатенация множеств $\boldsymbol{P}$ и $\boldsymbol{Q}$;

(5) $\boldsymbol{P} \cup \boldsymbol{Q}$ – объединение множеств $\boldsymbol{P}$ и $\boldsymbol{Q}$;

(6) $\boldsymbol{P}^*$ – итерация множества $\boldsymbol{P}$;

Рекурсивное заключение:

Регулярное множество в алфавите $\boldsymbol{T}$ определяются только правилами (1) – (6).

Итак, множество в алфавите $\boldsymbol{T}$ регулярно тогда и только тогда, когда оно либо $\varnothing$, либо $\{\varepsilon\}$, либо $\{t\}$ для некоторого $t \in \boldsymbol{T}$, либо его можно получить из этих множеств применением конечного числа операций объединения, конкатенации и итерации.

Определение V.1.3.2. Регулярные выражения, описывающие регулярные множества в конечном алфавите $\boldsymbol{T}$, определяются рекурсивно следующим образом:

Базис рекурсии:

(1) ϕ – регулярное выражение, задающее регулярное множество $\varnothing$ в алфавите $\boldsymbol{T}$;

(2) ε – регулярное выражение, задающее регулярное множество $\{\varepsilon\}$ в алфавите $\boldsymbol{T}$,

(3) если $t \in \boldsymbol{T}$, то t – регулярное выражение, задающее регулярное множество $\{t\}$ в алфавите $\boldsymbol{T}$;

Рекурсивное расширение:

Если p и q – регулярные выражения, обозначающие регулярные множества $\boldsymbol{P}$ и $\boldsymbol{Q}$ в алфавите $\boldsymbol{T}$ соответственно, то:

(4) $p \cdot q$ – регулярное выражение, задающее $\boldsymbol{P \cdot Q}$ в алфавите $\boldsymbol{T}$;

(5) $p \vee q$ – регулярное выражение, задающее $\boldsymbol{P \cup Q}$ в алфавите $\boldsymbol{T}$;

(6) p^* – регулярное выражение, задающее $\boldsymbol{P}^*$ в алфавите $\boldsymbol{T}$;

Рекурсивное заключение:

Регулярные выражения определяются только правилами (1) – (6).

Для построения регулярного выражения использовались операция конкатенация $\cdot$, операция дизъюнкция $\vee$ и операция итерация *. Наивысшим приоритетом обладает операция *, затем идет $\cdot$, а операция $\vee$ будет последним. Обычно в записи регулярного выражения знак операции «·» опускается.

Мы будем пользоваться записью p^+ для сокращенного обозначения выражения pp^* и, кроме того, будем устранять из регулярных выражений лишние скобки там, где это не может привести к недоразумениям. Так, например, регулярное выражение $0 \vee 10^*$ означает $(0 \vee (1(0^*)))$.

Для каждого регулярного множества можно найти по крайней мере одно регулярное выражение, задающее это множество и обратно, для каждого регулярного выражения можно построить регулярное множество, обозначаемое этим выражением.

Отметить, что для каждого регулярного множества существует бесконечно много обозначающих его регулярных выражений.

Будем говорить, что два регулярных выражения равны, если они обозначают одно и то же множество. Для эквивалентного преобразования имеются алгебраические законы.

При практическом описании лексических структур бывает полезно сопоставлять регулярным выражениям некоторые имена, и ссылаться на них по этим именам. Для определения таких имен мы будем использовать запись вида

$d_1 = r_1$

$d_2 = r_2$

...

$d_n = r_n$

где d_i - различные имена, а каждое r_i - регулярное выражение над символами $\boldsymbol{T} \cup \{d_1, d_2, ..., d_{i-1}\}$, т.е. символами основного алфавита и ранее определенными символами-именами. Таким образом, для любого r_i можно построить регулярное выражение над $\boldsymbol{T}$, повторно заменяя имена регулярных выражений на обозначаемые ими регулярные выражения.

Если α, β, γ – регулярные выражения, отличные от ϕ и ε, то:

(1) $\phi\cdot\alpha=\alpha\cdot\phi=\phi$

(2) $\varepsilon\cdot\alpha=\alpha\cdot\varepsilon=\alpha$

(3) $\alpha\cdot\beta\neq\beta\cdot\alpha$

(4) $\phi\vee\alpha=\alpha\vee\phi=\alpha$

(5) $\alpha\vee\alpha=\alpha$

(6) $\alpha\vee\beta=\beta\vee\alpha$

(7) $\alpha\cdot(\beta\cdot\gamma)=(\alpha\cdot\beta)\cdot\gamma=\alpha\cdot\beta\cdot\gamma$

(8) $\alpha\vee(\beta\vee\gamma)=(\alpha\vee\beta)\vee\gamma=\alpha\vee\beta\vee\gamma$

(9) $\alpha\cdot(\beta\vee\gamma)=\alpha\cdot\beta\vee\alpha\cdot\gamma$

(10) $(\alpha\vee\beta)\cdot\gamma=\alpha\cdot\gamma\vee\beta\cdot\gamma$

(11) $\varepsilon^*=\varepsilon$

(12) $\phi^*=\varepsilon$

(13) $\alpha^*=\alpha\vee\alpha^*$

(14) $(\alpha^*)^*=\alpha^*$

Примеры V.1.3.

1. Регулярное выражение $(0 \vee 1)^*011$ определяет регулярное множество всех цепочек, составленных из 0 и 1, оканчивающихся цепочкой *011*.

2. Регулярное выражение $a(a \vee 0)^*$ определяет регулярное множество всех цепочек из $\{0, a\}^*$, начинающихся с *a*.

3. Регулярное выражение $(a \vee b)(a \vee b \vee 0 \vee 1)^*$ определяет регулярное множество всех цепочек из $\{0, 1, a, b\}^*$, начинающихся с *a* или *b*.

4. Регулярное выражение $(00 \vee 11)^* \cdot ((01 \vee 10) \cdot (00 \vee 11)^* \cdot (01 \vee 10) \cdot (00 \vee 11)^*)^*$ определяет регулярное множество всех цепочек нулей и единиц, содержащих четное число нулей и четное число единиц.

5. Регулярное выражение $a(a \vee b)^*$ определяет множество всевозможных цепочек, состоящих из *a* и *b*, начинающихся с *a*.

6. Регулярное выражение $(a \vee b)^*(a \vee b)(a \vee b)^*$ определяет множество всех непустых цепочек, состоящих из *a* и *b*, т.е. множество $\{a, b\}^+$.

7. Регулярное выражение $((s \vee 1)(s \vee 1)(s \vee 1))^*$ определяет множество всех цепочек, состоящих из 0 и 1, длины которых делятся на 3.

8. Регулярное выражение $\varepsilon \vee \phi$ определяет множество $\{\varepsilon\} \cup \varnothing$.

9. Множества чисел в десятичной записи:

$Digit = 0 \vee 1 \vee 2 \vee 3 \vee 4 \vee 5 \vee 6 \vee 7 \vee 8 \vee 9$;

$Integer = Digit^+$;

$Fraction = Integer \vee \varepsilon$;

$Exponent = (E(+ \vee - \vee \varepsilon)Integer) \vee \varepsilon$.

Задания V.1.3.

1. Найти регулярное выражение, задающее множество $\{a, b\}^*$.

2. Найти регулярное выражение, задающее множество $\{a, bc^*\}^*$.

3. Найти регулярное выражение, задающее множество $\{ab, cd\}^*$.

4. Найти регулярное выражение, задающее множество $\{ab, b\}^*$.

5. Найти регулярное выражение, задающее множество $\{a^*, b^*\}^*$

6. Найти регулярное выражение для языка
$\{\omega \in \{a, b\}^* : \ (|\omega|_a - |\omega|_b) ... 3\}$.

7. Найти регулярное выражение для языка
$\{\omega \in \{a, b\}^* : \ (|\omega|_a - |\omega|_b) ... 4\}$.

8. Найти регулярное выражение для языка $\boldsymbol{L}_1 \cap \boldsymbol{L}_2 \cap \boldsymbol{L}_3$, где $\boldsymbol{L}_1 = (aaab \vee c \vee d)^*$, $\boldsymbol{L}_2 = (a^*ba^*ba^*bc \vee d)^*$, $\boldsymbol{L}_3 = ((a \vee b)^*c(a \vee b)^*cd)^*$.

1. Упростите регуляроное выражение $(a \vee b \vee ab)^*$.
2. Упростите регуляроное выражение $(a^*b)^* \vee (b^*a)^*$.
3. Упростите регуляроное выражение $(ba \vee a^*ab)^*$.

Вопросы V.1.3.

1. Как определяется регулярное выражение в алфавите $\boldsymbol{T}$?
2. Какое множество определяет регулярное выражение $(ab)^+$?
3. Какое множество определяет регулярное выражение $(aa \vee bb)$?
4. Какое множество определяет регулярное выражение $a(\varepsilon \vee a) \vee b$?
5. Какое множество определяет регулярное выражение $a(a \vee b)^*$?
6. Какое множество определяет регулярное выражение ε^*?
7. Какое множество определяет регулярное выражение $((a \vee bc)^* \cdot a)$?
8. Какое множество определяет регулярное выражение $(c \cdot (ab \vee cd)^*)$?
9. Какое множество определяет регулярное выражение ϕ?
10. Равны ли регулярные выражения $((a \vee b)^* \vee aa)^*$ и $(aa \vee b \vee ab)^*$?
11. Равны ли регулярные выражения $(ab \vee b)^*(a \vee b)$ и $(a \vee b)^*(a \vee b)^*$?

12. Равны ли регулярные выражения $(b \vee cd^*a)^*cd^*$ и $b^*c(d \vee ab^*c)^*$?

13. Как можно упростить следующие *регулярные выражения:*

$(00^*)0 \vee (00)^*$?

$(0 \vee 1)(\varepsilon \vee 00)^+ \vee (0 \vee 1)$?

$(a \vee b)(\varepsilon \vee ab)^+ \vee (b \vee a)$?

$(aa \vee \varepsilon)(a \vee b)\ ab$?

14. Будут ли эквивалентными заданные регулярные выражения $(p^*q^*)^* = (q^*p^*)^*$?

15. Будут ли эквивалентными заданные регулярные выражения $p(qp)^*$ и $(pq)^*p$?

16. Будут ли эквивалентными заданные регулярные выражения $p^*(p \vee q)^*$, $(p \vee qp^*)^*$ и $(p \vee q)^*$?

Тесты V.1.3.

1. Какое регулярное множество определяет регулярное выражение $(a \vee b)^*$?

A) $\{a, b\}^*$

B) $\{aa, b\}^*$

C) $\{aaa, b\}^*$

D) $\{a, bb\}^*$

E) $\{a, bbb\}^*$

2. Какое регулярное множество определяет регулярное выражение $a(\varepsilon \vee a) \vee b$?

A) $\{a, b, aa\}$

B) $\{a, a, aa\}$

C) $\{b, b, bb\}$

D) $\{a, b, bb\}$

E) $\{aa, bb, aa\}$

3. Какое регулярное множество определяет регулярное выражение ϕ^*?

A) $\{\varepsilon\}$

B) {ε, ε}
C) {ε, ε, ε}
D) {ε, ε, ε, ε}
E) {ε ε, ε, ε, ε}

V.2. Механизмы распознавания языка

V.2.1. Конечные автоматы

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 управляюшего устройства, а также может иметь рабочую (внешнюю и внутренюю) память.

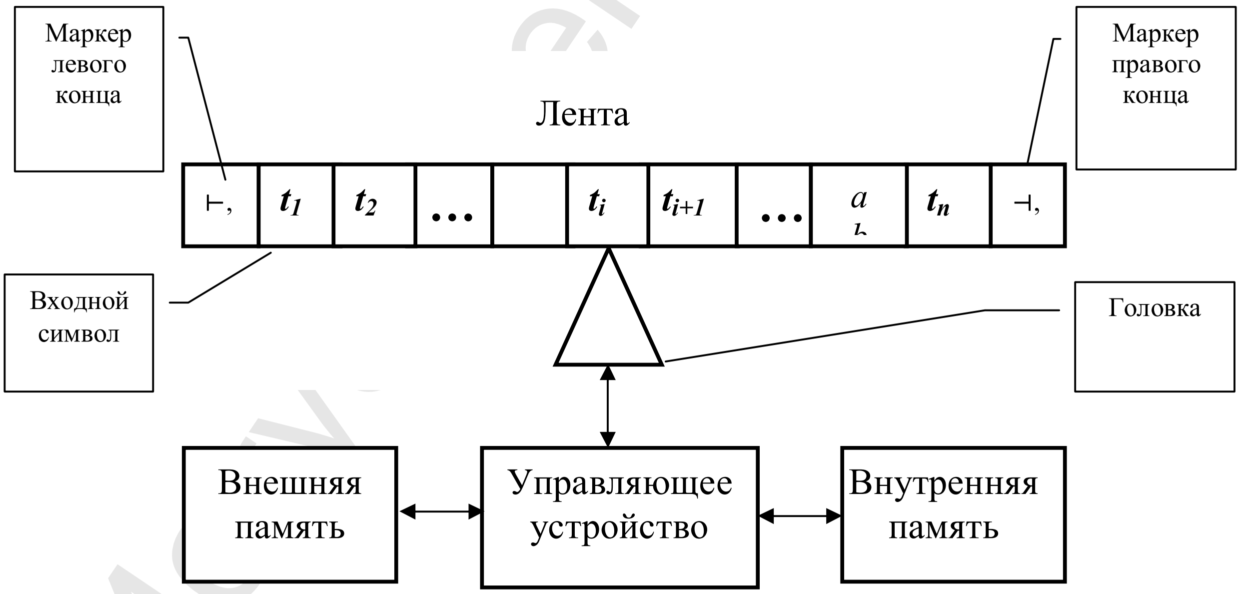


Рисунок V.2.1. Состав и структура конечного автомата.

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. 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 – это устройство, которое управляет поведением автомата и имеет конечную внутреннюю память для хранения конечного множества состояний. Он управляет поведением автомата с помощью функции (отношении), 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 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, являюшимися 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 обозревает 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 состояние записывается во внутренюю 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 V.2.1.1.

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

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

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

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 дается так:

Определение V.2.1.1. *Недетерминированный конечный автомат* (НКА) определяется семеркой $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{I}, \boldsymbol{F}, \vdash, \dashv, \Delta \rangle$, где:

$\boldsymbol{Q}$ – конечное множество состояний управляющего устройства;

$\boldsymbol{T}$ – конечное множество входных символов, $\boldsymbol{Q} \cap \boldsymbol{T} = \varnothing$;

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

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

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

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

Детерминированный конечный автомат (ДКА) является частным случаем НКА.

Определение V.2.1.2. Конечный автомат $\boldsymbol{M}$ = $<\boldsymbol{Q},\boldsymbol{T},\boldsymbol{I},\boldsymbol{F},\vdash,\dashv,\Delta>$ называется *детерминированным*, если:

(1) множество начальных состояний $\boldsymbol{I}$ содержит ровно один элемент;

(2) для каждого перехода $<q, \tau, p>\in\Delta$ выполняется $|\tau|= 1$;

(3) для любого состояния $q\in\boldsymbol{Q}$ и для любого символа $t\in\boldsymbol{T}$ существует не более одного состояния $p\in\boldsymbol{Q}$ со свойством $<q, t, p>\in\Delta$;

(4) остальные символы остаются такими же, как и в НКА.

Замечания V.2.1.1:

1. Иногда вместо множества отношения переходов Δ, принимающего логическое значение 'истина' или 'ложь', используется функция переходов δ, принимающая значение в виде символа из множества $\boldsymbol{Q}$, где δ: $\boldsymbol{Q}\times\boldsymbol{T}^*\rightarrow \vartheta(\boldsymbol{Q})$ – в случае НКА и δ: $\boldsymbol{Q}\times\boldsymbol{T}^*\rightarrow \boldsymbol{Q}$ – в случае ДКА. От функции δ можно легко перейти к отношению Δ, положив $\Delta = \{<q, \tau, \delta(q, \tau)>: \ q\in\boldsymbol{Q}, \tau \in\boldsymbol{T}^*\}$.

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

1)для отношения переходов: $<q,\tau,\{p\}>$ – в случае НКА, $<q,\tau,p>$ – 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 $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{I}, \boldsymbol{F}, \vdash,\dashv, \delta>$.

Переходы КА можно изображать в виде *диаграммы*, в которой каждое состояние обозначается кружком, а переход – стрелкой. Стрелка из состояния $q\in\boldsymbol{Q}$ в состояние $p\in\boldsymbol{Q}$ помеченная цепочкой $\tau\in\boldsymbol{T}^*$, показывает, что $<q, \tau, p>$ (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.

Примеры V.2.1.2.

1. Для КА $\boldsymbol{M}_1$ с одним переходом и параметрами: $\boldsymbol{Q}=\{q,p\}$; $\boldsymbol{T^*}=\{\tau\}$, $\boldsymbol{I}=\{q\}$, $\boldsymbol{F}=\{p\}$, $\delta(q,\ \tau)=p$ диаграмма показана на рис. V.2.1.2.

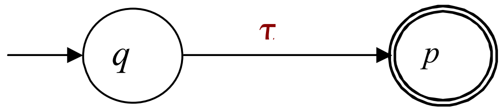


Рисунок V.2.1.2. Диаграмма КА $\boldsymbol{M}_1$ с одним переходом.

2. Пусть КА $\boldsymbol{M}_2$ имеет параметры: $\boldsymbol{Q}=\{1,2\}$, $\mathbf{T}=\{a,b\}$, $\boldsymbol{I}=\{1\}$, $\boldsymbol{F}=\{2\}$, $\Delta=\{<1, aaa, 1>, <1, ab, 2>, <1, b, 2>, <2, \varepsilon, 1>\}$. На рисунке V.2.1.3 показана диаграмма переходов НКА $\boldsymbol{M}_2$, в которой в качестве меток ребер использованы регулярные выражения *aaa*, *ab*, *b*, ε. Такое представление облегчает построение диаграммы, делая её компактной и наглядной.

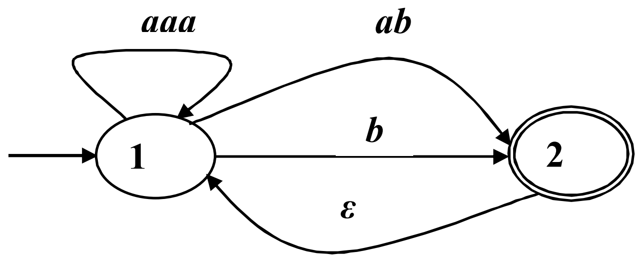


Рисунок. V.2.1.3. Диаграмма КА M_2 с регулярным выражением

3. КА $\boldsymbol{M}_3$ для распознавания идентификаторов, которые состоят только из букв и цифр и начинаются с буквы, будет иметь следующие параметры $\boldsymbol{Q}=\{1,2\}$, $\boldsymbol{T}=\{b,d\}$, $\boldsymbol{I}=\{1\}$, $\boldsymbol{F}=\{2\}$, $\delta(1,b)=2$, $\delta(2,b)=2$, $\delta(2,d)=2$, где b – буква, d – цифра. Диаграмма КА $\boldsymbol{M}_3$ представлена на рисунке V.2.1.4.

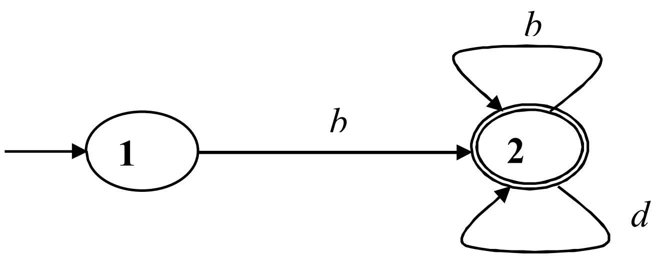


Рисунок V.2.1.4. Диаграмма КА $\boldsymbol{M}_3$ для идентификатора.

Замечание V.2.1.3. 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 V.2.1.5 представлена диаграмма КА $\boldsymbol{M}_4$ с параллельными переходами для цепочек *ab*, *b*.

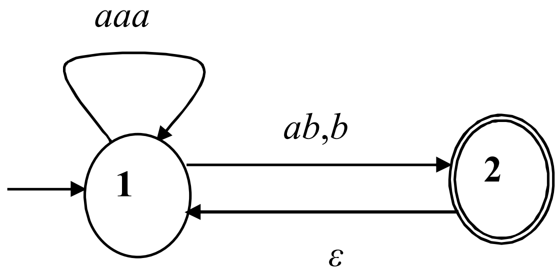


Рисунок V.2.1.5. Диаграмма КА $\boldsymbol{M}_4$. с параллельными переходами

Соглашение V.2.1.1. 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 V.2.1.3. В таблице V.2.1.1 показана функция переходов δ КА $\boldsymbol{M}_5$, определенная следующими множествами $\boldsymbol{Q} = \{q_s, q_1, q_2, q_3\}$ и $\boldsymbol{T} = \{t_1, t_2, t_3\}$.

Таблица V.2.1.1. Значения функции переходов δ КА $\boldsymbol{M}_5$.

δ		Вход		
		t_1	t_2	t_3
Состояние	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 V.2.1.1 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 $\boldsymbol{M}$ с начальным состоянием $q_s \in \boldsymbol{Q}$, текущим состоянием $q \in \boldsymbol{Q}$, заключительным состоянием $q_f \in \boldsymbol{Q}$ и неиспользованной текущей входной цепочкой $\tau \in \boldsymbol{T}^*$. Тогда можно дать следующее определение.

Определения V.2.1.3.

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

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

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

Замечание V.2.1.4. 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 $\boldsymbol{M}$ задается бинарным отношением $\models_{\boldsymbol{M}}$, определенным над его конфигурациями из множества $\boldsymbol{Q} \times \boldsymbol{T}^*$. При этом, если известен автомат, то букву $\boldsymbol{M}$ при отношении $\models_{\boldsymbol{M}}$ можно опустить.

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

Примеры V.2.1.4. Пусть $\tau = abba$. Тогда в диаграмме КА $\boldsymbol{M_2}$ на рисунке V.2.1.3 имеется такт в виде отношения $(1, abba) \models (2, ba)$.

Определение V.2.1.4. $\models^k$ является k–той степенью отношения $\models$, если существует цепочка из k+1 конфигураций

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

такая, что для любого i $(1 \le i \le k)$ 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 транзитовное замыкание отношения $\models$.

Определение V.2.1.5. Автомат $\boldsymbol{M}$ распознает входную цепочку τ, если выполняется отношение $(q_s,\ \tau) \models^* (q_f,\ \varepsilon)$.

Примеры V.2.1.5. Пусть $\tau = aaaab$. Тогда в КА $\boldsymbol{M}_2$ на рисунке V.2.1.3 выполняются следующие отношения $(1,\ aaaab) \models (1,\ ab)$ и $(1,\ ab) \models (2,\ \varepsilon)$.

Определение V.2.1.6. Если язык $\boldsymbol{L}$ состоит только из распознанных автоматом $\boldsymbol{M}$ входных цепочек, то этот язык распознается автоматом $\boldsymbol{M}$ и он обозначается через $\boldsymbol{L}(\boldsymbol{M})$, т.е.

$\boldsymbol{L}(\boldsymbol{M}) \rightleftharpoons \{\tau\colon\ \tau \in \boldsymbol{T}^*\ \ \&\ \ (q_s,\ \tau) \models^* (q_f,\ \varepsilon)\}$.

Примеры V.2.1.6. Пусть $\boldsymbol{M}_6 = \langle \{q_s, q_1, q_f\}, \{0,1\}, q_s, \{q_f\}, \vdash, \dashv, \delta \rangle$ имеет следующие отношения переходов:

$$\langle q_s,0,\{q_1\}\rangle, \langle q_s,1,\{q_s\}\rangle, \langle q_1,0,\{q_f\}\rangle, \langle q_1,1,\{q_s\}\rangle, \langle q_f,0,\{q_f\}\rangle, \langle q_f,1,\{q_f\}\rangle$$

КА $\boldsymbol{M}_6$ 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 $\boldsymbol{M}_6$ 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 $(q_s,01001) \models (q_1,1001) \models (q_s,001) \models (q_1,01) \models (q_f,1) \models (q_f, \varepsilon)$.

Итак, $01001 \in \boldsymbol{L}(\boldsymbol{M}_6)$.

Диаграмма этого автомата показана на рисунке V.2.1.6.

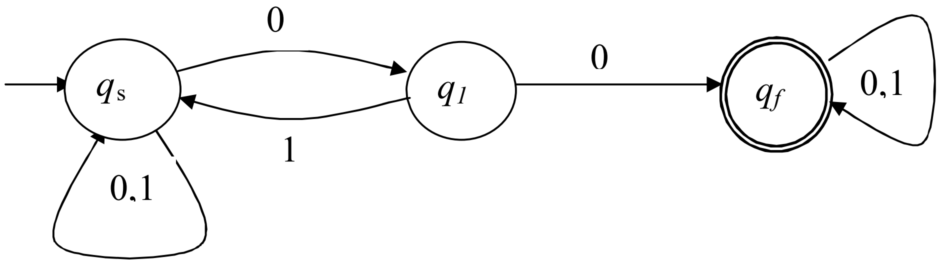


Рисунок V.2.1.6. Диаграмма КА $\boldsymbol{M}_6$.

***Определения V.2.1.7*:**

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

2. Путь называется *успешным* если его начало принадлежит $\boldsymbol{I}$, а конец принадлежит $\boldsymbol{F}$.

Замечание V.2.1.5. Для любого состояния $q \in \boldsymbol{Q}$ существует путь $<q>$. Его метка ε, начало и конец совпадают.

Примеры V.2.1.7. Рассмотрим КА $\boldsymbol{M_2}$ на рисунке V.2.1.3 Пусть $\tau = baaab$. Тогда путь <1,<1,*b*,2>,2,<2,ε,1>, 1,<1,*aaa*,1>, 1, <1,*b*,2>,2> является успешным. Его меткой является *baaab*, а длина равна 4, т.е.: $q_0=1$, $q_1=2$, $q_2=1$, $q_3=1$, $q_4=2$; $r_1=<1,b,2>$, $r_2=<2,\varepsilon,1>$, $r_3=<1,aaa,1>$, $r_4=<1,b,2>$; $\tau_1=b$, $\tau_2=\varepsilon$, $\tau_3=aaa$, $\tau_4=b$.

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

Определения V.2.1.8:

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

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

Замечание V.2.1.6. Если $\boldsymbol{I} \cap \boldsymbol{F} \neq \varnothing$, то язык, распознаваемый КА $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, \vdash, \dashv, \boldsymbol{I}, \boldsymbol{F}, \Delta \rangle$ содержит пустую цепочку ε.

Примеры V.2.1.8. Если КА $\boldsymbol{M}_7 = \langle \boldsymbol{Q}, \boldsymbol{T}, \vdash, \dashv, \boldsymbol{I}, \boldsymbol{F}, \Delta \rangle$ задан как $\boldsymbol{Q} = \{q_1,q_2\}$, $\boldsymbol{T} = \{a,b\}$, $\boldsymbol{I} = \{q_1\}$, $\boldsymbol{F} = \{q_1,q_2\}$, $\Delta = \{\langle q_1,a,q_2\rangle, \langle q_2,b,q_1\rangle\}$, то он является детерминированным и распознает следующий язык:

$\boldsymbol{L}(\boldsymbol{M}_7) = \{(ab)^n\text{: } n \geq 0\} \cup \{(ab)^n a\text{: } n \geq 0\}$.

Диаграмма этого автомата показана на рисунке V.2.1.7.

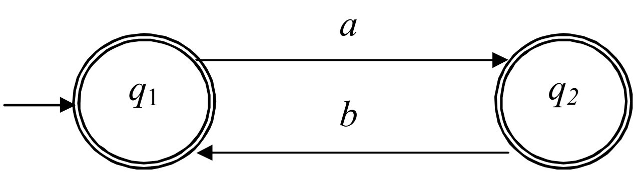


Рисунок V.2.1.7. Диаграмма КА $\boldsymbol{M}_7$.

Определение V.2.1.9. ДКА $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, \vdash, \dashv, \boldsymbol{I}, \boldsymbol{F}, \Delta \rangle$, называется *полным*, если для каждого состояния $q \in \boldsymbol{Q}$ и для каждого символа $t \in \boldsymbol{T}$ найдется такое состояние $p \in \boldsymbol{Q}$, что $\langle q, t, p \rangle \in \Delta$, т.е. $\delta(q, t) = p$.

Примеры V.2.1.9. На рисунке V.2.1.8 показана диаграмма автомата $\boldsymbol{M}_8$ с параметрами $\Delta = \{\langle 1,a,2\rangle, \langle 1,b,3\rangle, \langle 2,a,3\rangle, \langle 2,b,1\rangle, \langle 3,a,3\rangle, \langle 3,b,3\rangle\}$, $\boldsymbol{Q} = \{1,2,3\}$, $\boldsymbol{T} = \{a,b\}$, $q_s = \{1\}$, $\boldsymbol{F} = \{1,2\}$.

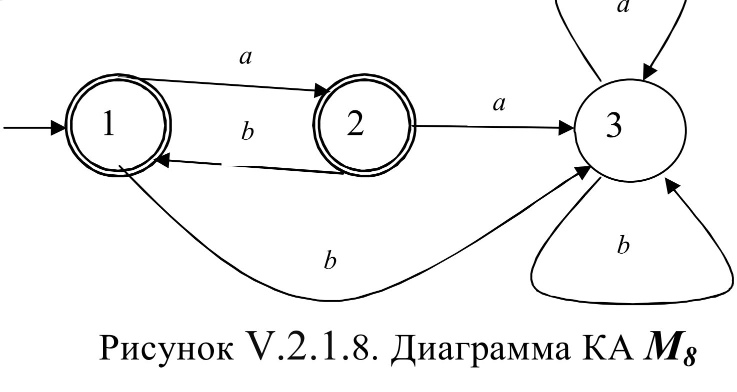


Рисунок V.2.1.8. Диаграмма КА $\boldsymbol{M}_8$

Задания V.2.1.

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 $(q,\ \tau)$, удовлетворяющие условию $(1,\ abaacdcc) \models^* (q,\ \tau)$, в КА $\boldsymbol{M_9}$ на рисунке V.2.1.9.

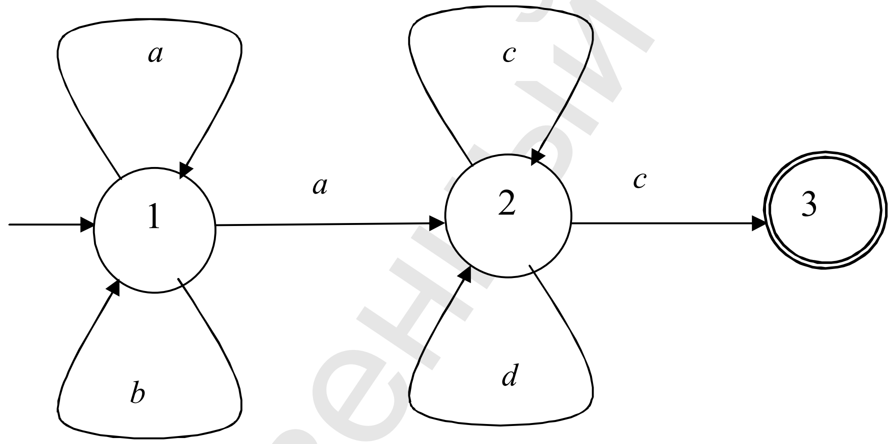


Рисунок V.2.1.9. Диаграмма КА $\boldsymbol{M_9}$.

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

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

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 V.2.1.

1. Является ли детерминированным КА ***M₁₀*** на рисунке V.2.1.10.

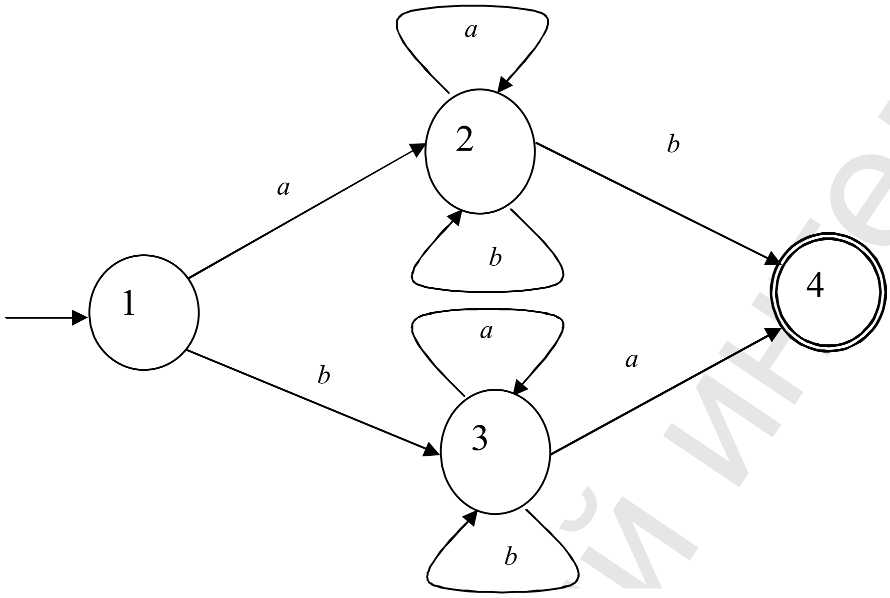


Рисунок V.2.1.10. Диаграмма КА $\boldsymbol{M_{10}}$.

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 $|\boldsymbol{Q}|$, $|\boldsymbol{T}|$, $|\Delta|$,$|\tau|$и число достижимых из $(q,\ \tau)$ 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 $\boldsymbol{M_{11}}$ с алфавитом $\boldsymbol{T} = \{a, b, c\}$ на рисунке V.2.1.11.

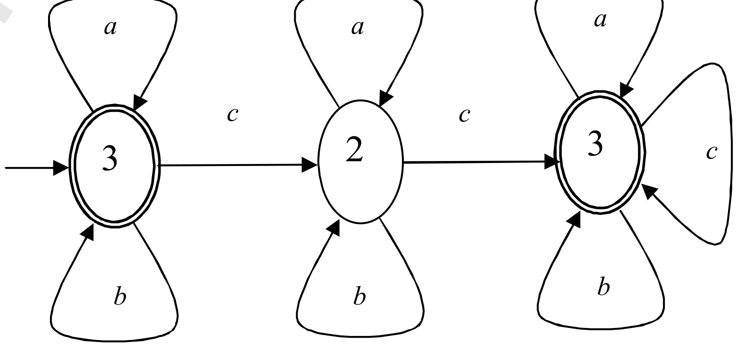


Рисунок V.2.1.11. Диаграмма КА $\boldsymbol{M_{11}}$.

11. Является ли полным детерминированный конечный автомат $\boldsymbol{M_{12}}$ с алфавитом $\boldsymbol{T} = \{a, b\}$ на рисунке V.2.1.12.

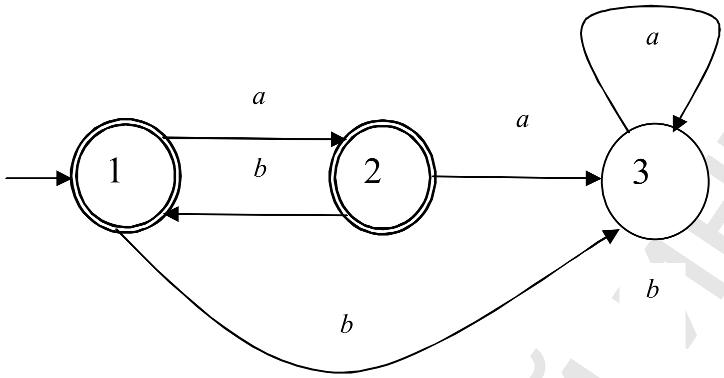


Рисунок V.2.1.12 Диаграмма КА $\boldsymbol{M_{12}}$.

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

$$\boldsymbol{G} = < \{+, -, *\}, \{A, B, C\}, \boldsymbol{P}, C > .$$

$$\boldsymbol{P} : A \rightarrow + \mid A +$$

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

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

Тесты V.2.1.

1. Конечный автомат переходит в то или иное состояние согласно:

A) таблице переходов в памяти автомата;

B) по данному заданию;

C) рисункам;

D) направлениям;

E) содержанию.

2. Какой автомат называется детерминированным??

A) 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;

B) 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;

C) Если распознаватель может иметь такую допустимую конфигурацию, для которой существует некоторое конечное множество конфигураций, возможных на следующем шаге работы;

D) Если распознаватель допускает чтение входных символов только в одном направлении («слева направо»);

E) 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 $\boldsymbol{M} = <Q, T, \delta, q_0, F>$, где Q

A) Конечное множество состояний;

B) Конечное множество допустимых входных символов;

C) Функцией переходов;

D) Начальное состояние;

E) 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. Meduna, Alexander (2014), Formal Languages and Computation: Models and Their Applications, CRC Press, p. 233. For more on this subject, see undecidable problem.
6. Ginsburg, Seymour (1975). Algebraic and automata theoretic properties of formal languages. North-Holland. pp. 8–9.
7. Akimov O.E. The discrete mathematics: logic, groups, graphs.- Moscow: The Laboratory of base knowledge, 2001.
8. Novikov F.A. The discrete mathematics for programmers: the Textbook for high schools. –St-Petersburg: Piter, 2008.
9. Meyer B., Baudoin C. Programming methods. In two parts: Part 2. Translate from French Y.A. Pervina - Moscow: Mir,1982.
10. http://www.regentsprep.org/regents/math/algebra/AP1/Interval
11. Хопкрофт, Дж., Мотвани, Р., Дж. Ульман. Введение в теорию автоматов, языков и вычислений. — М.: Вильямс, 2002 (пер. издания Addison Wesley). — 528 с. — ISBN 5-8459-0261-4
12. Harrison, Michael A. (1978). Introduction to Formal Language Theory. Reading, Mass.: Addison-Wesley Publishing Company.p.13.
13. Шарипбай А.А. Информатика, Алматы, Эверо, 2015, -313 с.
14. 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

PROMIS

ШАРИПБАЙ А.А.

МАТЕМАТИКА ДЛЯ КОМПЬЮТЕРНЫХ НАУК
(Учебное пособие)

Подписано к печати 29.05.2017 г. Бумага офсетная №1.
Формат 60Х84 1/16 Объем 9,5 усл.печ.л.
Тираж 50 экз. Заказ0168
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 Алтынбек 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.e-zerde.kz). Его научными интересами являются теоретические и прикладные проблемы информатики и информационных технологий: теория и технология 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 Летно-испытательском институте, г. Жуковск; «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 акционерных обществ г. Алматы; «Автоматизированная система учета коммунальных платежей» для Департамента коммунальных услуг и КСК 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 доктора PhD по группе специальностей «Информатика, вычислительная техника и управления». 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.