МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Самарский государственный аэрокосмический университет
имени академика С.П. Королева

Калентьев А.А.

Основы информатики

Учебное пособие

Самара –2002

ОГЛАВЛЕНИЕ.

Предисловие.

Последние 50 лет характеризуются бурным развитием информатики как науки. Современные компьютеры и существующие программные приложения позволяют решать очень широкий круг инженерных задач.

Бурное внедрение компьютеров и информационных технологий в промышленности не могло не сказаться на развитии учебного процесса в области информатики. Современный студент и будущий инженер с необходимостью должен владеть методами компьютерного проектирования. При этом он должен не только уметь работать в компьютерной среде, но и быть уверенным в правильности ее работы. Как охватить и осмыслить это огромное море информации, наступающей со всех сторон, в ограниченных временных рамках учебного процесса. Вывод один. Студенту надо дать базовые фундаментальные знания в области информатики, которые помогут ему в дальнейшем разобраться в этом безбрежном море информации.

В предлагаемом учебном пособии в качестве основ информатики выделяется два раздела: «данные» и «алгоритмы и функции». Это фундаментальные понятия, опирающиеся на основы математики.

Развитие новых информационных CALS-технологий подтверждает необходимость совершенствования базовой подготовки по информатике студентов конструкторских специальностей. В рамках CALS-технологии будущему инженеру-конструктору предстоит не только работать с разноформатными программными приложениями по исследованию предметной области, но и хранить данные в интегрированной базе данных в текстовом формате стандарта STEP, принимать и передавать информацию об изделии в синтаксически ориентированном формате. Сегодня уже недостаточно учить студента только умению делать расчет детали на компьютере, необходимо также уметь создавать электронное описание как самой детали, так и результатов расчета с последующей передачей их в интегрированную базу данных.

Умение формализовать предметную область, представить ее и расчетные данные синтаксически ориентированными средствами достигается на базе высокой математической и информационной культуры, т.е. информатика для будущего инженера-конструктора становится не только сопутствующим, но и одним из основных предметом.

Смысл совершенствования базовой подготовки состоит в углублении теоретических знаний. Если вчера на конструкторских специальностях акцент делался на практическое знание компьютера в смысле «как на нем работать», то сегодня надо знать компьютер в смысле «как он работает». Это видно на примере CAD/CAM систем: Autocad, Cadds5, Cimatron, Компас и т.д. Можно научиться работать и добиться совершенства на одной из этих систем. Однако они так часто меняются, что очень трудно пройти по жизни, работая в одной системе. Переход к другим системам может быть очень болезненным, если не знать общих принципов их работы. Недостаточно учить студента работе на клавиатуре без понимания общих принципов работы компьютера.

Предлагаемое учебное пособие является теоретическим введением в информатику. Ее изучение базируется на таких математических дисциплинах, как математическая логика, теория алгоритмов, теория вычислимых функций. В свою очередь, развитие прикладной (компьютерной) информатики явилось основой для таких дисциплин, как теория формальных грамматик и алгоритмических языков, теория типов данных и баз данных, широкое развитие получила теория реляционной алгебры.

Так как лекции по информатике читаются до лекций по математической логике и теории алгоритмов, то соответствующий материал используется в ограниченном объеме либо с даются соответствующие определения.

Пособие написано для студентов, обучающихся по конструкторским специальностям и от читателя на требуется дополнительных знаний, выходящих за пределы средней школы.

Введение.

Испокон века человечество стремилось научиться писать, читать и считать. Поэтому первое, что было изобретено человечеством в этом направлении был алфавит или набор знаков для общения.

Под словом писать мы понимаем возможность изобразить, например, на бумаге, некоторый объект в виде текста либо рисунка, полученного из наперед заданного алфавита.

Под словом читать мы понимаем возможность правильного понимания текста или рисунка, полученного из наперед заданного алфавита.

Под словом считать мы понимаем возможность, например, из двух текстов, описывающих разные объекты, полученных из одного алфавита, получить текст нового объекта, также из этого алфавита. При этом должно быть задано правило, по которому из первых двух объектов формируется новый объект. Важно подчеркнуть, что новый объект должен быть также представлен текстом в этом алфавите.

Понятие алфавит, объект, текст (изображение) объекта в заданном алфавите, правило, по которому формируется новый объект из первоначальных объектов; - становятся теми ключевыми понятиями, которые и образуют предмет информатики.

Информатика это, с одной стороны, наука, изучающая:

-способы синтаксического описания (представления или изображения) математических объектов;

-способы обработки и преобразования этих объектов в новые по правилам их семантического (содержательного) описания.

С другой стороны, это практика применения компьютеров как для синтаксического представления математических объектов, так и для способов преобразования этих объектов.

С методической точки зрения, информатика это стремление человечества научиться писать, читать и считать с помощью компьютера.

В соответствии с приведенными определениями и построено данное учебное пособие.

Первые два раздела посвящены стремлению научиться писать с помощью компьютера. Первый раздел посвящен описанию основных математических объектов, с которыми приходится иметь дело студенту на протяжении всего периода обучения и не только в курсе информатики. Понятие множества, отношения, отображения являются базовыми. Через них можно построить другие более сложные объекты. Если в первом разделе эти понятия определяются, то во втором разделе рассматриваются вопросы изображения этих объектов языковыми средствами, т.е. рассматриваются вопросы их синтаксического представления в заданном алфавите. Вводится определение алфавита, рассматриваются вопросы кодирования математических объектов словами из заданного алфавита, рассматриваются вопросы представления чисел в числовом алфавите или в системе счисления по основанию 10, либо по основанию р. После рассмотрения вопросов кодирования происходит переход к представлению как отдельных элементов множества, так и всего множества, как отдельных отношений, так и всего множества отношений, как отдельного отображения, так и всего множества отображений. Представление математических объектов языковыми средствами или в заданном алфавите определяет только наименование этого объекта и называется синтаксическим описанием объекта. Такое описание объекта не полно в том смысле, что одними и теми же языковыми средствами я могу обозначить элементы разных множеств: множеств целых либо вещественных чисел, либо символы русских букв и т.д. Поэтому каждый математический объект помимо синтаксического описания должен иметь свое содержательное описание. Только пара синтаксического и содержательного описаний объекта определяют его однозначно. Содержательным описанием объекта называется его функция интерпретации, рассматриваемая в следующем разделе.

Третий раздел направлен на реализацию стремления читать с помощью компьютера. Читать означает понимать однозначно написанное и понимать одинаково всеми читающими. При этом читателем может быть как студент, так и компьютер. И они должны одинаково понимать написанное. Исходя из этих соображений и вводится функция интерпретации, которая любому математическому объекту ставит в соответствие конкретную область определения его элементов. Это может быть числовая целочисленная либо вещественная либо символьная область определения всех элементов этого математического объекта.

Последние два раздела посвящены задаче научиться считать с помощью компьютера. Все вычисления на компьютере ведутся с помощью алгоритмов.

В математике существует гипотеза, что всякий алгоритм вычисляет некоторую функцию. И наоборот, для всякой вычисляемой функции существует алгоритм, вычисляющий ее значения. Поэтому к изучению алгоритмов можно подходить как со стороны механической процедуры их выполнения, так и со стороны вычислимых функций. В четвертом разделе рассматриваются элементы теории вычислимых функций, т.е. функций, которые по построению являются вычислимыми. Эта теория интересна как с точки зрения построения сложных вычислимых функций и доказательства существования для них алгоритмов, так и как самостоятельная теория, полезная для будущих инженеров – конструкторов. В ней новые вычислимые функции строятся из конечного набора заданных и интуитивно вычислимых функций с помощью конечного набора операций над этими функциями.

Алгоритм можно рассматривать как некую механическую процедуру преобразования одного или двух языковых описаний математических объектов в описание нового математического объекта на том же языке. При этом всякое такое преобразование существенно опирается на содержательное описание этих объектов. В последнем разделе рассматривается такая механическая процедура выполнения алгоритма. Называется она машиной Тьюринга, по имени ученого, который ее сконструировал. Принципы, заложенные в этой машине, определяют архитектуру современного компьютера. Студентам будет полезно ознакомиться с математической моделью обработки словарных (алфавитных) функций на примере машины Тьюринга. В пособии эта модель демонстрируется на примере выполнения операций сложения.

Отметим одну особенность изложения материала. В первом разделе мы вводим базовые математические объекты, даем их определение, способы представления, функцию интерпретации и т.д. Одновременно с этим для изложения основного материала мы пользуемся этими же понятиями. Так вопросы кодирования и функцию интерпретации мы определяем через понятие отображения, введенное в первом разделе как основной материал.

Задачи и упражнения в конце параграфов позволят глубже понять соответствующие разделы.

При изучении материала лучше всего придерживаться той последовательности изложения, в которой пособие напечатано.

1.Данные.

1.1.Множества.

Определение 1.1. Множеством называется совокупность объектов произвольной природы, наделенных общим свойством.

Например, множество студентов в группе, множество листов в тетради, множество ячеек памяти в компьютере, множество комплектующих изделий в самолете.

Множество элементов будем обозначать заглавными буквами английского либо русского алфавита. Элементы множества будем обозначать малыми буквами английского либо русского алфавита и писать их в фигурных скобках вида: $\{, \}$. Будем стремиться элементы множества обозначать теми же буквами, что и само множество.

Например, $A = \{a_1, a_2, a_3\}$ - множество, состоящее из трех элементов.

$B = \{b_1, b_2, b_3, \ldots, b_n\}$ - множество, состоящее из **n** элементов.

Множество $C = \{c_1, c_2, \ldots\}$ состоит из счетного или бесконечного числа элементов.

Элементами множества могут быть другие множества.

Например,

$M = \{m_1, P, m_2\} = \{m_1, \{p_1, p_2, p_3, \ldots, p_n\}, m_2\}$ – множество, состоящее из трех элементов, одним из которых является множество P, состоящее из n элементов. Случай, когда элементом множества M является само это множество, мы рассматривать не будем.

Для конечного множества A число его элементов называется мощностью множества и обозначается $|A|$.

Тот факт, что элемент a принадлежит множеству A, обозначается следующим образом: $a \in A$.

Определение 1.2. Два множества называются одинаковыми, если они состоят из одних и тех же элементов.

Например, одинаковыми являются множества A1 и A2 вида $A1 = \{a_1, a_2, a_3, a_4\}$, $A2 = \{a_1, a_2, a_3, a_2, a_4, a_1\}$.

Множество может состоять из одного элемента. При этом следует различать элемент **a** некоторого множества, множество **{a}**, состоящее из одного элемента **a**, множество **{{a}}**, состоящее из единственного элемента – множества **{a}**.

Определение 1.3. Пустым называется множество, не имеющее ни одного элемента. Все пустые множества по определению равны и обозначаются символом $\varnothing$**,** т. е. на множестве всех множеств имеется единственное пустое множество. Множество $\{\varnothing\}$, единственным элементом которого является пустое множество $\varnothing$, не пусто.

Определение 1.4.Множество A называется подмножеством множества B, если каждый элемент множества A является элементом множества B.

Символически это записывается в виде $A \subseteq B$. Согласно этому определению множество A и пустое множество $\varnothing$ являются подмножествами множества A.

Если для множеств A и B имеем $A \subseteq B$ и $B \subseteq A$, то будем считать множества A и B равными и писать $A = B$.

Всякое множество можно представить в виде совокупности его частей или подмножеств. Пустое множество $\varnothing$ имеет лишь одну часть - $\varnothing$**.** Множество $\{\varnothing\}$, состоящее из одного элемента, имеет две части: пустое множество $\varnothing$ и множество, состоящее из одного элемента $\{\varnothing\}$, т.е. множество всех частей множества, состоящего из одного элемента, состоит из двух элементов $\{\varnothing, \{\varnothing\}\}$. Множество $\{\varnothing, \{\varnothing\}\}$, состоящее из двух элементов, имеет четыре части $\varnothing$, $\{\varnothing\}$, $\{\{\varnothing\}\}$, $\{\varnothing,\{\varnothing\}\}$. В общем случае множество, состоящее из конечного числа **N** элементов, имеет 2^N различных частей.

1.1.1. Операции над множествами.

С помощью операций над множествами можно получать новые множества или новые объекты, новые данные. Операции над множествами в некотором смысле аналогичны операциям над числами.

Определение 1.5. Объединением множеств A и B (суммой множеств) называется множество $(A \cup B)$, элементами которого являются элементы множества A либо элементы множества B. Если некоторый элемент входит как во множество A, так и во множество B, то он входит и в их объединение.

Данная операция аналогична операции сложения чисел.

Например, для $A= \{a_1, a_2, a_3\}$ и $B= \{b_1, b_2, b_3, a_3\}$, $A \cup B= \{a_1, a_2, a_3, b_1, b_2, b_3\}$. Исходя из определения для любого множества A имеем $A \cup A = A$, $A \cup \varnothing = A$.

Определение 1.6. Пересечением множеств A и B (произведением множеств) называется множество $(A \cap B)$, элементами которого являются элементы, принадлежащие одновременно как множеству A, так и множеству B.

Данная операция аналогична операции произведения чисел.

Например, для $A= \{a_1, a_2, a_3\}$ и $B= \{b_1, b_2, b_3, a_3\}$, $A \cap B = \{ a_3 \}$. Исходя из определения для любого множества A имеем $A \cap A = A$, $A \cap \varnothing = \varnothing$.

Определение 1.7. Разностью множеств A и B называется множество (A / B), элементами которого являются элементы, принадлежащие множеству A и не принадлежащие множеству B.

Например, для $A= \{a_1, a_2, a_3\}$ и $B= \{b_1, b_2, b_3, a_3\}$, $A / B= \{a_1, a_2 \}$, $B / A = \{b_1, b_2, b_3\}$. Исходя из определения для любого множества A имеем $A / A = \varnothing$, $A / \varnothing = A$, $\varnothing$ / **A =** $\varnothing$**.** Все введенные операции над множествами являются бинарными, т.е. каждая операция выполняется над двумя операндами.

Множество A / B, полученное с помощью операции разности, называют еще дополнением множества B до множества A и обозначают $\overline{B}$. На практике дополнение множества строится относительно некоторого универсального, заданного заранее, множества, обозначаемого через U, т.е. $\overline{B} = U / B$.

Очень часто дополнение множества рассматривается как унарная операция – взятие дополнения множества B относительно некоторого универсального множества U. В следующем параграфе мы так и поступим. Будем рассматривать операцию взятие дополнения множества вместо операции разности множеств.

Задачи и упражнения.

1.Проверить на равенство множества:

$\{2, 4, 6\} = \{2, 4, 4, 4, 6, 6\}$;

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

$\{2, 4, 6\} = \{\{2, \{4, 4, 4\}, 6, 6\}$;

$\{\{1\}\}=\{1\}$.

2.Дано множество $A = \{1, 6\}$. Множество корней уравнения $x^2- 7x +6 = 0$ обозначим через B. Выполняется ли равенство A=B. Построить множество всех частей множества B.

3.Дано множество $A = \{1, 6\}$. Решение системы уравнений

$2x + y = 8$

$7x – y = 1$

обозначим через B. Выполняется ли равенство A=B. Построить множество всех частей множества B.

4 Проверить на равенство множества:

$A = \{a, b, c\}$, $B=\{a, \{b\}, c\}$, $C=\{a, \{b,\{c\}\}\}$, $D=\{\{a, b, c\}\}$.

5. Доказать: $\{1, 2, 3\}\neq \{\{1, 2, 3\}\}$.

6. Построить множество всех подмножеств для множества, состоящего из трех элементов.

7.Вычислить следующие выражения:

$\varnothing\cap\{\varnothing\}$;$\{\varnothing\}\cap\{\varnothing\}$; $\{\varnothing,\{\varnothing\}\}/ \varnothing$; $\{\varnothing,\{\varnothing\}\}/ \{\varnothing\}$;$\{\varnothing,\{\varnothing\}\}/ \{\varnothing\}$.

8.Доказать, что если A, B, C - такие подмножества множества U, что

$A \cap B \subseteq (U /C)$ и $A \cup C \subseteq B$, то $A \cap C = \varnothing$. Указание, см. рис.1.

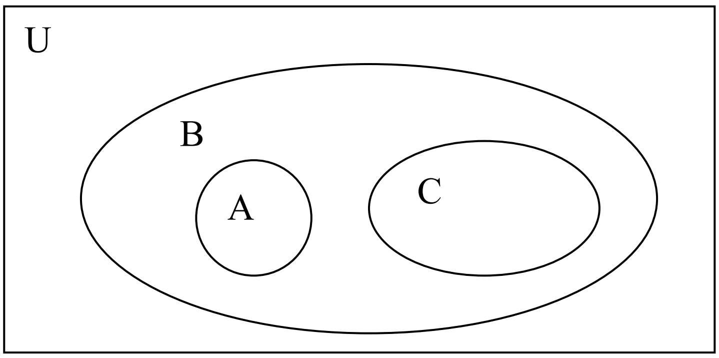

Рис. 1.

9.Доказать, что для произвольных множеств A, B, C выполняется равенство: $(A/B)/C = (A/C) /(B/C)$.

1.1.2.Алгебра множеств.

Мы рассмотрели понятие множества и операции над множествами. Рассмотрим некоторое универсальное множество U и совокупность всех его подмножеств вместе с введенными операциями. Оказывается, такая совокупность подчиняется некоторым правилам –тождествам или законам, независимо от того, какое универсальное множество мы выбрали. Такая совокупность называется алгеброй множеств.

Дадим строгое определение алгебры множеств. Дано универсальное множество U и множество всех его подмножеств 2^U. Даны операции над множествами: объединение ($\cup$), пересечение ($\cap$), дополнение ($\overline{\ \ }$) и отношение включения ($\subseteq$) между множествами. Совокупность вида

$A(U) = < 2^U, \ \cup, \cap, \ \overline{\ \ }\ , \ \subseteq >$

называется алгеброй множеств на универсальном множестве U.

Элементом такой алгебры является всякое подмножество множества U. Эти элементы могут быть связаны отношением включения. На множестве этих элементов вводятся операции, с помощью которых можно из одних множеств строить другие в рамках универсального множества U. В некотором смысле алгебра множеств представляет собой теоретико-множественный аналог алгебры действительных чисел с операциями сложения (+), умножения (*) и отношением порядка (≥) над числами.

Алгебра множеств задается совокупностью равенств относительно операций над множествами. Выпишем их:

1. $A\cup(B\cup C)= (A\cup B)\cup C$ ассоциативный закон относительно операции объединения;	1. $A\cap(B\cap C)= (A\cap B)\cap C$ ассоциативный закон относительно операции пересечения;
2. $A\cup B = B\cup A$ коммутативный закон относительно операции объединения;	2. $A\cap B = B\cap A$ коммутативный закон относительно операции пересечения;
3. $A\cup(B\cap C)= (A\cup B)\cap(A\cup C)$ дистрибутивный закон операции объединения относительно операции пересечения;	3. $A\cap(B\cup C)= (A\cap B)\cup(A\cap C)$ дистрибутивный закон операции пересечения относительно операции объединения;
4. $A \cup \varnothing = A$;	4. $A \cap U = A$;
5. $A \cup \overline{A} = U$;	5. $A \cap \overline{A} = \varnothing$;
6. Если для всех A справедливо $A \cup B = A$, то $B = \varnothing$.	6. Если для всех A справедливо $A\cap B = A$, то $B = U$.
7. Если $A \cup B = U$ и $A \cap B = \varnothing$, то $B = \overline{A}$;	7. Если $A \cup B = U$ и $A \cap B = \varnothing$, то $B = \overline{A}$;
8. $\overline{(\overline{A})} = A$ дополнение к дополнению множества A равно множеству A;	8. $\overline{(\overline{A})} = A$ дополнение к дополнению множества A равно множеству A;
9. $\overline{\varnothing} = M$;	9. $\overline{U} = \varnothing$;
10. $A \cup A = A$ закон идемпотентности;	10. $A \cap A = A$ закон идемпотентности;
11. $A \cup U = U$;	11. $A \cap \varnothing = \varnothing$;
12. $A \cup (A \cap B) = A$;	12. $A \cap (A \cup B) = A$;
	13. $\overline{(A \cap B)} = \overline{A} \cup \overline{B}$.

13. $\overline{(A \cup B)} = \overline{A} \cap \overline{B}$.	

Ассоциативный закон по операции объединения говорит о том, что в каком бы порядке мы ни проводили объединение множеств, все равно получится один и тот же результат. И еще он говорит о том, что с помощью бинарной операции объединения можно объединить n – множеств. Это же распространяется и на ассоциативный закон операции пересечения.

Коммутативный закон относительно любой операции говорит о том, что от перестановки мест слагаемых результат не изменится.

Справедливость каждого из равенств можно проверить, показав, что множество, стоящее по левую сторону равенства включено во множество, стоящее по правую сторону знака равенства и, наоборот, множество, стоящее по правую сторону равенства включено во множество, стоящее по левую сторону знака равенства. В качестве примера докажем равенство 3.

Сначала докажем истинность включения $A\cup(B\cap C) \subseteq (A\cup B)\cap(A\cup C)$.

Пусть $x \in A\cup(B\cap C)$. Так как x принадлежит объединению множеств A и $(B\cap C)$, то , по определению операции объединения, $x \in A$ либо $x \in (B\cap C)$. Если $x \in A$, то x может одновременно принадлежать $(A\cup B)$ и $(A\cup C)$, т.е. $x \in (A\cup B)$ и $x \in (A\cup C)$. Следовательно $x \in (A\cup B) \cap (A\cup C)$. Включение доказано.

Докажем теперь истинность включения $(A\cup B)\cap(A\cup C) \subseteq A\cup(B\cap C)$.

Пусть $x \in (A\cup B)\cap(A\cup C)$. Так как x принадлежит пересечению множеств $(A\cup B)$ и $(A\cup C)$, то, по определению операции пересечения, $x \in (A \cup B)$ и одновременно $x \in (A\cup C)$. Если $x \in (A\cup B)$, то $x \in A$, либо $x \in B$. И одновременно с этим $x \in (A\cup C)$, т.е. $x \in A$ либо $x \in C$. Это означает, что x может одновременно принадлежать A и C, либо A и B, либо B и C. Из последнего вытекает, что $x \in (B\cap C)$. Если $x \in (B\cap C)$, то, по определению операции объединения, справедливо $x \in A\cup(B\cap C)$. Включение доказано.

Мы доказали включение в обе стороны, следовательно, равенство 3 доказано. Аналогично доказываются и другие равенства.

С помощью равенств алгебры множеств можно упрощать различные сложные выражения подобно тому, как это делается в алгебре чисел. Рассмотрим такие преобразования на примерах. Упростить выражения:

Пример1. $(\overline{(A \cap \overline{B})}) \cup B$

$\overline{(A \cap \overline{B})} \cup B = (\overline{A} \cup \overline{(\overline{B})}) \cup B$ (согласно равенства 13 для операции пересечения);

$(\overline{A} \cup \overline{(\overline{B})}) \cup B = (\overline{A} \cup B) \cup B$ (согласно закона двойного дополнения);

$(\overline{A} \cup B) \cup B = \overline{A} \cup B$ (согласно закона идемпотентности).

Пример2. $(A\cap B\cap C)\cup(\overline{A} \cap B \cap C) \cup \overline{B} \cup \overline{C}$

$(A\cap B\cap C)\cup(\overline{A} \cap B \cap C) \cup \overline{B} \cup \overline{C} = ((A\cup \overline{A}) \cap (B\cap C)) \cup \overline{B} \cup \overline{C}$ (согласно равенства 3 для операции пересечения);

$((A\cup \overline{A}) \cap (B\cap C)) \cup \overline{B} \cup \overline{C} = (M \cap (B\cap C)) \cup \overline{B} \cup \overline{C}$ (согласно равенства 11 для операции объединения);

$(M \cap (B \cap C)) \cup \overline{B} \cup \overline{C} = (B \cap C) \cup \overline{B} \cup \overline{C}$ (согласно равенства 4 для операции пересечения);

$(B \cap C) \cup \overline{B} \cup \overline{C} = (B \cap C) \cup \overline{(B \cap C)}$ (согласно равенства 13 для операции пересечения);

$(B \cap C) \cup \overline{(B \cap C)} = M$ (согласно равенства 5 для операции объединения).

В алгебре множеств имеется своя теория решения уравнений, отличающаяся от решения алгебраических уравнений.

Задачи и упражнения.

1.Доказать тождества 3, 4, 5 для операции пересечения.

2.Доказать тождества:

$(A \cap B \cap X) \cup (A \cap B \cap C \cap X \cap Y) \cup (A \cap X \cap \overline{A}) = A \cap B \cap X$;

$(A \cap B \cap C) \cup (\overline{A} \cap B \cap C) \cup \overline{B} \cup \overline{C} = U$;

$(A \cap B \cap C \cap \overline{X}) \cup (\overline{A} \cap C) \cup (\overline{B} \cap C) \cup (C \cap X) = C$.

1.1.3.Множества с повторениями.

Каждый элемент множества с повторениями может появляться в этом множестве несколько раз. Число таких вхождений одного элемента называется его кратностью. Множества с повторениями называют еще мультимножествами. В дальнейшем будем пользоваться этим наименованием.

Мультимножество на множестве А – это пара $MA = <A, k(a)>$, где

А – множество элементов;

$k: A \to N_0$ – функция кратности элементов множества А.

Например, пара $A = \{a_1, a_2, a_3\}$, $k(a) = \{2, 3, 1\}$ образует мультимножество $MA = \{2* a_1, 3*a_2, 1*a_3\} = \{a_1, a_1, a_2, a_2, a_2, a_3\}$. В мультимножестве порядок элементов не существенен. Существенна только кратность элементов.

Два мультимножества называются одинаковыми, если они состоят из одних и тех же элементов и их функции кратности совпадают. По данному определению мультимножества $MA = \{a_1, a_1, a_2, a_2, a_2, a_3\}$ и $MB = \{a_1, a_2, a_3\}$ называются разными.

Будем говорить, что мультимножество МВ есть подмножество МА и писать $MB \subseteq MA$, если кратность каждого элемента в МВ не больше кратности этого элемента в МА.

Пусть $MA = <A, k(a)>$, $A = \{a_1, a_2, \dots. a_r\}$, $k(a) = \{k_1, k_2, \dots. k_r\}$, $k_i = 1, 2, .., k_i$, для всех $i = 1, 2, \dots, r$.

Мощностью мультимножества МА будем называть число $|MA| = k_1 + k_2 + \dots. + k_r$.

Как само мультимножество, так и каждое его подмножество МВ можно однозначно задать упорядоченной последовательностью длины r чисел:

$<m_1, m_2, \dots., m_r>$, $0 \le m_i \le k_i$, $i = 1, 2,\dots, r$.

Здесь m_i означает кратность элемента a_i в МВ. Если $m_i = 0$, то это означает, что элемент a_i не входит в подмножество МВ. Отсюда следует, что число всех подмножеств МА равно $|MB| = (k_1+1) * (k_2 + 1) * \dots. * (k_r+1)$.

1.2.Отношения.

Определение 1.8. Декартовым произведением $A \times B$ множеств A и B называется множество всех пар вида (a, b), где $a \in A$, $b \in B$.

Из этого определения следует, что пара (b, a) не принадлежит декартовому произведению $A \times B$.

Если A и B – конечные множества и A содержит m элементов, а B содержит n элементов, то их декартово произведение будет содержать (m*n) пар.

$A \times B = \{(a_i, b_j), a_i \in A, i=1, 2, \ldots, m; b_j \in B, j=1, 2, \ldots, n\}$.

Определение 1.9. Всякое подмножество α декартова произведения $A \times B$ называется отношением. $\alpha \subseteq A \times B$.

Если пара $(a, b) \in \alpha$, то будем говорить, что элемент a находится в отношении α с элементом b или, что отношение α истинно для (a, b). Этот факт иногда записывают следующим образом: $a \alpha b$, т.е. элементы a и b связаны отношением α.

Так как отношением на множестве $A \times B$ является всякое его подмножество, то число всех отношений есть множество всех подмножеств множества $A \times B$, или мощность множества всех отношений равна 2^{m*n}.

Декартово произведение можно построить на одном множестве A. Для этого образуем множество, являющееся степенью множества A: множество A^2 или $A \times A$ есть множество кортежей длины 2; $A^3 = A \times A \times A$ – множество кортежей длины 3; …; A^n - множество кортежей длины n. Договоримся, под множеством A нулевой степени A^0 понимать множество, состоящее из одного элемента – пустого множества $A^0 = \{\varnothing\}$.

Отношение, заданное на декартовом произведении $A \times A$ двух множеств A называется бинарным отношением, заданным на множестве A.

Отношение, заданное на декартовом произведении n множеств, называется n – арным отношением.

Примером унарного отношения, заданного на множестве A, является отношение принадлежности ($\in$) элемента множеству A.

Задачи и упражнения.

1.Построить декартово произведение $A \times B \times C$ для множеств $A =\{a_1, a_2\}$, $B=\{b_1, b_2, b_3\}$, $C=\{c_1, c_2\}$.

2. Выписать все бинарные отношения на множестве $A =\{a_1, a_2\}$.

3.Построить все отношения на декартовом произведении $A \times B$ множеств $A =\{a_1, a_2\}$, $B=\{b_1, b_2\}$.

4.Построить все унарные отношения на множестве $A =\{a_1, a_2, a_3\}$.

5. Подсчитать число всех n – арных отношений, заданных на множестве A, состоящем из m элементов.

1.2.1.Операции над отношениями.

Поскольку отношения, заданные на фиксированной паре множеств A и B суть подмножества декартова произведения A × B, то совокупность всех этих отношений образует алгебру отношений относительно операций объединения, пересечения и дополнения.

Пример. $A=\{a_1, a_2\}$, $B=\{b_1, b_2, b_3\}$.

$\alpha 1=\{(a_1, b_1), (a_1, b_3), (a_2, b_3)\}$,

$\alpha 2=\{(a_2, b_1), (a_2, b_2), (a_2, b_3)\}$, тогда

$\alpha 1 \cup \alpha 2=\{(a_1, b_1), (a_1, b_3), (a_2, b_3), (a_2, b_1), (a_2, b_2)\}$,

$\alpha 1 \cap \alpha 2=\{(a_2, b_3)\}$,

$\alpha 1/\alpha 2=\{(a_1, b_1), (a_1, b_3)\}$.

Помимо этих операций на множестве отношений важное значение имеют еще две операции: обращение отношений и умножение отношений.

Пусть на декартовом произведении A × B задано отношение α.

Определение 1.10. Обратным α^{-1} к отношению $\alpha \subseteq A \times B$ называется отношение α^{-1}, определенное на декартовом произведении B × A и состоящее из всех тех пар $(b, a) \in B \times A$, для которых истинно $(a, b) \in \alpha$.

$\alpha^{-1} = \{(b, a) \mid (a, b) \in \alpha.\}$.

Пример. Для отношения $\alpha 1$ из предыдущего примера обратным будет отношение $\alpha 1^{-1} \subseteq B \times A$: $\alpha 1^{-1} =\{(b_1, a_1), (b_3, a_1), (b_3, a_2)\}$,

Пусть заданы отношения $\alpha \subseteq A \times B$ и $\beta \subseteq B \times C$.

Определение 1.11. Произведением отношений α и β или композицией называется отношение $\alpha\beta$, определенное на декартовом произведении A × C и такое, что $(a, c) \in \alpha\beta$ истинно тогда и только тогда, когда (т.т.т.к.) найдется такой элемент $b \in B$, для которого истинно $(a, b) \in \alpha$ и $(b, c) \in \beta$.

Пример. $A=\{a_1, a_2\}$, $B=\{b_1, b_2, b_3\}$, $C=\{c_1, c_2\}$.

$\alpha=\{(a_1, b_1), (a_1, b_3), (a_2, b_3)\}$,

$\beta=\{(b_1, c_1), (b_1, c_2), (b_3, c_1)\}$, тогда

$\alpha\beta =\{(a_1, c_1), (a_1, c_2), (a_2, c_1)\}$, $\alpha\beta \subseteq A \times C$.

Задачи и упражнения.

1. Построить отношение α^{-1}, обратное к отношению $\alpha \subseteq A \times B$, где $A=\{a_1, a_2\}$, $B=\{b_1, b_2, b_3\}$ и $\alpha=\{(a_1, b_1), (a_1, b_3), (a_2, b_2), (a_2, b_3)\}$.
2. Построить композицию отношений $\alpha\alpha^{-1}$, $\alpha^{-1}\alpha$, взятых из предыдущей задачи.

3. Дано: декартово произведение множеств $A \times B$, где $A = \{a_1\}$, $B=\{b_1, b_2, b_3\}$. На множестве $A \times B$ построить множество всех бинарных отношений.
4. Дано: декартово произведение множеств $A \times B \times C$, где $A = \{a_1\}$, $B=\{b_1, b_2, b_3\}$, $C=\{c_1\}$. На множестве $A \times B \times C$ построить множество всех бинарных отношений.
5. Доказать, что

$(A \cap B) \times (C \cap D) = (A \times C) \cap (B \times D)$;

$(A \cup B) \times C = (A \times C) \cup (B \times C)$;

$A \times (B \cup C) = (A \times B) \cup (A \times C)$.

6. Найти α^{-1}, $\alpha\alpha$, $\alpha\alpha^{-1}$, $\alpha^{-1}\alpha$ для отношения α:

a1) $\alpha = \{(x, y) \mid x \in N, y \in N$ и x делит $y\}$,

a2) $\alpha = \{(x, y) \mid x \in R, y \in R$ и $x + y =< 0\}$. R – множество действительных чисел.

7. Доказать, что

a1) $(\alpha_1 \cup \alpha_2)^{-1} = \alpha_1^{-1} \cup \alpha_2^{-1}$;

a2) $(\alpha_1 \cap \alpha_2)^{-1} = \alpha_1^{-1} \cap \alpha_2^{-1}$;

a3) $(\alpha_1\alpha_2)^{-1} = \alpha_2^{-1}\alpha_1^{-1}$.

1.2.2.Свойства отношений.

Введем некоторые часто употребляемые отношения на множестве $A \times A$:

-отношение, называемое диагональю, $\tau_A = (a_i, a_i)$ для всех $i = 1,\dots, n$.

-полное отношение $\Delta_A = (a_i, a_j)$ для всех пар (i, j).

Определение 1.12.Бинарное отношение α называется рефлексивным, если в него входит единичное отношение, т.е. $\tau_A \subseteq \alpha$.

Определение 1.13.Отношение α называется симметричным, если из того факта, что $(a, b) \in \alpha$, следует, что и $(b, a) \in \alpha$.

Определение 1.14. Отношение α называется транзитивным, если из того факта, что $(a, b) \in \alpha$ и $(b, c) \in \alpha$ следует, что и $(a, c) \in \alpha$.

На множестве $A = \{a, b, c\}$ $\tau_A = \{(a, a), (b, b), (c, c)\}$, $\Delta_A = \{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c), (c, a), (c, b), (c, c)\}$.

Пример транзитивного отношения на A: $\alpha = \{(a, a) (a, b) (a, c) (b, a) (c, a))$.

Определение 1.15. Отношение α называется антисимметричным, если выполняется включение $\alpha \cap \alpha^{-1} \subseteq \tau_A$.

Определение 1.16. Отношение α называется антирефлексивным, если из того, что $(a, b) \in \alpha$ следует, что $a \neq b$ или $\alpha \cap \tau_A = \varnothing$, т.е. отношение α выполняется только для несовпадающих элементов.

1.2.3.Отношение частичного порядка.

Определение 1.17. Бинарное отношение α, наделенное свойствами рефлексивности, антисимметричности и транзитивности, называется отношением частичного порядка.

Пример. На множестве $A= \{a_1, a_2, a_3\}$ рассмотрим множество всех его подмножеств $2^A =\{\varnothing, \{a_1\}, \{a_2\}, \{a_3\},\{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}\}$. На множестве 2^A введем отношение частичного порядка Δ по включению $\subseteq$.

$\Delta = \{ (\varnothing,\varnothing), (\varnothing, a_1), (\varnothing, a_2), (\varnothing, a_3), (\varnothing,\{a_1, a_2\}), (\varnothing,\{a_1, a_3\}), (\varnothing,\{a_2, a_3\}), (\varnothing,\{a_1, a_2, a_3\}), (\{a_1\}, \{a_1\}), (\{a_1\},\{a_1, a_2\}), (\{a_1\},\{a_1, a_3\}), (\{a_1\},\{a_1, a_2, a_3\}), (\{a_2\}, \{a_2\}), (\{a_2\},\{a_1, a_2\}), (\{a_2\},\{a_2, a_3\}), (\{a_2\},\{a_1, a_2, a_3\}), (\{a_3\}, \{a_3\},), (\{a_3\},\{a_2, a_3\}), (\{a_3\},\{a_1, a_3\}), (\{a_3\},\{a_1, a_2, a_3\}), (\{a_1, a_2\}, \{a_1, a_2\}), (\{a_1, a_2\}, \{a_1, a_2, a_3\}), (\{a_1, a_3\}, \{a_1, a_3\}), (\{a_1, a_3\}, \{a_1, a_2, a_3\}), (\{a_2, a_3\}, \{a_2, a_3\}), (\{a_2, a_3\}, \{a_1, a_2, a_3\}), (\{a_1, a_2, a_3\}, \{a_1, a_2, a_3\})\}$.

1.2.4.Отношение эквивалентности.

Отношение α называется отношением эквивалентности, если оно рефлексивно, симметрично и транзитивно. Т.е. выполняется:

-$(a, a) \in \alpha$, для любого $a \in A$ (рефлексивность);

-из $(a, b) \in \alpha$ следует, что и $(b, a)\in \alpha$ (симметричность);

-из $(a, b) \in \alpha$ и $(b, c) \in \alpha$ следует, что $(a, c) \in \alpha$ (транзитивность).

Пример. На множестве $A= \{a_1, a_2, a_3, a_4, a_5\}$ зададим отношение эквивалентности θ в виде таблицы на декартовом произведении $A\times A$:

a_5	*			*	*
a_4				*	*
a_3		*	*		
a_2		*	*		
a_1	*				*
	a_1	a_2	a_3	a_4	a_5

В данном примере множество A разбито на два класса эквивалентности

$\theta =\{(a_1, a_4, a_5), (a_2, a_3)\}$. Самостоятельно выписать все пары, входящие в отношение θ.

Задачи и упражнения.

1.Доказать, что если отношения α_1, α_2 рефлексивны, то рефлексивны отношения: $\alpha_1 \cup \alpha_2, \alpha_1 \cap \alpha_2, \alpha_1^{-1}, \alpha_1\alpha_2$.

2.Доказать, что если отношения α_1, α_2 симметричны, то симметричны отношения: $\alpha_1 \cup \alpha_2, \alpha_1 \cap \alpha_2, \alpha_1^{-1}, \alpha_1\alpha_2$.

3.На множестве $N \times N$ определим следующее бинарное отношение:

$<(a, b), (c, d)> \in \alpha$ т.т.т.к. выполняется условие: $a+d = b+c$. Доказать, что α является отношением эквивалентности.

4. Пусть A – множество всех прямых на плоскости. Являются ли отношениями эквивалентности следующие отношения:

-отношение, содержащее параллельные прямые;

-отношение, содержащее перпендикулярные прямые.

5. На множестве R – действительных чисел определим отношение α следующим образом: $(a, b) \in \alpha$, т.т.т.к. $(a - b)$ – рациональное число.

6. Доказать, что если α есть отношение эквивалентности, то и α^{-1} также есть отношение эквивалентности.

1.3.Отображения.

Определение 1.18. Отношение α, определенное на паре множеств A, B называется отображением из A в B, если для каждого $a \in A$ существует один и только один элемент $b \in B$, удовлетворяющий отношению $(a, b) \in \alpha$ [5]. Элемент $b \in B$ называется образом элемента $a \in A$ при отображении α. Элемент $a \in A$ называется прообразом элемента $b \in B$ при отображении α. Совокупность всех прообразов элемента $b \in B$ в A называется полным прообразом этого элемента в A и обозначается $f^{-1}(b)$. Для отображения f из A в B будем использовать следующую запись:

$f: A \to B$, или $b = f(a)$.

Задать отображение из множества A в множество B означает указать правило, которое каждому элементу $a \in A$ ставит в соответствие единственный элемент $b \in B$, называемый значением отображения.

Самый простой способ задания отображения заключается в задании всех пар вида (a, b), где $a \in A$, $b \in B$, образующих отображение $f: A \to B$. Например, пусть $A = \{a_1, a_2, a_3, a_4\}$, $B =\{b_1, b_2, b_3\}$. Зададим отображение следующим образом $f: \{(a_1, b_3), (a_2, b_2), (a_3, b_2), (a_4, b_1)\}$.

Другой способ задания отображения – путем задания арифметического выражения, определяющего зависимость значения отображения от значения элемента из области определения. Например, рассмотрим функцию следования Пеано $\sigma: N \to N$. По определению функция Пеано каждому натуральному числу n ставит в соответствие число $(n+1)$. Запись функции Пеано в виде арифметического выражения имеет вид $\sigma(n) = n + 1$.

Пусть $A' \subseteq A$ –подмножество множества A. Образом множества A' при отображении f назовем объединение образов $f(a)$, для всех $a \in A'$ и будем обозначать его $f(A')$. Естественно, что выполняется включение $f(A') \subseteq B$. В частности, $f(A)$ есть образ всего множества A.

Аналогично, если $B' \subseteq B$, то полным прообразом множества B' (обозначается через $f^{-1}(B')$) называется объединение прообразов всех элементов, входящих в B'. Естественно, что выполняется включение $f^{-1}(B') \subseteq A$. В частности, $f^{-1}(B)$ есть образ всего множества B.

Определим так называемое единичное отображение на множестве A:

$\tau_A : A \to A$, которое каждому элементу $a \in A$ сопоставляет этот же самый элемент. Легко видеть, что единичное отображение - это тоже самое, что и диагональное отношение на A.

Каждому отображению соответствует его арность. Введенное выше отображение $f: A \to B$ называется унарным, т.е. это такое отображение, которое отображает на множество B элементы одного множества, множества A.

Отображение вида $f: A_1 \times A_2 \to B$ называется бинарным или обладает арностью, равной двум. Бинарное - это такое отображение, которое отображает на множество B элементы двух множеств, A_1 и A_2. Область определения бинарного отображения задается на декартовом произведении двух множеств. Бинарное отображение называют еще функцией от двух переменных.

В общем случае n – арным отображением называется отображение вида

$f: A_1 \times A_2 \times \ldots \times A_n \to B$, заданное на декартовом произведении n множеств. Как правило арность отображения либо не указывается, если из контекста понятно о какой арности идет речь, либо указывается в виде верхнего индекса функционального символа f. n- арное отображение называют еще функцией от n переменных.

Например, $f^{(n)}: A^n \to B$ или $b = f(a_1, a_2, \ldots, a_n)$.

При n=0 можно говорить о нуль-арной функции. Нуль – арной называется функция, заданная на множестве A^0 или на пустом множестве. Нуль – арная функция может быть либо всюду определенной (на пустом множестве), либо нигде не определенной. Нуль – арные функции определяются как константные функции. Возьмем произвольное число $x \in N$. Через x^0 обозначим нуль- арную функцию, тождественно равную элементу $x \in N$. В частности, $0^0 = 0$, $1^0 = 1$ и т.д.

Из n –арной функции $f^{(n)}: A^n \to B$ при фиксировании одного i-го аргумента получается (n - 1)- арная функция. Например, рассмотрим функцию от двух переменных $f^2: A^2 \to B$ или $f(x, y)$. Зафиксируем прямую вида $y = y_0$. Тогда, для любого $x \in A$ существует функция $f(x, y_0) \in B$. Будем говорить, что функция f от двух переменных индуцирует функцию g от одной переменной:

$g(x) = f(x, y_0)$.

Пусть даны отображения $f: A \to B$ и $g: B \to C$. Тогда $f \bullet g : A \to C$ есть произведение отображений f и g, и является отображением из A в C.

Пусть $f: A \to B$; отображение $f^{-1} : B \to A$ называется обратным к f , если выполняются равенства: $f \bullet f^{-1} = \tau_A$, $f^{-1} \bullet f = \tau_B$. Это означает, что отображение f^{-1} переводит любой образ $f(a)$ в a, а отображение f переводит любой прообраз $f^{-1}(b)$ в b.

Пусть мощность множества A равна $|A| = n$, а мощность множества B - $|B| = m$. Тогда число всех отображений $f: A \to B$ равно

$\mid f: A \to B \mid = m^n$.

Действительно, всякое отображение b = f (a) можно представить вектором длины n вида $(b_1, b_2, \ldots, b_n)$. В этом векторе всякая i-я компонента может принять любое значение из m. Тогда, число всех таких векторов или число всех отображений есть m^n.

Задачи и упражнения.

1.Установить взаимно однозначное соответствие или построить функцию между множествами:

а) $(A\times B)$ и $(B\times A)$;

б) $(A\times B)^C$ и $(A^C\times B^C)$;

в) $(A^B)^C$ и $A^{(B\times C)}$.

2. Пусть A, B, A_1, B_1 – такие множества, что между A и A_1 и B и B_1 существуют взаимно однозначные соответствия: φ: $A \to A_1$, φ_1: $B \to B_1$. Установить взаимно однозначное соответстиве между:

а) $A\times B$ и $A_1\times B_1$;

б) A^B и A_1^{B1};

в) $A\cup B$ и $A_1 \cup B_1$, если $A\cap B= \varnothing$ и $A_1 \cap B_1 = \varnothing$.

3.Доказать, что для любой функции ф выполняются равенства:

а) $f(A \cup B) = f(A) \cup f(B)$;

б) $f^{-1}(A \cup B) = f^{-1}(A) \cup f^{-1}(B)$;

в) $f(A) / f(B) \subseteq f(A / B)$.

1.3.1.Свойства отображений.

Определение 1.19. Отображение f множества A в B называется сюръективным или отображением A на все множество B, если каждый элемент $b \in B$ имеет в A хотя бы один прообраз, т.е. если уравнение f (x) = b для любого b имеет хотя бы одно решение $x \in A$.

Пример сюръективного отображения: $A = \{a_1, a_2, a_3, a_4, a_5\}$, $B =\{b_1, b_2, b_3\}$, $f:A\to B$: $\{(a_1, b_1), (a_2, b_1), (a_3, b_2), (a_4, b_3), (a_5, b_1)\}$; или это можно записать так:

$f(a_1) = b_1$, $f(a_2) = b_1$, $f(a_3) = b_2$, $f(a_4) = b_3$, $f(a_5) = b_1$.

Определение 1.20. Отображение f множества A в B называется инъективным, если для любого элемента $b \in B$ его полный прообраз $f^{-1}(b)$ содержит не более одного элемента, т.е. для любых a_1, a_2 $f(a_1) \neq f(a_2)$.

Пусть $|A| = n$, $|B| = m$. Тогда число инъективных отображений равно

$|f_{ин}| = m*(m-1)*(m-2)*\ldots*(m-n+1)$.

Действительно, значение первой компоненты b_1 вектора функции мы выбираем m возможными способами. Вторая компонента b_2 выбирается на множестве (m-1) значений, так как одно значение уже занято первой компонентой. И т.д. Последняя компонента b_n может быть выбрана из оставшихся (m-n+1) значений.

Пример инъективного отображения: $A = \{a_1, a_2, a_3\}$, $B =\{b_1, b_2, b_3, b_4, b_5\}$, $f:A\to B$: $\{(a_1, b_1), (a_2, b_3), (a_3, b_2)\}$; или это можно записать так:

$f(a_1) = b_1$, $f(a_2) = b_3$, $f(a_3) = b_2$.

Определение 1.21. Отображение f множества A в B называется биективным, если оно одновременно сюръективно и инъективно.

Из сюръективности следует, что $|f^{-1}(b)| \geq 1$ для $\forall b \in B$. Из инъективности следует, что $|f^{-1}(b)| \leq 1$ для $\forall b \in B$. Тогда говорят, что биективное отображение f устанавливает взаимно – однозначное соответствие между множествами A и B. Когда множества A и B конечны, это означает, что их мощности равны. $|A| = |B|$. Число биективных отображений равно $|f\colon A \to B| = n!$, так как $|A| = |B| = n$.

Пример биективного отображения: $A = \{a_1, a_2, a_3, a_4, a_5\}$, $B = \{b_1, b_2, b_3, b_4, b_5\}$, $f\colon A \to B$: $\{(a_1, b_1), (a_2, b_5), (a_3, b_2), (a_4, b_4), (a_5, b_3)\}$; или это можно записать так:

$f(a_1) = b_1$, $f(a_2) = b_5$, $f(a_3) = b_2$, $f(a_4) = b_4$, $f(a_5) = b_3$.

Определение 1.22. Два отображения $f\colon A \to B$ и $\varphi\colon A' \to B'$ равны, если $A = A'$, $B = B'$ и существует соответствие $\theta\colon A \to A'$ такое, что $f(a) = \varphi(\theta(a))$ для $\forall a \in A$.

Определение 1.23. Отображение вида $f\colon A \to A$ называется преобразованием или операцией на множестве A.

Операция называется n- арной, если отображение имеет вид $f : A^n \to A$, если ее область определения задана на декартовом произведении n множеств. Здесь число n называется также местностью операции.

Введем отображение, называемое предикатной функцией. $p\colon X \to \{0, 1\}$.

$$p(x) = \begin{cases} 1, & \text{для любого } x \in X, \text{ наделенного заданным свойством } p; \\ 0, & \text{для любого } x \in X, \text{ не обладающего заданным свойством } p. \end{cases}$$

Аргументами предикатной функции могут быть элементы произвольной природы, а область значений состоит из двух элементов: истина и ложь (или 1 и 0). Предполагается, что свойство p задано заранее и для любого $x \in X$ можно вычислить значение этого свойства и ответить на вопрос: обладает элемент $x \in X$ этим свойством или нет.

Предикатной функцией от n – переменных называется функция $p(x_1, x_2, \dots x_n)$, аргументами которой являются элементы декартова произведения n- множеств, а область значений также состоит из двух элементов: истина и ложь. Такая функция всякую область определения делит на две части или на два подмножества: подмножество, на котором функция принимает истинное значение и второе подмножество, на элементах которого предикатная функция принимает ложное значение.

Задачи и упражнения.

1.На множествах $A = \{a_1, a_2, a_3, a_4, a_5\}$, $B = \{b_1, b_2, b_3\}$:

а) построить сюръективное отображение $f\colon A \to B$;

б) построить инъективное отображение $f\colon B \to A$; подсчитать число инъективных отображений;

в) построить биективное отображение $f\colon A \to B$; $A = \{a_1, a_2, a_3, a_4, a_5\}$, $B = \{b_1, b_2, b_3, b_4, b_5\}$; подсчитать число биективных отображений.

2. Записать в виде выражения предикатные функции, заданные на множестве $A = \{a_1, a_2, a_3, a_4, a_5\}$:

P=1					
P=0	*	*	*	*	*
	a_1	a_2	a_3	a_4	a_5

P=1	*	*	*	*	*
P=0					
	a_1	a_2	a_3	a_4	a_5

P=1			*	*	*
P=0	*	*			
	a_1	a_2	a_3	a_4	a_5

2. Записать в виде выражения предикатные функции, заданные на множестве $A \times A$; $A = \{a_1, a_2, a_3, a_4, a_5\}$:

a_5					*
a_4				*	
a_3			*		
a_2		*			
a_1	*				
	a_1	a_2	a_3	a_4	a_5

a_5					*
a_4				*	*
a_3			*	*	*
a_2		*	*	*	*
a_1	*	*	*	*	*
	a_1	a_2	a_3	a_4	a_5

a_5					
a_4					*
a_3				*	*
a_2			*	*	*
a_1		*	*	*	*
	a_1	a_2	a_3	a_4	a_5

a_5		*		*	
a_4	*		*		*
a_3		*		*	
a_2	*		*		*
a_1		*		*	

	a_1	a_2	a_3	a_4	a_5

3. Для отображений $f: N \to N$, $f(n) = 2n$ $g: N \to N$, $g(n) = n+1$ построить композицию:

а) $f \bullet g$;

б) $g \bullet f$..

1.3.2.Частичные отображения.

Отображение $f: A \to B$ называется частичным отображением множества A во множество B, если $f: A_1 \to B_1$, $A_1 \subseteq A$, $B_1 \subseteq B$. Частичное отображение можно рассматривать как отображение, положив $A_1 = A$. В этом смысле всякое отображение является частным случаем частичного отображения. На частичные отображения распространяются введенные ранее понятия образа и прообраза и арности отображения. В дальнейшем A, A_1 , B, B_1 будем интерпретировать на множестве натуральных чисел N.

Кроме частичных отображений вида $f: N \to N$ и $f: N^r \to N$ можно говорить и о частичных отображениях вида $f: N \to N^s$ и в более общем виде $f: N^r \to N^s$ $(s>0)$. Отображение вида $f: N \to N^s$ строится следующим образом: берется s одноместных функций вида $f: N \to N$. Кортеж вида $<f_1(x), f_2(x), \ldots, f_s(x)> \in N^s$, где $x \in N$, представляет отображение $f: N \to N^s$. Каждому $x \in N$, входящему в области определения всех функций f_i, $i=1, 2, \ldots, s$, эти s функций ставят в соответствие кортеж $<f_1(x), f_2(x), \ldots, f_s(x)> \in N^s$, определяя тем самым отображение N в N^s.

Построим отображение вида $f: N^r \to N^s$ $(s>0)$. По аналогии с предыдущим построением возьмем s r-местных функций вида $N^r \to N$. Каждому $x \in N^r$, входящему в области определения всех функций f_i, $i=1, 2, \ldots, s$, эти s функций ставят в соответствие кортеж $<f_1(x_1,\ldots, x_r), f_2(x_1,\ldots, x_r), \ldots, f_s(x_1,\ldots, x_r)> \in N^s$, определяя тем самым отображение N^r в N^s.

Рассмотрим функцию $f: N^r \to N$. Графиком G_f функции f называется множество кортежей вида $< x_1,\ldots, x_r, y> \in N^{r+1}$, удовлетворяющих условию $f(x_1,\ldots, x_r) = y$ или это можно записать:

$$G_f = \{< x_1,\ldots, x_r, y> \in N^{r+1} \mid f(x_1,\ldots, x_r) = y \},$$

т.е. график r –местной функции лежит в (r+1)-местном пространстве. В частности, график нульместной фнкцуии f^0 лежит в одноместном пространстве.

Задачи и упражнения.

1.Выписать графики функций ддя задач из предыдущего раздела.

2.Выписать графики предикатных функций ддя задач из предыдущего раздела.

3.$y_1 = f_1(ax^2+ bx + c)$; $y_2 = f_2(dx + b)$ – в каком пространстве задано данное отображение; показать вид графика данного отображения.

4. $y_1 = f_1(a_1x^3+ a_2x^2 + a_3x + a_4)$ - в каком пространстве задано данное отображение; показать вид графика данного отображения.

2.Способы представления данных.

2.1.Понятие алфавита.

Формальное изучение любого круга вопросов начинается с замены реальных объектов некоторыми подходящим образом выбираемыми их абстрактными описаниями. При формализации понятия объекта будем считать, что алгоритм, обрабатывающий объект оперирует не с самими объектами, а с их изображениями. Например, 123, 341, 907 – это изображения соответствующих натуральных чисел в десятичной системе счисления (в римской системе число 123 будет иметь вид CXXIII).

Рассмотрим ряд вертикальных черточек |||||, полученных путем написания одной черточки, затем приписывания к ней справа другой черточки и последовательного приписывания справа к имеющимся двум еще трех черточек. Имеющиеся пять черточек мы можем назвать изображением натурального числа «пять». В принципе такими черточками можно изобразить и число 123, хотя это и займет много места. Таким образом, одно и тоже натуральное число мы можем изображать по разному.

Нетрудно заметить, что и объекты, не являющиеся натуральными числами, могут изображаться по разному. Элементарные знаки, участвующие в изображении данных будем называть буквами.

Определение 2.1. Совокупность букв, попарно различных между собой назовем алфавитом.

Так число 123 есть изображение в алфавите A = $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. На самом деле для числа 123 можно использовать и меньший алфавит $A' = \{1, 2, 3\}$, являющийся подмножеством алфавита A. Алфавит A естественно употребить для изображения любого числа множества целых чисел, элементом которого является число 123; изображение CXXIII этого же числа есть изображение в алфавите A = {I, V, X. C, L }. Изображение числа «пять» вертикальными черточками ||||| есть изображение в алфавите A = {|}. Причем, скобки {, } и запятые не являются элементами указанных алфавитов.

Конечные последовательности букв в заданном алфавите будем называть словами. Множество всех слов алфавита A будем обозначать A^*. Число всех слов в любом конечном алфавите A не более чем счетно. По аналогии с пустым множеством, на множестве всех слов A^* введем пустое слово λ, не содержащее ни одной буквы. Например, $A_1 = \{a_1\}$, тогда $A_1^* = \{\lambda, a_1, a_1a_1, a_1a_1a_1, \ldots\}$.

Пусть $A_р$ = {а, б, в, г, …., ю, я}- алфавит русских букв. Тогда множество всех слов этого алфавита $A_р^*$ содержит как все слова русского языка, так и все бессмысленные комбинации букв, например, аааа, бббааб, яяяяг, вакбю.

На множестве слов $A_р^*$ алфавита $A_р$ введем операцию конкатенации (*), определяемую следующим образом: пусть абба $\in A_р^*$ и бааб $\in A_р^*$. С помощью операции конкатенации получаем новое слово из $A_р^*$ вида:

абба*бааб = аббабааб.

Отметим следующие свойства операции конкатенации:

λ*абба = абба; абба*λ = абба; $\lambda*\lambda = \lambda$.

2.2.Кодирование данных.

2.2.1.Кодирование слов в алфавите.

Как следует из предыдущего параграфа, любой объект, будь то число, целое или вещественное, представим в виде слова в некотором алфавите.

Пусть дано множество слов в алфавите A, содержащем p букв. Ставится вопрос – можно ли изобразить эти слова в алфавите B, содержащем r букв. Рассмотрим два случая.

Случай 1. $p \le r$, т.е. мощность алфавита A не более мощности алфавита B. В этом случае можно считать, что имеет место включение $A \subseteq B$. Введем инъективное отображение

$\psi: A \to B$ вида $\psi(a_i) = b_i$.

Тогда, всякому слову $w(A) = a_{i1}\, a_{i2} \ldots a_{il}$ длины l в алфавите A можно поставить в соответствие слово длины l в алфавите B следующим образом

$w(B) = \psi(a_{i1})\, \psi(a_{i2}) \ldots \psi(a_{il})$. Это справедливо для слов любой конечной длины.

Случай 2. $p > r$. В этом случае не удается каждой букве алфавита A поставить в соответствие одну букву алфавита B, так как $p > r$. Поступим следующим образом: каждой букве алфавита A поставим в соответствие некоторое слово из букв алфавита B. Предварительно выделим в алфавите B множество слов, достаточное для представления всех букв алфавита A. Пусть для определенности $r = 2$, т.е. алфавит $B = \{b_1, b_2\}$. В этом алфавите число однобуквенных слов равно двум: b_1, b_2. Число двубуквенных слов равно четырем: $b_1 b_1$, $b_1 b_2$, $b_2 b_1$, $b_2 b_2$. Пусть число букв в слове равно трем. Тогда число всех трехбуквенных слов в алфавите равно числу всех отображений $f: X \to B$, где $|X| = 3$, $|B| = 2$. По формуле числа всех отображений имеем $|f: X \to B| = 2^3$. Выпишем эти отображения или все трехбуквенные слова в алфавите B:

Таблица 1

	1	2	3
1	b_1	b_1	b_1
2	b_1	b_1	b_2
3	b_1	b_2	b_1
4	b_1	b_2	b_2
5	b_2	b_1	b_1
6	b_2	b_1	b_2
7	b_2	b_2	b_1
8	b_2	b_2	b_2

Каждое из этих отображений можно рассматривать как слово в алфавите B длины три. В таблице 1 приведены все слова длины три в алфавите B. Нетрудно заметить, что число слов длины n в алфавите B равно 2^n. Для случая, когда алфавит B содержит r букв имеем число слов длины n равно r^n.

Вернемся к нашей задаче. Нам необходимо каждой букве алфавита А поставить в соответствие слово из алфавита В. В алфавите А p букв. Каждой букве алфавита А поставим в соответствие слово из алфавита В длины k, полученное из условий

$$r^{k-1} < p \le r^k.$$

Первое условие ($r^{k-1} < p$) означает, что слов длины (к-1) недостаточно для отображения всех p букв. Второе условие ($p \le r^k$) означает, что число слов длины к не меньше p букв. Введем инъективное соответствие

$$\psi: A \to B^k, \text{ или } \psi(a_i) = (b_{i1}, b_{i2}, \ldots, b_{ik}),$$

которое каждой букве a_i алфавита А ставит в соответствие слово длины k алфавита В. Здесь B^k – обозначение множества всех слов длины k в алфавите В

Тогда, всякому слову $w(A) = a_{i1}\, a_{i2} \ldots a_{il}$ длины l в алфавите А можно поставить в соответствие слово длины $L = l*k$ в алфавите В следующим образом

$$w(B) = \psi(a_1)\, \psi(a_2) \ldots \psi(a_l) =$$

$$(b_{11}, b_{12}, \ldots, b_{1k}, b_{21}, b_{22}, \ldots, b_{2k}, \ldots, b_{l1}, b_{l2}, \ldots, b_{lk})$$

Это справедливо для слов любой конечной длины.

Задачи и упражнения.

1.Какой должна быть длина слов (число k) в алфавите В с r = 3 для отображения p=10 букв алфавита А. Сколько можно построить таких отображений.

2.Закодировать алфавит А, состоящий из p=16 символов, словами из алфавита В, состоящего из r=5 символов.

2.2.2.Системы счисления.

Рассмотрим кодирование целых чисел. Как правило, при представлении чисел, мы пользуемся десятичной системой счисления или системой счисления по основанию 10. Это означает, что в ней при представлении любого числа могут использоваться только десять различных цифр, а именно: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

В общем случае, в системе счисления по основанию p всякое число N задается единственным образом в виде

$$N = \sum_{i=0}^{k-1} \propto_i p^i, \ \propto_i \in \{0, 1, \ldots, p-1\}, \ i = 0, 1, \ldots, k; \ k\text{- разрядность числа.}$$

Например, в системе счисления по основанию p = 10 число 324 представимо в виде $324 = 3*10^2 + 2*10^1 + 4*10^0$. Запись этого числа в системе счисления по основанию p = 10 есть упорядоченная строка цифр $(\propto_2, \propto_1, \propto_0) = (3, 2, 4)$, k = 2. При его вычислении каждый коэффициент $\propto_i$ умножается на соответствующее значение весовой функции $p^i = 10^i$.

В системе счисления по основанию p = 2 число 324 представимо в виде:

$$1*2^8 + 0*2^7 + 1*2^6 + 0*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0.$$

Запись этого числа в системе счисления по основанию p = 2 есть упорядоченная строка цифр $(\propto_8, \propto_7, \propto_6, \propto_5, \propto_4, \propto_3, \propto_2, \propto_1, \propto_0) = (1, 0, 1, 0, 0, 0, 1, 0, 0)$,

$k = 8$. При его вычислении каждый коэффициент $\propto_i$ умножается на соответствующее значение весовой функции $p^i = 2^i$.

Системы счисления, в которых положение коэффициента $\propto_i$ строго фиксировано, называются позиционными.

Важной позиционной системой счисления является система счисления по основанию $p = 16$. Ее основанием являются символы $p = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$. В этой системе счисления число 324 представимо в виде: $1*16^2 + 4*16^1 + 4*16^0$. Запись этого числа в системе счисления по основанию $p = 16$ есть упорядоченная строка цифр $(\propto_2, \propto_1, \propto_0) = (1, 4, 4)$, $k = 2$. При его вычислении каждый коэффициент $\propto_i$ умножается на соответствующее значение весовой функции $p^i = 16^i$.

Число 2A4F в шестнадцатиричной системе счисления равносильно числу 10831 в десятичной системе счисления.

$2A4F = 2*16^3 + A*16^2 + 4*16^1 + F*16^0 \Rightarrow 2*4096 + 10*256 + 4*16 + 15*1 = 8192 + 2560 + 64 + 15 = 10831$.

2.2.2.1.Перевод чисел из десятичной системы счисления в двоичную.

п.1.Число N делится на основание системы счисления $p = 2$. Запоминаем остаток (ost1) от деления и частное N/2.

п.2.Повторять п.1. до тех пор, пока на k-м шаге не выполнится условие $N/2^k < 2$. Тогда, образом числа N в двоичной системе счисления будет упорядоченная строка цифр вида: ($N/2^k$, ost (k-1), …, ost2, ost1), ost(k-1)) – остаток от деления числа $N/2^{k-1}$ на основание системы счисления p=2 на (k-1)-м шаге.

2.2.2.2.Перевод чисел из p - ичной системы счисления в десятичную.

п.1.Возьмем p - ичное представление числа

$N = \propto_к*p^к + \propto_{к-1}*p^{к-1} + \ldots + \propto_0*p^0$.

п.2.Выполним (к+1) - раз операцию возведения в степень числа p, (к+1) – раз операцию умножения вида $\propto_i*p^i$, $i = 0, 1, \ldots, k$ и к- раз операцию сложения над (к+1) – м слагаемым вида $\propto_i*p^i$, $i = 0, 1, \ldots, k$. Будем все операции выполнять в десятичной системе счисления. После выполнения всех операций получим число в десятичной системе счисления.

Задачи и упражнения.

1.Дать двоичное представление чисел: 2, 3, 4, 7, 9, 11, 19, 111111, 987654321.

2.Дать троичное представление чисел: 2, 3, 4, 7, 9, 11, 19, 111111, 987654321.

3.Дать восьмеричное представление чисел: 2, 3, 4, 7, 9, 11, 19, 111111, 987654321.

4.Дать шестнадцатеричное представление чисел: 2, 3, 4, 7, 9, 11, 19, 111111, 987654321.

5.Дать двоичное представление шестнадцатеричных чисел: 2, 3, 4, 7, 9, 11, F4, 1B207, 3AD68B1.

2.3.Представление множеств.

В предыдущем параграфе приводится правило кодирования слова или объекта в заданном алфавите. Как правило, всякий объект является элементом некоторого множества. Встает вопрос, как представить множество объектов. При этом, будем предполагать, что все элементы множества или объекты является словами некоторого наперед заданного алфавита.

Существуют различные способы задания множества. Рассмотрим некоторые из них.

Первый способ – это перечисление элементов этого множества, т.е. множество задается своими элементами. В частности, элементы множества можно задать их индексами, о чем говорит следующее утверждение.

Пусть задано конечное множество $X = \{x_1, x_2, \ldots, x_n\}$.

Утверждение1. Всякое конечное множество X может быть представлено в цифровом алфавите ЦА = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

Действительно, введем множество целых положительных чисел длины n: $N = \{1, 2, \ldots, n\}$. Очевидно, что каждый элемент этого множества легко может быть представлен в алфавите ЦА. Установим взаимно – однозначное соответствие $f : X \to N$, которое x_i элементу множества X ставит в соответствие i-й элемент множества N. Тогда изображением элемента x_i множества X является изображение i-го числа множества N.

Второй способ - это указание правила, проверяющего принадлежность любого объекта множеству. В качестве такого правила выступает предикатная функция p(x) такая, что, если p- предикат, а x – объект, то

p(x) = истина, в случае, когда x принадлежит исходному множеству, либо

p(x) = ложь, в противном случае.

Пример. Рассмотрим множество $A = \{a_1, a_2, \ldots, a_n\}$, элементы которого являются целыми числами. Запишем предикатную функцию, выделяющую на этом множестве все числа, которые больше наперед заданного числа $a_0=15$:

$$p(a_i) = \begin{cases} 1, \text{ если } a_i > a_0, \\ 0, \text{ в противном случае.} \end{cases}$$

Если p(x) – некоторое условие на объект x, а X – множество, то через $Y = \{x \mid p(x), x \in X\}$ обозначаем множество $Y \subseteq X$, содержащее в качестве элементов те и только те элементы $x \in X$, которые удовлетворяют условию p(x).

Третий способ - описание множества по индукции.

Пусть X – множество, $X_0 \subseteq X$ и $f_1, \ldots, f_k$ – операции на X, местности которых равны $n_1, \ldots, n_k$ соответственно. Определим множество $W \subseteq X$ следующим образом: $x \in W$ тогда и только тогда, когда (т.т.т.к) существует последовательность $x_0, x_1, \ldots, x_m$ элементов множества X, обладающая следующим свойством: $x_m = x$ и для любого $i \le m$ либо $x_i \in X_0$, либо $x_i = f_j(x_{i1}, x_{i2}, \ldots, x_{inj})$ для некоторого $j \in \{1, \ldots, k\}$ и некоторых $i_1, i_2, \ldots, i_{nj} < i$. Если множество W построено таким образом, то будем говорить, что множество W определено по индукции с помощью следующего определения:

1. если $x \in X_0$, то $x \in W$;
2. если $i \in \{1, \ldots, k\}$ и $x_{i1}, x_{i2}, \ldots, x_{inj} \in W$, то $f_j\,(x_{i1}, x_{i2}, \ldots, x_{inj}) \in W$.

Пример. Пусть $X_0 = \{a, b\}$, $f_1(a, a) = a$, $f_2(a, b) = ab$, $f_3(b, a) = ba$, $f_4(b, b) = b$. Тогда множество W будет содержать следующие элементы:

W = {a, b, ab, ba, abb, abbb, baa, baaa, bab, aba,….}.

2.4.Представление отношений.

Всякое n - арное отношение $\Re$ есть подмножество декартова произведения множеств. $\Re \subseteq A_1 \times A_2 \times \ldots \times A_n$. Элемент отношения можно записать в виде вектора длины n: $(a_1, a_2, \ldots, a_n) \in \Re$.

Для задания всего отношения $\Re$ необходимо и достаточно задать предикатную функцию p: $A_1 \times A_2 \times \ldots \times A_n \to \{0, 1\}$, заданную на всем декартовом произведении множеств $A_1 \times A_2 \times \ldots \times A_n$ и принимающую единичное значение в точках, входящих в отношение $\Re$.

Описание отношений рассмотрим на примере бинарного отношения. На декартовом произведении множеств A×B, $A = \{a_1, a_2, a_3, a_4, a_5\}$, $B= \{b_1, b_2, b_3\}$ рассмотрим отношение α, помеченное звездочками в таблице 2:

Таблица 2

b_3			*	*	
b_2		*			
b_1	*				*
	a_1	a_2	a_3	a_4	a_5

Его можно записать следующим образом: $\alpha = \{(a_1, b_1), (a_2, b_2), (a_3, b_3), (a_4, b_3), (a_5, b_1)\}$. Чтобы описать данное отношение с помощью предикатной функции введем нумерацию всех элементов декартового произведения, представленной в таблице 3.

Таблица 3

b_3	(3, 1)	(3, 2)	(3, 3)	(3,4)	(3, 5)
b_2	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
b_1	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
	a_1	a_2	a_3	a_4	a_5

Зная нумерацию элементов всего декартового произведения и само отношение α можно задать предикатную функцию, описывающую это отношение, следующим образом:

$$p((a_i, b_j) = \begin{cases} 1, \text{если } (i, j) = (1, 1) \cup (2, 2) \cup (3, 3) \cup (3, 4) \cup (1, 5); \\ 0, \text{в противном случае.} \end{cases}$$

Для задания бинарного отношения первым способом необходимо от двумерной нумерации элементов декартова произведения перейти к одномерной. Это можно сделать по разному, например, как в приведенных ниже таблицах 4, 5, 6.

Таблица 4

b_3	(3)	(6)	(9)	(12)	(15)
b_2	(2)	(5)	(8)	(11)	(14)
b_1	(1)	(4)	(7)	(10)	(13)
	a_1	a_2	a_3	a_4	a_5

Таблица 5

b_3	(11)	(12)	(13)	(14)	(15)
b_2	(6)	(7)	(8)	(9)	(10)
b_1	(1)	(2)	(3)	(4)	(5)
	a_1	a_2	a_3	a_4	a_5

Таблица 6

b_3	(6)	(9)	(12)	(14)	(15)
b_2	(3)	(5)	(8)	(11)	(13)
b_1	(1)	(2)	(4)	(7)	(10)
	a_1	a_2	a_3	a_4	a_5

Переход от двумерной нумерации к одномерной для таблицы 4 запишется следующим образом: n = (j-1)+3*(i –1) +1.

При такой нумерации предикатная функция для таблицы 4 будет иметь вид:

$$p(n) = \begin{cases} 1, \text{ если } n = 1 \cup 5 \cup 9 \cup 12 \cup 13; \\ 0, \text{ в противном случае.} \end{cases}$$

Переход от двумерной нумерации к одномерной для таблицы 5 запишется следующим образом: n = (j-1)+5*(i –1) +1. Предикатная функция для таблицы 5 будет иметь вид:

$$p(n) = \begin{cases} 1, \text{ если } n = 1 \cup 7 \cup 13 \cup 14 \cup 5; \\ 0, \text{ в противном случае.} \end{cases}$$

Переход от двумерной нумерации к одномерной для таблицы 6 запишется следующим образом:

$$n = \begin{cases} (0) + 1, \text{ если } (i + j) = 0; \\ (2*i + j) + 1, \text{ если } (i + j) = 1; \\ (3*i + j + 1) + 1, \text{ если } (i, j) = (0, 2); \\ (3*i + j) + 1, \text{ если } (i, j) = (1, 1); \\ (3*i + j - 1) + 1, \text{ если } (i, j) = (2, 0); \end{cases}$$

$$\begin{cases} (3*i + 2*j) + 1, \text{ если } (i + j) = 3; \\ (3*i + 2*j + 1) + 1, \text{ если } (i + j) = 4 \cup 5; \\ (3*i + 2*j) + 1, \text{ если } (i + j) = 6. \end{cases}$$

Предикатная функция для таблицы 6 будет иметь вид:

$$p(n) = \begin{cases} 1, \text{ если } n = 1 \cup 5 \cup 10 \cup 12 \cup 14; \\ 0, \text{ в противном случае.} \end{cases}$$

2.5.Представление отображений.

Всякое отображение f: $A \to B$ есть бинарное отношение на декартовом произведении множеств $A \times B$, $|A| = n$, $|B| = m$. Поэтому его можно задать множеством пар $\{(a_i, b_j)\}$, где $a_i \in A$, $b_j \in B$.

Заменим множество A на множество целых чисел такой же мощности $N = \{1, 2, \ldots, n\}$. Тогда, всякое отображение f: $N \to B$ можно задать одной строкой вида $(b_1, b_2, \ldots, b_n)$, где b_i – образ элемента a_i, $i = 1, 2, \ldots, n$ отображения f: $N \to B$. Таким образом, любое отображение можно задать словом из элементов множества B. В свою очередь, элементы множества B могут быть представлены в некотором заданном алфавите.

Рассмотрим вопрос о представлении множества всех отображений f: $N \to B$. Положим для определенности, $n = 2$, $m = 4$. Тогда, число всех отображений есть $m^n = 4^2 = 16$, а множество всех отображений можно представить в виде таблицы 7.

Таблица 7

N / f(N)	1	2
0	b_0	b_0
1	b_0	b_1
2	b_0	b_2
3	b_0	b_3
4	b_1	b_0
5	b_1	b_1
6	b_1	b_2
7	b_1	b_3
8	b_2	b_0
9	b_2	b_1
10	b_2	b_2
11	b_2	b_3
12	b_3	b_0
13	b_3	b_1
14	b_3	b_2

15	b_3	b_3

Обозначим N^* - множество всех отображений $f: N \to B$ при $n = 2$, $m = 4$. Нетрудно заметить, что это множество есть множество чисел от 0 до 15. Каждому элементу $j \in \{0, \ldots, 15\}$ этого множества соответствует одно отображение вида (b_i, b_k). Паре (b_i, b_k) поставим в соответствие набор ее индексов (i, k). Последнюю пару будем рассматривать как число в системе счисления по основанию $m = 4$. Тогда, $j = i*4^1 + k*4^0$. Скажем, для $j = 9$ паре (b_2, b_1) ставится в соответствие пара индексов $(2, 1)$ или число $2*4^1 + 1*4^0 = 9$. Верно и обратное утверждение: по числу j можно определить соответствующую ему пару (i, k). Таким образом, множество всех отображений можно представить множеством чисел в интервале от 0 до $m^n - 1$.

Задачи и упражнения.

1.Какой мощности надо выбрать множество N, чтобы проиндексировать (первый способ) множество всех отображений
$f: A \to B$, если $A = \{a_1, a_2, a_3\}$, $B= \{b_1, b_2, b_3\}$.

2. Какой мощности надо выбрать множество N, чтобы проиндексировать множество всех бинарных отношений, взятых на декартовом произведении множеств $A \times B$, $A = \{a_1, a_2, a_3\}$, $B= \{b_1, b_2, b_3\}$.

3.Записать предикатную функцию, выделяющую на множестве целых положительных чисел N:

а) четные числа;

б) нечетные числа;

в) числа, кратные трем;

г) числа, кратные трем и пяти.

4.Проверить будут ли элементы aaaa, bbbbbb, abaaaaaaa, abababab, babababa принадлежать множеству W, полученному по индукции в разделе 2.3.

3.Интерпретация данных.

Формальное изучение любого круга вопросов начинается с замены реальных объектов некоторыми подходящим образом выбираемыми их абстрактными описаниями. Эти описания или идеализации выбираются таким образом, чтобы в них были отражены именно те свойства исходных объектов, которые мы собираемся изучать.

В нашем случае реальными объектами являются данные (D): множества, отношения, отображения, математические структуры. Их абстрактными описаниями будут некоторые синтаксические конструкции (S), являющиеся способом представления данных. Связь между синтаксическими конструкциями (реальных объектов) и самими реальными объектами будет осуществляться через *функцию интерпретации* ν: $S \to D$, которая приписывает каждой синтаксической конструкции в качестве ее значения соответствующие реальные объекты.

Рассмотрим множество $X = \{x_1, \dots, x_n\}$. Будем интерпретировать элементы этого множества на множестве натуральных чисел N. В данном примере в качестве синтаксической конструкции выступает множество X, заданное своими элементами. Объектами реального мира выступает множество N. Таким образом, на множестве X задана функция интерпретации ν: $X \to N$, которая каждый элемент множества X интерпретирует числом из натурального ряда N, т.е. значением элемента x_i для $i \in \{1, \dots, n\}$ может быть любое из чисел: $\{0, 1, 2, \dots\}$.

Фактически всякое множество задается парой $\mathbf{X} = < X, \nu >$, где X – множество, ν: $X \to N$ – функция интерпретации. Если же множество задается одним именем, например, X, то функция интерпретации предполагается известной и заданной. Нельзя различные элементы множества X одновременно интерпретировать разными функциями. В литературе множество N называется областью интерпретации или основным множеством.

Рассмотрим множество $X = \{x_1, \dots, x_n\}$. Будем интерпретировать его элементы на множестве целых чисел $N^ц$. В данном примере на множестве X задана функция интерпретации ν: $X \to N^ц$, по которой значением элемента x_i для $i \in \{1, \dots, n\}$ может быть любое из чисел: $\{\dots, -2, -1, 0, 1, 2, \dots\}$.

Задание множества есть пара $\mathbf{X} = < X, \nu >$, где X – множество, ν: $X \to N^ц$ – функция интерпретации.

Рассмотрим бинарное отношение $\propto \subseteq A\times B$, заданное на декартовом произведении множеств A, B. Элементами отношения $\propto$ являются пары $(a, b) \subseteq A\times B$. Введем функцию интерпретации для элементов множества A: ν_1: $A \to N$, функцию интерпретации для элементов множества B: ν_2: $B \to N^ц$. Тогда областью интерпретации или основным множеством для бинарного отношения $\propto$ будет декартово произведение $N\times N^ц$. Зачастую множества A и B могут интерпретироваться на одном множестве.

Рассмотрим отображение f: $X \to Y$. Введем функцию интерпретации для элементов множества X: ν_1: $X \to N$, функцию интерпретации для элементов множества Y: ν_2: $Y \to N$. Фактически это означает, что задано отображение f: $N \to N$. Для полного и однозначного задания отображения необходимо указать конкретный вид символа f, т.е. интерпретировать символ f. Например, зададим это отображение следующим образом: $f(n) = n+1$. Это отображение всякому элементу $n \in N$ ставит в соответствие элемент $(n+1) \in N$.

Пример. Рассмотрим следующую функцию $y= g(h(x), y)$, где $h(x)= x$.

Рассмотрим первую интерпретацию этой функции. Переменные x, y будем интерпретировать на множестве натуральных чисел, т.е. $x, y \in N$, $x = n$. Функциональные символы будем интерпретировать следующими операциями:

$h(x)= x +1$- операция сложения числа x с единицей;

$g(h(x), y) = h(x) *y$ – операция умножения функции h(x) на переменную y; и пусть при $x = 0$ $y = h(x)$. При заданных интерпретациях функцию можно записать следующим образом:

$$y = \begin{cases} h(x), & \text{при } x = 0; \\ h(x) *y, & \text{при } x =< n; \\ y\ , & \text{при } x > n. \end{cases}$$

При данной интерпретации приведенная функция вычисляет значение n! или n- факториал.

Рассмотрим вторую интерпретацию этой функции. Переменные x, y будем интерпретировать на множестве слов в алфавите, например, русских букв, т.е. $x, y \in A_p^*$, т.е. x есть слово в русском алфавите, например, $x = (a_1, a_2, \ldots, a_n)$ – слово, длины n, $l(x) = n$. Функциональные символы будем интерпретировать следующими операциями:

одноместная функция h(x) стирает первую букву в слове x, т.е. $h(a_1, a_2, \ldots, a_n) = (a_2, \ldots, a_n)$;

двухместная функция g(h(x), y) приписывает первую букву первого слова первой буквой второго слова, т.е. $g((a_2, \ldots, a_n), (a_1)) = (a_2, a_1)$; и пусть при $l(x) =n$, или $x = (a_1, a_2, \ldots, a_n)$ $y = h((a_1, a_2, \ldots, a_n)) =\lambda$, где λ - пустое слово. При заданных интерпретациях функцию можно записать следующим образом:

$$y = \begin{cases} \lambda = h(x), & \text{при } l(x)=n; \\ g(h\ (a_i, a_{i+1}, \ldots, a_n), (a_{i-1}, a_{i-2}, \ldots, a_1)) = (a_i, a_{i-1}, a_{i-2}, \ldots, a_1, \lambda), & \text{при } l(x) < n; \\ y\ , & \text{при } l(x) > n. \end{cases}$$

При данной интерпретации приведенная функция любое слово длины n
$x = (a_1, a_2, \ldots, a_n)$ в заданном алфавите переводит в «перевернутое» слово
$y = (a_n, a_{n-1}, \ldots, a_1)$.

4.Элементы теории вычислимости.

4.1.Понятие терма.

В разделе 1.3 было введено определение функции или отображения. При записи отображения мы пользовались некоторым языком. Например, произвольное отображение можно было записать в виде $y = f(x)$ либо $f: X \to Y$. Если имеется в виду некоторое конкретное отображение, тогда требуется интерпретировать предметные и функциональный символы, как указано в разделе 3. Рассмотрим правила построения записей отображений или функций.

Для записи функций воспользуемся следующим формальным языком. Его алфавит (А) составляют элементы трех непересекающихся множеств. Элементы первого множества назовем *предметными символами* (ПС). В качестве таких символов будем употреблять буквы a, b, x, y, … или же эти буквы с индексами: a_0, b_0, b_1, x_0, x_1, x_2, … Символы второго множества назовем функциональными (ФС). Будем их обозначать буквами с верхними или нижними индексами: f^1, f^2, g_n, … Верхний индекс n означает местность функционального символа. Элементы третьего множества – это символы – разделители (Р) типа левой, правой скобок, запятой: ‘(‘, ‘)’,’,’. В последнем предложении символы - разделители взяты в апострофы. А = {ПС}$\cup${ФС}$\cup${Р}.

Описанием функции в алфавите А называется слово, записанное с помощью символов алфавита А по определенным правилам.

Определим алфавит А следующим образом:

А = {ПС}$\cup${ФС}$\cup${Р} = {x, y}$\cup${f, g}$\cup${(,), ‘,’}.

В последних фигурных скобках символ запятой, взятой в апострофы, есть элемент множества разделителей Р. Первые две запятые в этих скобках не являются элементами множества разделителей, а выступают как символы (метасимволы), с помощью которых описано множество разделителей Р.

Правила построения записей функций рассмотрим на примерах. Приведем примеры правильно записанных слов в алфавите А: x, $f^1(x)$,
$g^2(x, y)$, $g^3(x, f^2(x,y), y)$. Здесь x – однобуквенное слово, являющееся предметной переменной, т.е. $x \in$ ПС. $f^1(x)$ – запись одноместной функции f от предметной переменной x. Скобки отделяют функциональный и предметный символы. $g^3(x, f^2(x, y), y)$ – запись трехместной функции g от предметной переменной x, двуместной функции $f^2(x,y)$, предметной переменной y. Внутри скобок переменные разделяются запятой.

Приведем примеры неправильно записанных слов в алфавите А: записи вида)(, xf, fx, f, (f)xy не являются описаниями функций в нашем алфавите.

Слова в алфавите А, правильно описывающие функции, называются термами. Терм – это словарное описание функции. Дадим определение терма или правильно записанного слова в нашем алфавите. Такое определение дается индуктивно, где индукция берется по длине терма.

Определение терма. Термами длины 1 называются однобуквенные слова из множества предметных символов ПС, т.е. всякий символ из множества ПС есть правильно описанный терм.

Пусть, для некоторого $k > 1$ определены термы длины, меньшей k. Тогда слово длины k называется термом, если оно имеет вид $f^n(a_1, \dots, a_n)$, где $a_1, \dots, a_n$ – некоторые термы меньшей длины; f – n-местный функциональный символ. Приведенные выше слова x, $f^1(x)$, $g^2(x, y)$, $g^3(x, f^2(x,y), y)$ являются термами, длины которых соответственно равны 1, 5, 7, 15.

По нашему определению термами являются либо предметные переменные из множества ПС, либо функциональные выражения вида $f^n(a_1, \dots, a_n)$. Здесь важно, чтобы арность n функции f соответствовала числу аргументов в записи функции, т.е. в скобках должно быть n аргументов иначе предложение неправильное. Пример неправильного предложения: $f^3(x, y)$, в котором арность функции (=3) не соответствует числу аргументов (=2).

4.2.Интерпретация терма.

Определенный в предыдущем разделе терм – есть синтаксическое описание функции. Это только символьная запись функции в заданном алфавите. Такую запись можно было определить и по другим правилам. Например, двухместную функцию можно было записать по следующему правилу xf^2y. Здесь два аргумента расположены с двух сторон относительно функционального символа f. Важно подчеркнуть, что одну и ту же функцию можно описать по разным синтаксическим правилам. Задание таких правил позволяет описание функции сделать однозначным. Помимо синтаксического описания функции есть ее семантическое содержание, называемое значением или интерпретацией терма.

Определим значение терма при заданных значениях входящих в него функциональных и предметных символов. Предметные символы будем интерпретировать, например, значениями из множества N^+ целых положительных чисел. n- местный функциональный символ будем интерпретировать n- местной операцией f_0, определенной на основном множестве N^+. Тогда терм будет интерпретироваться следующим образом.

Значением терма длины 1 является значение предметного символа, образующего этот терм, и взятое из основного множества N^+.

Рассмотрим терм k- й длины $\propto = f^n(a_1, \dots, a_n)$, где $a_1, \dots, a_n$ – некоторые термы меньшей длины; f – n-местный функциональный символ. По индукции можно считать, что значения термов $a_1, \dots, a_n$ уже определены своими значениями $\underline{a_1}, \dots, \underline{a_n}$ из основного множества N^+. Функциональному символу f^n поставлена в соответствие n – местная операция f_0, определенная на N^+. Значением этой операции в точке ($\underline{a_1}, \dots, \underline{a_n}$) будет некоторый элемент $\underline{a_0}$ из N^+. Элемент $\underline{a_0} \in N^+$ будет значением терма $\propto = f^n(a_1, \dots, a_n)$ при заданных интерпретациях предметных и функционального символов.

Пример. Рассмотрим терм $\propto = g^2(f^2(x, y), x)$. Будем интерпретировать его в арифметике натуральных чисел. Предметным переменным поставим в соответствие конкретные значения из N: x =3, y = 9. Двухместный функциональный символ f будем интерпретировать операцией сложения (+). Двухместный функциональный символ g будем интерпретировать операцией умножения (*). Тогда вышеназванный терм можно записать $\propto = ((3+9) *3) = 36$.

В этой же интерпретации терм $\beta = ((2x+y)*x)+7 = ((2*3+9) *3) +7 = 52$. Терм β можно рассматривать как новую бинарную операцию, которая по двум аргументам x, y вычисляет новое значение на N в соответствии с описанием терма. Таким образом, имея конечное число операций типа (+, *), заданных на фиксированном множестве N – можно при помощи термов записать не более чем счетное множество новых операций, определенных на том же множестве.

Отметим, что положение предметных переменных в терме упорядочено, так как изменение порядка переменных в терме может привести к изменению значения терма. Например, $f^2(g^2(2, x), y) = 2*x +y \neq 2*y +x$, для любых x, y, таких, что $x \neq y$.

Все дальнейшие рассмотрения будем делать над числовыми функциями, определенными на множестве натуральных чисел N. Это не умаляет общности наших рассуждений и не требует постоянного напоминания об интерпретации функций.

4.3.Вычислимые функции.

Первое неформальное определение вычислимой функции – это такая функция, для которой существует алгоритм, вычисляющий ее значения [11] на всей области определения. В этом определении вычислимость функции связывается с понятием алгоритма. Задать функцию означает задать закон, по которому одним объектам, называемым значениями аргумента, ставятся в соответствие другие объекты, называемые значениями функции. При этом ничего не говорится о характере этого закона. В частности, этот закон может быть таким, что он не дает реальной возможности вычислить по значению аргумента значение функции. Рассмотрим классический пример такого закона [12]. Введем следующую функцию:

$$f(n) = \begin{cases} 1, \text{ если в десятичном разложении числа } \pi \text{ имеется } n \text{ нулей подряд;} \\ 0, \text{ в противном случае.} \end{cases}$$

Так описанная функция f считается заданной. Однако, современное состояние наших знаний не дает способа вычислять значения этой функции для любых n. Такой способ либо не существует вовсе, либо существует, но мы его пока не знаем. Следовательно, такой функции мы не можем поставить в соответствие алгоритм, вычисляющий ее значения, следовательно, эта функция не является вычислимой.

Таким образом, каждой вычислимой функции должен соответствовать алгоритм, вычисляющий ее значения; на множестве всех функций вычислимые функции образуют некоторое подмножество.

С другой стороны, всякий алгоритм реализует некоторую функцию. Процесс вычисления функции называется процессом выполнения алгоритма. Этот процесс заключается в последовательном выполнении совокупности элементарных действий. Между этими действиями алгоритм находится в одном из своих состояний. Некоторые состояния алгоритма объявляются как заключительные - это такие состояния, после которых алгоритм заканчивает свою работу с выдачей результата. Переход из одного состояния в следующее задается правилами перехода. При выполнении алгоритма возможны три пути протекания этого процесса: 1) каждое состояние сменяется следующим и процесс останавливается, но безрезультатно; 2) на некотором шаге возникает состояние, к которому не применимы ни правила непосредственного перехода, ни правила окончания процесса и таким образом, алгоритм работает бесконечно, не останавливаясь; 3) на некотором шаге процесс переходит в заключительное состояние, происходит остановка алгоритма с получением результата. Алгоритм следовательно применим лишь к тем функциям, для которых процесс выполнения развивается по третьему пути.

После описания алгоритмического процесса уточним понятие вычислимой функции. Даны два множества элементов X – область определения функции f; Y – область значений функции f. Функция $f: X \to Y$ называется вычислимой, если существует алгоритм такой что, для любого $x \in X$ результат применения алгоритма к x совпадает со значением f(x).

В данном определении говорится о существовании алгоритма, который, будучи применен к некоторому значению аргумента x из области определения функции f в результате некоторого процесса, за конечное число шагов дает результат, совпадающий со значением функции.

Если же на заданном значении аргумента функция не определена, то алгоритм об этом ничего сказать не может.

Действительно, если к такому значению аргумента применить алгоритм вычисления значения функции, то алгоритм может вести себя по разному: либо привести к заведомо неверному результату и закончить свою работу, либо работать бесконечно. В первом случае алгоритм фактом завершения своей работы даст ответ на вопрос, что функция не определена на заданном значении аргумента. Во втором случае, когда алгоритм будет работать бесконечно, его работу относительно функции можно интерпретировать по разному: либо он работает правильно, но еще не дошел до конца, либо он будет работать бесконечно и функция не определена в этой точке. Т.е. алгоритм не может подтвердить того факта, что функция не определена в заданной точке x.

Для уточнения понятия вычислимой функции введем множество базовых функций, которые в силу своей простоты вычислимы. Далее рассмотрим набор базовых операций над этими функциями. В результате применения этих операций к базовым функциям снова получается вычислимая функция. Тогда получается следующее утверждение. Если произвольная заданная функция может быть представлена комбинацией базовых функций посредством базовых операций, то заданная функция вычислима. Важно только, чтобы набор базовых функций был по возможности функционально полным. Это необходимо, чтобы покрыть как можно большее число функций.

4.4.Базовые функции.

Под базовыми будем понимать функции, которые интуитивно - вычислимы.

Введем следующие числовые функции.

$\mathbf{s}^1(x) = x+1$ – функция следования; это унарная функция или функция от одной переменной, заданной на множестве N. Значения этой функции также принадлежат множеству N. Ее можно записать и так: $\mathbf{s}^1$: $N \rightarrow N$. Содержательно или семантически эта функция каждому элементу $x \in N$ ставит в соответствие элемент $(x+1) \in N$.

Алгоритм, реализующий эту функцию: каждому элементу $x \in N$ ставится в соответствие следующий за ним элемент $(x+1) \in N$.

$\mathbf{o}^n(x_1, \ldots , x_n) = 0$ – функция, присваивающая своим аргументам нулевое значение; это n – арная функция или функция от n переменных. Ее можно записать и так: $\mathbf{o}^n$: $N^n \rightarrow N$. Содержательно или семантически эта функция каждой из n переменных $x_i \in N$ ставит в соответствие значение нуль, $x_i = 0$ для любого $i \in \{1, 2, \ldots, n\}$.

Алгоритм, реализующий эту функцию: каждому элементу $x_i \in N$ для любого $i \in \{1, 2, \ldots, n\}$ ставится в соответствие значение нуль.

$\mathbf{I}^n{}_m(x_1, \ldots, x_n) = x_m$, где $m \in \{1,\ldots, n\}$, $n = 1, 2, \ldots$ - функция проецирования или функция выбора аргумента; это n – арная функция или функция от n переменных. Ее можно записать и так: $\mathbf{I}^n{}_m$: $N^n \rightarrow N$. Содержательно или семантически эта функция объявляет своим значением m – ю компоненту n- вектора $(x_1, \ldots, x_n)$, равную x_m, $m \in \{1, 2, \ldots, n\}$. Число m задано в качестве нижнего индекса функционального символа $\mathbf{I}^n{}_m$.

Алгоритм, реализующий эту функцию: из n- вектора $(x_1, \ldots, x_n)$, заданного в описании функции, надо выбрать m-ю компоненту x_m и объявить ее значением функции.

Пример функции проецирования. $\mathbf{I}^1{}_1(x) = x$, $\mathbf{I}^2{}_1(x_1, x_2)=x_1= \mathbf{I}^1{}_1(x_1)$. У функции $\mathbf{I}^n{}_m(x_1, \ldots, x_n)$ (n-1) фиктивных аргументов.

Пусть функция g^1 определяется равенством $g(x)=x$. Введем функцию f^2: $f(x, y) = x$. Как видно из определения функций g^1 и f^2 для всех пар (x, y) имеет место равенство $f(x, y) = g(x)$. Про переход от функции g^1 к функции f^2 говорят: функция f^2 получена введением фиктивного аргумента (переменной y) в функцию g^1.

4.5.Базовые операции над функциями.

Введение операций над функциями необходимо для получения новых функций из заданных. При этом предполагается, что если заданные функции вычислимы в интуитивном смысле, то и новые функции, получаемые с помощью этих операций, также будут вычислимыми в этом же смысле.

4.5.1.Операция подстановки.

Пусть заданы n числовых функций $f^m{}_1, \ldots, f^m{}_n$, каждая из которых от m – переменных или m- местная. Зададим также функцию f от n переменных.

Случай1. Регулярная подстановка.

В операции подстановки участвует n отображений вида $f^m{}_i$: $A^m \to B$, $i \in \{1, 2, \ldots, n\}$. n – арное отображение f можно записать в виде f^n: $B^n \to C$.

Определим новую функцию g от m – переменных через заданные функции с помощью операции подстановки следующим образом:

$g^m (x_{1, \ldots,} x_m) = f^n (f^m{}_1(x_1, \ldots, x_m), \ldots, f^m{}_n(x_1, \ldots, x_m))$.

Операция подстановки заключается в следующем. В функцию f^n в качестве первого аргумента была подставлена функция $f^m{}_1$. В качестве второго аргумента подставлена функция $f^m{}_2$, и т.д. по всем n аргументам. Так как все эти n функций определены от одних и тех же переменных $(x_1, \ldots, x_m)$, заданных на одном множестве A^m, то и полученная функция будет зависеть только от этих аргументов с той же областью определения. При заданных определениях исходных функций $f^m{}_1, \ldots, f^m{}_n$, f^n и по определению операции подстановки функция g есть функция вида g^m: $A^m \to C$. Таким образом, если n функций $f^m{}_i$ переводят элементы одного множества A^m в множество B, функция f^n переводит элементы множества B^n в элементы множества C, то функция g^m переводит элементы множества A^m в элементы множества C.

Будем говорить, что функция g получается путем подстановки функций $f^m{}_1, \ldots, f^m{}_n$ в функцию f^n.

Такая подстановка, в которой области определения всех функций $f^m{}_i$ совпадают и число аргументов функции f^n равно числу функций $f^m{}_i$, называется регулярной.

Свойсво1 операции подстановки. Операция подстановки сохраняет интуитивную вычислимость: если функция g^m получена подстановкой функций $f^m{}_1, \ldots, f^m{}_n$ в функцию f^n и функции $f^m{}_1, \ldots, f^m{}_n$, f^n интуитивно вычислимы, то и функция g^m интуитивно вычислима. Это видно из следующего алгоритма:

1.Вычислить значения функций $f^m{}_1, \ldots, f^m{}_n$ в заданной точке исходного множества A^m.

2.Подставить полученные n значений в функцию f^n и вычислить ее значение.

3.Значением функции g^m считать значение функции f^n.

4.Повторить п.1-3 для всех точек области определения функций $f^m{}_1, \ldots, f^m{}_n$.

Свойсво2 операции подстановки. Операция подстановки сохраняет всюду определенность функций: если функция g^m получена подстановкой функций $f^m{}_1, \ldots, f^m{}_n$ в функцию f^n и функции $f^m{}_1, \ldots, f^m{}_n$, f^n всюду определены, то и функция g^m всюду определена. Покажем это. Если функции $f^m{}_1, \ldots, f^m{}_n$ всюду определены, то для любой комбинации значений входного вектора $(x_1, \ldots, x_m)$ эти функции принимают определенное значение. n значений функций $f^m{}_1, \ldots, f^m{}_n$ образуют входной вектор для функции f^n. Так как функция f^n всюду определена, то для любого входного вектора существует определенное значение, являющееся значением функции g^m. Т.е. функция g^m всюду определена, так как для любого входного вектора $(x_1, \ldots, x_m)$ существует ее значение.

Случай 2. Нерегулярная подстановка.

Рассмотрим следующие примеры подстановок.

Пример1.

$g^2 : g(x, y) = f^2(f_1(x), f_2(y))$.

Здесь функция $f_1(x)$ определена как унарная функция вида f_1: $X \to X$ функция; $f_2(y)$ определена как унарная функция вида f_2: $Y \to Y$. Тогда функция g^2 есть бинарная функция вида g^2: $X \times Y \to Z$. Здесь базовые функции $f_1(x)$, $f_2(y)$ принимают значения на разных множествах, поэтому окончательная функция g^2 определена как функция от двух переменных. Такая подстановка возможна, но она не является регулярной.

Пример2.

$h^1 : h^1(x) = f^2(x, f_1(x))$,

Здесь функция $f_1(x)$ определена как унарная функция вида f_1: $X \to X$ функция. Функция f^2 от двух переменных при подстановке функции $f_1(x)$ стала зависеть от одной переменной x. Поэтому функция h^1 есть унарная функция вида h^1: $X \to Y$. Такая подстановка возможна, но она не является регулярной.

Пример3.

$\varphi^3 : \varphi(x, y, z) = f(f_1(x), f_2(x))$.

Здесь функция $f_1(x)$ определена как унарная функция вида f_1: $X \to X$ функция; $f_2(x)$ также определена как унарная функция вида f_2: $X \to Y$. Здесь базовые функции $f_1(x)$, $f_2(x)$ принимают значения на двух разных множествах и по определению функция φ должна быть функцией от двух переменных. Однако, в представленной записи функции $\varphi(x, y, z)$ введен фиктивный аргумент z, поэтому окончательно функция φ^3 определена как функция от трех переменных. Тогда функция φ^3 есть тернарная функция вида φ^3: $X \times Y \times Z \to N$. Такая подстановка возможна, но она не является регулярной.

Можно привести и другие примеры нерегулярных подстановок. Ограничимся этими тремя примерами. Отметим, что свойства 1 и 2 справедливы и для нерегулярных подстановок.

Задачи и упражнения.

1.Функция g^3 (x_1, x_2, x_3) получена из функции f^3 (x_3, x_1, x_2) перестановкой аргументов. Записать эту перестановку с помощью операции подстановки.

2.Какой вид будет иметь функция g^2 $(f^1(x_1), f^2(x_1, x_3))$ после подстановки в нее функций вида $f^1(x_1) = ax_1^2 + bx_1x_3 + c$,

$f^2(x_1, x_3) = ax_1^2 + bx_1 + cx_3$,

$g^2 (x_1, x_2) = x_1 - x_2$.

4.5.2.Операция примитивной рекурсии.

Определение 4.3. Рекурсией называется способ задания функции, при котором значения определяемой функции для произвольных значений аргументов выражаются через значения определяемой функции для меньших значений аргументов. Примитивная рекурсия является одним из простейших видов рекурсии.

Пусть заданы: g – n – местная функция; h – $(n+2)$ – местная функция. Будем говорить, что $(n+1)$- местная функция f возникает из функций g и h с помощью операции *примитивной рекурсии,* если для всех натуральных $x_1, \ldots, x_n$, y выполняются следующие равенства:

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n), \qquad (1)$$

$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)). \qquad (2)$$

При $n = 0$ равенства (1), (2) для одноместной функции f будут иметь вид:

$f(0) = c$,

$f(y+1) = h(y, f(y))$.

Здесь c- константа для g.

Как показано в [10] для любых функций g и h соответственно от n и $(n+2)$ переменных на множестве всех натуральных чисел существует единственная функция f от $(n+1)$ переменных. Ее можно построить путем последовательного применения операции примитивной рекурсии:

$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$,

$f(x_1, \ldots, x_n, 1) = h(x_1, \ldots, x_n, 0, f(x_1, \ldots, x_n, 0))$,

. . . .

$f(x_1, \ldots, x_n, m+1) = h(x_1, \ldots, x_n, m, f(x_1, \ldots, x_n, m))$.

Таким образом, с помощью операции примитивной рекурсии можно строить новые функции f, если известны и существует способ вычисления функций g и h. Для нахождения значения функции f в точке $(x_1 = a_1, \ldots, x_n = a_n, y = m)$ достаточно последовательно вычислить числа:

$b_0 = g(a_1, \ldots, a_n)$,

$b_1 = h(a_1, \ldots, a_n, 0, b_0)$,

$b_2 = h(a_1, \ldots, a_n, 1, b_1)$,

…….

$b_{m+1} = h(a_1, \ldots, a_n, m, b_m)$.

Полученные числа b_0, b_1, …., b_{m+1},… образуют значение функции f по переменной $y \in N = \{0, 1, \ldots\}$ при фиксированных значениях вектора $(x_1, \ldots, x_n)$.

4.5.3.Операция минимизации.

Рассмотрим некоторую n – местную числовую функцию f, для которой существует правило вычисления ее значений. При этом будем предполагать, что значение функции f не определено тогда и только тогда, когда для заданных исходных данных это правило работает бесконечно, не выдавая никакого определенного результата. Зафиксируем первые $(n-1)$ аргументов функции некоторыми своими значениями $x_1 = a_1, \ldots, x_{n-1} = a_{n-1}$ и рассмотрим уравнение вида

$$f(a_1, \ldots, a_{n-1}, x_n) = k, \qquad (3)$$

относительно переменной x_n, где k – const, некоторое фиксированное число. Решением этого уравнения будет такое значение переменной x_n, при котором приведенное равенство (3) выполняется.

Будем искать решение этого уравнения при помощи существующего правила вычисления значений функции f путем последовательного вычисления $f(a_1, \ldots, a_{n-1}, x_n)$ для $x_n = 0, 1, 2, \ldots$. Наименьшее значение $x_n = a$, для которого выполнится равенство (3)

$f(a_1, \ldots, a_{n-1}, a) = k$,
назовем решением уравнения (3) и обозначим
$a = \mu_{x_n} (f(a_1, \ldots, a_{n-1}, x_n) = k)$. (4)

Данная запись означает, что a есть результат некоторого процесса μ решения уравнения (3).

Описанный процесс (4) не существует или будет продолжаться бесконечно долго в следующих случаях:

- значение функции $f(a_1, \ldots, a_{n-1}, 0)$ не определено;
- значения функции $f(a_1, \ldots, a_{n-1}, x_n)$ для $x_n = 0, 1, \ldots, (a-1)$ определены, но отличны от k, а значение $f(a_1, \ldots, a_{n-1}, a)$ не определено;
- значения функции $f(a_1, \ldots, a_{n-1}, x_n)$ определены для всех $x_n = 0, 1, \ldots$, но отличны от k.

Во всех выше перечисленных случаях выражение (4) считается неопределенным. Во всех остальных случаях этот процесс обрывается и дает наименьшее решение $x_n = a$ уравнения (3).

Например, для функции $f(x, y) = x - y$ при $x = 10$, $k = 2$ имеем

$a = \mu_y (f(x, y) = k) = \mu_y (10 - y = 2)$. $a = 8$.

Если не фиксировать значение переменной x, то выражение

$\mu_y (f(x, y) = k)$

есть функция от x. Эту функцию будем обозначать Mf, где M – символ операции минимизации, переводящий функцию f(x, y) в

$M f(x, y) = \mu_y (f(x, y) = k)$.

Для одноместных функций f(x) операция минимизации Mf обращает функцию f: $\mu_x (f(x) = k)$. Это можно записать так $x = f^{-1}(k)$.

4.6.Примитивно рекурсивные функции.

Определение 4.4. Класс функций называется примитивно рекурсивно замкнутым, если он содержит базовые функции $\mathbf{s}^1$, $\mathbf{o}^1$, $\mathbf{I}^n_m$ и замкнут относительно операций подстановки и примитивной рекурсии.

Минимальный примитивно рекурсивно замкнутый класс (содержащийся в любом другом примитивно рекурсивно замкнутом классе) называется классом примитивно рекурсивных функций (*П*), а его элементы называются примитивно рекурсивными функциями.

Определение. Функция f называется *примитивно рекурсивной*, если она получена конечным числом операций подстановки и примитивной рекурсии, исходя из базовых функций $\mathbf{s}^1$, $\mathbf{o}^1$, $\mathbf{I}^n_m$.

Из определения ясно, что операции подстановки и примитивной рекурсии, примененные к примитивно рекурсивным функциям, дают примитивно рекурсивную функцию.

Рассмотрим несколько примеров.

Базовые одноместные функции $\mathbf{s}^1(x) = x+1$, $\mathbf{o}^1(x) = 0$, $\mathbf{I}^1_1(x) = x$ и многоместные функции

$\mathbf{I}^n_m(x_1, \ldots, x_n) = x_m$ для $m \in \{1, \ldots, n\}$, $n = 1, 2, \ldots$ - примитивно рекурсивны по определению.

Функция $f(x) = const = c$ допускает представление в виде терма:

$\mathbf{s}^1(\mathbf{s}^1(\ldots.\mathbf{s}^1(\mathbf{s}^1(\mathbf{o}^1 \ (x))) \ldots.))$,

где операция подстановки выполняется с раз. Отсюда можно сделать вывод, что все постоянные функции примитивно рекурсивны.

Рассмотрим двухместную функцию $f^2(x, y) = x + y$. Ее можно представить с помощью операции примитивной рекурсии:

$f(x, 0) = g(x) = \mathbf{I}^1_1(x)$,

$f(x, y+1) = h(x, y, f(x, y)) = f(x, y) +1$, или

$x + 0 = x$,

$x + (y + 1) = (x + y) + 1 = \mathbf{s}^1(x + y)$.

Функция $f^2(x, y) = x + y$ примитивно рекурсивна, так как возникает из примитивно рекурсивных функций $\mathbf{I}^1_1(x)$, $h(x, y, f(x, y)) = f(x, y) +1$ с помощью операции примитивной рекурсии.

Двухместную функцию $f^2(x, y) = x * y$ можно представить с помощью операции примитивной рекурсии:

$f(x, 0) = g(x) = \mathbf{o}^1(x)$,

$f(x, y+1) = h(x, y, f(x, y)) = f(x, y) + x$, или

$x * 0 = 0$,

$x * (y + 1) = (x * y) + x$.

Функция $f^2(x, y) = x * y$ примитивно рекурсивна, так как возникает из примитивно рекурсивных функций $\mathbf{o}^1(x)$, $h(x, y, f(x, y)) = f(x, y) + x$ с помощью операции примитивной рекурсии.

Определим на N функцию знака (sg(x) – читается как сигнум x):

$$sg(x) = \begin{cases} 0, \text{ если } x = 0, \\ 1, \text{ если } x > 0. \end{cases}$$

Функция $sg(x)$ является примитивно рекурсивной, так как может быть представлена с помощью операции примитивной рекурсии:

$sg(0) = g(0) = 0$,

$sg(x + 1) = h(x, f(x)) = 1 - sg(x)$.

Введем функцию, обратную функции знака:

$$\underline{sg}(x) = \begin{cases} 1, \text{ если } x = 0, \\ 0, \text{ если } x > 0. \end{cases}$$

Как видно из определений этих функций, для них имеет место равенство

$\underline{sg}(x) = 1 - sg(x)$.

Теорема. Пусть n –местная функция g – примитивно рекурсивна. Тогда n – местная функция f, определяемая равенством

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{x_n} g(x_1, \ldots, x_{n-1}, i),$$

примитивно рекурсивна.

Доказательство.

Из заданного равенства вытекает

$f(x_1, \dots, x_{n-1}, m) = g(x_1, \dots, x_{n-1}, 0) + g(x_1, \dots, x_{n-1}, 1) + \dots + g(x_1, \dots, x_{n-1}, m)$.

Или это можно переписать

при $x_n = 0$ $f(x_1, \dots, x_{n-1}, 0) = g(x_1, \dots, x_{n-1}, 0)$,

при $x_n = 1$ $f(x_1, \dots, x_{n-1}, 1) = g(x_1, \dots, x_{n-1}, 0) + g(x_1, \dots, x_{n-1}, 1) =$
$f(x_1, \dots, x_{n-1}, 0) + g(x_1, \dots, x_{n-1}, 1)$,

… … … …

при $x_n = m$ $f(x_1, \dots, x_{n-1}, m) = g(x_1, \dots, x_{n-1}, 0) + \dots + g(x_1, \dots, x_{n-1}, m) =$
$f(x_1, \dots, x_{n-1}, m-1) + g(x_1, \dots, x_{n-1}, m)$.

Как видно из представленных выражений функция f получена посредством операции примитивной рекурсии из примитивно рекурсивных функций:

$g(x_1, \dots, x_{n-1}, 0)$,

$h(x_1, \dots, x_{n-1}, y, z) = f(x_1, \dots, x_{n-1}, y-1) + z$, где $z = g(x_1, \dots, x_{n-1}, y)$.

Поэтому функция f примитивно рекурсивна. Конец доказательства.

Задачи и упражнения.

1.Функция P задается следующим образом:

$P_0(a, x) = a + x$, $P_1(a, x) = ax$, $P_2(a, x) = a^x$. Функции P_0, P_1, P_2 связаны соотношениями:

$P_1(a, x + 1) = P_0(a, P_1(a, x))$, $P_1(a, 1) = a$,

$P_2(a, x + 1) = P_1(a, P_2(a, x))$, $P_2(a, 1) = a$. Продолжая эту цепочку для n = 2, 3, … получим

$P_{n+1}(a, x + 1) = P_n(a, P_{n+1}(a, x))$, $P_{n+1}(a, 1) = 1$,

$P_{n+1}(a, 0) = 1$, n = 1, 2, …

Требуется построить функцию P для n = 3, 4, 5. Функция P называется быстро растущей функцией.

2.Показать, что следующие функции являются примитивно рекурсивными:

а) $f(x, y) = x^y$, $x^0 = 1$;

б) функция усеченной разности $f(x, y) = x \div y = [(x - y)$, если $x \geq y]$ либо $[x \div y = 0$, если $x < y]$;

в) $f(x, y, z) = x + y*z$;

г) $f(x, y, z) = (x + y)*z$;

д) $f(x, y, z) = (x + y)^z$.

4.6.1.Примитивно рекурсивные множества.

В предыдущем разделе мы рассмотрели множество, называемое замкнутым классом примитивно рекурсивных функций (*П*). Все функции этого класса являются всюду определенными, т.е. в каждой точке на всей области определения они имеют конкретное значение.

А что можно сказать про подмножества различных множеств. Рассмотрим, например, такие подмножества: область определения примитивно рекурсивной функции, область значений функции, произвольное подмножество M, принадлежащее некоторому множеству N^S.

Определение. Множество M, являющееся подмножеством некоторого универсального множества U ($M \subseteq U$), называется примитивно рекурсивным, если его характеристическая функция примитивно рекурсивна.

С точки зрения теории алгоритмов изучение этих множеств интересно решением следующей задачи. Требуется определить входит ли число a в множество A или нет. Для ее решения требуется построить алгоритм вычисления значения характеристической функции множества A. Оказывается, в силу тезиса Черча, существование такого алгоритма равносильно рекурсивности характеристической функции множества A.

Рассмотрим такие множества и их свойства.

Характеристическими функциями пустого множества $\varnothing$ и множества всех натуральных чисел N являются постоянные одноместные функции 1 и 0. Эти функции примитивно рекурсивны, поэтому и множества $\varnothing$ и N примитивно рекурсивны.

Характеристической функцией для конечного множества чисел $\{a_1, a_2, \ldots, a_n\}$ служит примитивно рекурсивная функция

$$sg\,(\,|x-a_1|*|x-a_2|*\ldots*|x-a_n|\,).$$

Поэтому каждое конечное множество натуральных чисел примитивно рекурсивно.

Теорема 1. Дополнение примитивно рекурсивного множества, а также объединение и пересечение любой конечной системы примитивно рекурсивных множеств, есть примитивно рекурсивные множества.

В качестве доказательства рассмотрим характеристические функции конечной системы множеств. Обозначим, $f_1(x)$, $f_2(x)$, …, $f_n(x)$ – характеристические функции примитивно рекурсивных множеств A_1, A_2, …, A_n. Тогда соответственно для дополнения, объединения и пересечения характеристические функции будут иметь вид:

$F_{доп}(x) = \underline{sg}\,(f_1(x))$;

$F_{об}(x) = f_1(x)*f_2(x)*\ldots*f_n(x)$;

$F_{пер}(x) = sg\,(f_1(x) + f_2(x) + \ldots + f_n(x))$.

Если функции $f_1(x)$, $f_2(x)$, …, $f_n(x)$ примитивно рекурсивны, то и построенные функции также будут примитивно рекурсивными.

Теорема 2. Если всюду определенная функция f (x) примитивно рекурсивна, то множество **A** решений уравнения

$$f\,(x) = 0$$

примитивно рекурсивно.

Действительно, характеристической функцией множества **A** является функция $\underline{sg}\,(f\,(x))$, примитивно рекурсивная вместе с функцией f (x).

В общем случае, совокупность всех значений, принимаемых примитивно рекурсивной функцией, может и не быть примитивно рекурсивным множеством и даже может не быть рекурсивным множеством. Следующая теорема выделяет примитивно рекурсивное множество для функции f (x):

Теорема 3.Если примитивно рекурсивная функция f (x) удовлетворяет условию f (x) >= x, для x = 0, 1, 2, …, в частности, если f (x) монотонно возрастает, то совокупность **A** всех значений функции f (x) есть примитивно рекурсивное множество.

Действительно, из неравенства f (y) >= y следует, что характеристическая функция множества **A**

$$\chi(x) = \prod_{y \in \{0, \ldots, x\}} sg\,(\,|\,f\,(y) - x\,|\,)$$

является примитивно рекурсивной.

Для примитивно рекурсивных функций и примитивно рекурсивных множеств выполняется следующее утверждение: множество тогда и только тогда примитивно рекурсивно, когда его характеристическая функция примитивно рекурсивна.

4.7.Частично рекурсивные функции.

Определение 4.5. Класс функций называется частично-рекурсивно замкнутым, если он содержит базовые функции $\mathbf{s}^1$, $\mathbf{o}^1$, $\mathbf{I}^n_m$ и замкнут относительно операций подстановки, примитивной рекурсии и операции минимизации μ.

Минимальный частично-рекурсивно замкнутый класс (содержащийся в любом другом частично-рекурсивно замкнутом классе) называется классом частично-рекурсивных функций (*Ч*), а его элементы называются частично-рекурсивными функциями.

Определение 4.6. Функция f называется *частично-рекурсивной*, если она может быть получена из базовых функций **o**, **s,** $\mathbf{I}^n_m$ конечным числом операций подстановки, примитивной рекурсии и минимизации.

Из определения вытекают следующие свойства.

1.Класс частично рекурсивных функций шире класса примитивно рекурсивных функций за счет операции минимизации.

2.Операции подстановки, примитивной рекурсии и минимизации, произведенные над частично рекурсивными функциями, вновь дают частично рекурсивные функции.

Понятие частично рекурсивной функции – одно из главных понятий в теории алгоритмов. Роль этого понятия связана с алгоритмическим понятием вычислимости. С одной стороны, всякая частично рекурсивная функция вычислима, т.е. существует правило или механическая процедура, с помощью которой за конечное число шагов можно вычислить ее значение. Собственно частично рекурсивные функции мы и строили так, чтобы они вычислялись. С другой стороны, какую бы механическую процедуру, приводящую к вычислению, мы не формировали, мы всегда можем утверждать, что с помощью построенной процедуры вычисляется функция, являющаяся частично рекурсивной. Это и отражено в гипотезе, называемой тезисом Черча:

Класс алгоритмически вычислимых числовых функций совпадает с классом всех частично-рекурсивных функций.

4.7.1. Рекурсивно перечислимые множества.

Аналогом примитивно рекурсивного множества для примитивно рекурсивных функций вводится понятие рекурсивно-перечислимого множества для частично-рекурсивных функций.

Определение 4.7. Множество **A** называется рекурсивно перечислимым, если существует двуместная примитивно рекурсивная функция f (a, x), такая, что уравнение f (a, x) = 0 имеет решение x тогда и только тогда, когда a ∈ **A.**

Следствие. Каждое примитивно рекурсивное множество рекурсивно перечислимо.

Действительно, характеристическая функция f (x) произвольного примитивно рекурсивного множества **A** является примитивно рекурсивной. Рассмотрим уравнение

f (a) + x = 0.

Оно имеет решение x тогда и только тогда, когда a ∈ **A**, где **A** по определению рекурсивно перечислимое множество**.**

Приведем без доказательства следующую теорему: непустое множество **A** тогда и только тогда рекурсивно перечислимо, когда оно совпадает с совокупностью всех значений некоторой примитивно рекурсивной функции.

Понятие рекурсивно перечислимого множества служит уточнением интуитивного понятия перечислимого множества. Множество **A** называется перечислимым, если существует алгоритм (функция), порождающий элементы множества **A**. Это алгоритм, позволяющий для любого элемента множества **A** положительно ответить на вопрос **a** ∈ **A**. Подчеркнем, что только положительный ответ определяет перечислимое множество **A**. Отрицательный ответ означает бесконечную работу алгоритма. Но бесконечная работа алгоритма трактуется неоднозначно: либо **a** ∉ **A**, либо **a** ∈ **A**, но алгоритм ещё не закончил свою работу.

Для решения аналогичных задач вводится понятие общерекурсивного множества.

Определение 4.8. Множество **A** из некоторого универсума **U** называется общерекурсивным, если оно само **(A)** и его дополнение **(U\A)** являются рекурсивно-перечислимыми множествами. Понятие общерекурсивного множества служит уточнением интуитивного понятия разрешимого множества. В этом случае существует функция f, порождающая элементы множества **A**, а также функция g, порождающая элементы множества **U\A**. Поэтому, для любого элемента **a** выполняется либо **a** ∈ **A**, либо **a** ∈ **U\A**, т.е. принадлежность элемента **a** множеству **A** идентифицируется однозначно.

4.7.2. Рекурсии 2-й ступени.

В данном разделе покажем, что класс частично-рекурсивных функций шире класса примитивно рекурсивных функций. Для этого построим общерекурсивную функцию, не являющуюся примитивно рекурсивной.

Как задать операцию примитивной рекурсии по двум переменным для функции от двух аргументов F (x, y).

При рассмотрении операции примитивной рекурсии мы существенно пользуемся естественной упорядоченностью на множестве натуральных чисел: 0, 1, 2, … Для рассмотрения этой операции по двум переменным, мы должны каким-то образом предварительно упорядочить все пары чисел (x, y). Это можно сделать по разному. Выберем для определенности алфавитный порядок. По определению алфавитного порядка пара (x, y) предшествует паре (x_1, y_1), если $(x < x_1)$ либо $((x = x_1)$ и $(y < y_1))$. Выпишем пары в алфавитном порядке.

$$(0, 0) < (0, 1) < \ldots < (1, 0) < (1, 1) < \ldots < (2, 0) < (2, 1) < \ldots \qquad (1)$$

В таком порядке пара (0, 0) является начальной. Пары вида (n, 0) играют особую роль. У них нет непосредственно предшествующей пары.

Перейдем к рекурсивному представлению функции F (x, y) по двум аргументам. Исходя из заданного порядка, мы должны выразить функцию F (x, y) в точке (x, y) через значения функции F (u, v) для пар (u, v), предшествующих паре (x, y).

Здесь возможны следующие случаи:

1) Требуется найти F (0, y). Предыдущие значения функции имеют вид F(0, v), где v < y.
2) Требуется найти F (x +1, 0). Предыдущие значения имеют вид F (u, y), где u =< x + 1, а y произвольно. В частности, если предыдущие значения функции известны, то можно считать известными и значения следующих выражений:

$$\begin{aligned}&F(f_1(x), g(y)),\\ &F(f_1(x), F(f_2(x), g(y))),\\ &F(f_1(x), F(f_2(x), F(f_3(x), y))), \qquad (2)\\ &\ldots\ldots\ldots\ldots\ldots\ldots \quad \ldots\ldots\ldots\end{aligned}$$

Здесь $f_i(x)$, $g(y)$ – наперед заданные функции, удовлетворяющие условиям

$$f_i(x) =< x,\ i = 1, 2, \ldots . \qquad (3)$$

3) Требуется найти F (x +1, y +1). Предыдущие значения функции могут быть следующих типов:

$$\begin{aligned}&F(x+1, f_1(y)),\\ &F(f_1(x), F(f_2(x), g(y))),\\ &F(f_1(x), F(x+1, f_2(y))), \qquad (4)\\ &\ldots\ldots\ldots\ldots \quad \ldots\ldots\ldots\ldots\end{aligned}$$

Таким образом, чтобы задать функцию F (x, y) рекурсией по двум переменным, мы должны задать число F (0, 0), функцию одной переменной F (0, y) и выражения для функций: F (x, 0), F (x, y) через выражения вида (2), (3), (4).

Определение рекурсии 2-й ступени. Пусть заданы всюду определенные функции

$$G(x_0, x_1,\ldots, x_m),\ H(x_0, x_1,\ldots, x_k),\ h(x)$$

и всюду определенные функции $f_i(x)$, $g_j(x)$, удовлетворяющие требованиям

$$f_i(x) =< x,\ g_j(x) =< x,\ i \in \{1,..,m\},\ j \in \{1,..,k\}. \qquad (5)$$

Будем говорить, что всюду определенная функция F (x, y) возникает из функций G, H, h, f_i, g_j рекурсией 2-й ступени, если для всех x, y выполняются равенства:

$F(x + 1, y + 1) = G(x + 1, y, F(f_1(x), F(x + 1, y)), F(f_2(x), F(f_3(x), y + 1)),$

$F(f_4(x), y + 1), F(f_5(x), y + 1), \ldots\ldots\ldots, F(f_m(x), y + 1),).$ (6)

$F(x + 1, 0) = H (x + 1, F(g_1(x), F(g_2(x), 0)), F(n, 0), F (g_3(x), 0), F (g_4(x), 0), \ldots,$
$F (g_k(x), 0)).$

$F(0, y) = h(y).$

Покажем, что каковы бы ни были функции G, H, h и функции $f_i (x)$, $g_j (x)$, удовлетворяющие условиям (5), существует единственная функция F (x, y), удовлетворяющая схеме (6).

По определению для всех y полагаем $F(0, y) = h (y)$ (формула 3 из схемы (6)).

Если для всех пар $(u, v) < (x + 1, 0)$ значения F (u, v) определены и для них соблюдены условия (6), то

$F(x + 1, 0) = H (x + 1, F(g_1(x), F(g_2(x), 0)), F(n, 0), F (g_3(x), 0), F (g_4(x), 0), \ldots,$
$F (g_k(x), 0))$ (формула 2 из схемы (6)) .

Если для пар $(u, v) < (x + 1, y +1)$ значения F (u, v) определены в соответствии со схемой (6), тогда

$F (x + 1, y + 1) = G (x + 1, y, F(f_1(x), F(x + 1, y)), F (f_2(x), F(f_3(x), y + 1)),$
$F(f_4(x), y + 1), F(f_5(x), y + 1),\ldots\ldots, F(f_m(x), y + 1),)$ (формула 1 из схемы (6)).

Полученная таким образом функция F (x, y) по построению удовлетворяет условиям (6). С другой стороны, поскольку значения функции F (x, y) строились по формулам (6), то эти формулы определяют функцию F (x, y) однозначно.

Теорема о рекурсии 2-й ступени. Если указанные в рекурсии 2-й ступени функции G, H, h, f_i, g_j примитивно рекурсивны, то возникающая из них по схеме (6) функция F (x, y) общерекурсивна.

Рассмотрим эту теорему для m=4, k=3. Рассмотрим график функции F (x, y). По определению графиком **M** функции называется совокупность троек вида (x, y, F (x, y)), т.е. **M** = {(x, y, F (x, y))}. Для доказательства теоремы достаточно показать, что множество **M** рекурсивно перечислимо.

Формула 1 из схемы (6) относительно множества **M** может быть перефразирована следующим образом:

Если тройки
$(x +1, y, u),$
$(f_1(x), u, v),$
$(f_3(x), y + 1, p),$ (7)
$(f_2(x), p, q),$
$(f_4(x), y + 1, w)$ принадлежат **M**, то тройка
$(x + 1, y + 1, G(x +1, y, v, q, w)) \in$ **M**. (8)

Формула 2 из схемы (6) относительно множества **M** может быть перефразирована следующим образом:

Если тройки
$(g_2(x), 0, u),$
$(g_1(x), u, v),$ (9)
$(x, 0, w),$
$(g_3(x), 0, p)$, принадлежат **M**, то тройка
$(x + 1, 0, H (x +1, v, w, p)) \in$ **M**. (10)

Формула 3 из схемы (6) относительно множества **M** может быть перефразирована следующим образом: множество **M** содержит все тройки вида

(0, x, h(x)).

Обозначим через $\mathbf{N^3}$ совокупность всех числовых троек. Определим на $\mathbf{N^3}$ такие операции:

- $\alpha(u_1, u_2, u_3, u_4, u_5)$, которая для наборов троек (7) формирует значение вида (8);

-$\beta(u_1, u_2, u_3, u_4)$, которая для наборов троек (9) формирует значение вида (10).

Полагаем по определению

$\alpha<(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4), (x_5, y_5, z_5)> =$

$= < x_1, y_3, G(x_1, y_1, z_2, z_4, z_5)>,$ (11)

если выполнены условия:

$x_2 = f_1(x_1 \div 1), y_2 = z_1, x_3 = f_3(x_1 \div 1), y_3 = y_1 + 1,$ (12)

$x_4 = f_2(x_1 \div 1), y_4 = z_3, x_5 = f_4(x_1 \div 1), y_5 = y_1 + 1;$

$\alpha<(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4), (x_5, y_5, z_5)> = <0, 0, h(0)>,$ (13)

если условия (12) не выполнены.

Аналогично полагаем по определению

$\beta<(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)> =$

$<x_3 + 1, 0, H(x_3 + 1, z_2, z_3, z_4)>,$ (14)

если выполнены условия:

$x_1 = g_2(x_3), x_2 = g_1(x_3), y_2 = z_1, x_4 = g_3(x_3),$ (15)

$\beta<(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)> = <0, 0, h(0)>,$ (16)

если условия (15) не выполнены.

Функции G, H, h, f_i, g_j примитивно рекурсивны. Известно, для того, чтобы операция, определенная на $\mathbf{N^n}$, была примитивно рекурсивной, необходимо и достаточно, чтобы таковыми были координатные функции этой операции. Именно такими и являются операции α, β, т.е. они являются примитивно рекурсивными операциями.

Обозначим через $\mathbf{M_0}$ совокупность троек вида $< 0, x, h(x)>$. Известно, для того, чтобы совокупность n-ок, (в частности, n=3), была рекурсивно перечислимой необходимо и достаточно, чтобы она была совокупностью n–ок вида

$<f_1(x), \ldots, f_n(x)>$,

где $f_1(x), \ldots, f_n(x)$- примитивно рекурсивные функции. Из примитивной рекурсивности функции h(x) следует, что совокупность $\mathbf{M_0}$ рекурсивно перечислима.

Теорема о порождаемых совокупностях гласит: пусть на множестве $\mathbf{N^n}$ всех n-ок натуральных чисел задано конечное число примитивно рекурсивных операций и некоторое рекурсивно перечислимое множество $\mathbf{M_0}$. Тогда совокупность $\mathbf{M^*}$ n-ок, порожденная элементами $\mathbf{M_0}$ с помощью заданных операций, также рекурсивно перечислимо.

Поэтому, совокупность всех троек $\mathbf{M^*}$, которые можно получить из троек $\mathbf{M_0}$ с помощью операций α, β, является рекурсивно перечислимой и надо лишь показать, что эти множества совпадают.

Заметим, что **M** содержит порождающее множество $\mathbf{M_0}$. Далее, условия (11), (13) и (14), (16) показывают, что совокупность **M** замкнута относительно операций α, β. Совокупность $\mathbf{M}^*$ по определению есть наименьшая совокупность, замкнутая относительно операций α, β и содержащая $\mathbf{M_0}$. Таким образом, $\mathbf{M}^* \subseteq \mathbf{M}.$

Обратное включение $\mathbf{M} \subseteq \mathbf{M}^*$ можно доказать индукцией по параметрам x, y для троек $<x, y, F(x, y)>$. Для $x=0$ график **M** содержит лишь тройки $<0, y, h(y)>$. Все они входят в $\mathbf{M}^*$. Пусть для некоторого x все тройки $<r, y, F(r, y)>$, удовлетворяющие условию $r =< x$ входят в $\mathbf{M}^*$. Тогда и тройка $<x + 1, 0, F(x+1, 0)>$ входит в $\mathbf{M}^*$. Действительно, положим

$u= F(g_2 (x), 0)$, $v= F(g_1 (x), u)$, $w = F(x, 0)$, $p = F (g_3 (x), 0)$

и в силу (6) имеем

$$F (x + 1, 0) = H (x + 1, v, w, p).$$

Так как $g_i (x) =< x$, то все тройки (9) по предположению входят в $\mathbf{M}^*$. Применяя к тройкам (9) операцию β получим тройку $<x + 1, 0, F (x + 1, 0)>$. Совокупность $\mathbf{M}^*$ замкнута относительно операции β. Следовательно, тройка $<x + 1, 0, F (x + 1, 0)>$ принадлежит $\mathbf{M}^*$. Аналогично доказывается случай для троек $<r, s, F (r, s)>$ с условием $(r, s) < (x +1, y+1)$. Если все тройки вида $<r, s, F (r, s)>$ с условием $(r, s) < (x +1, y+1)$ содержится в $\mathbf{M}^*$, то и тройка $<x +1, y+1, F (x + 1, y+1)>$ содержится в $\mathbf{M}^*$. Тем самым выполняется включение $\mathbf{M} \subseteq \mathbf{M}^*$ и следовательно, равенство $\mathbf{M} = \mathbf{M}^*$.

5.Алгоритмы и машины Тьюринга.

Понятие алгоритма – одно из основных понятий математики, возникшее задолго до появления ЭВМ. Интуитивно под алгоритмом решения некоторого класса задач понимают набор точных предписаний конечного набора действий, выполнение которых приводит к решению конкретной задачи из рассматриваемого класса.[8].

Однако, такое определение алгоритма не является точным, строгим и однозначным. В рамках этого определения можно описать многие процессы, не являющиеся алгоритмами.

Пример. Опишем процедуру изучения книги по программированию, предложенной в [9]. Пусть книга содержит k – глав.

Начало примера.

1.Установить i =1.

2.Начать чтение главы i.

3.Интересен ли предмет этой главы. Если да, перейти к шагу 5, если нет – к шагу 4.

4.Отмечена ли эта глава как обязательная для изучения. Если да, то перейти к шагу 5, если нет, перейти к шагу 6.

5.Прочитать главу i.

6.Увеличить i на 1. Если $i \le k$, то перейти к шагу 2, если $i > k$, перейти к шагу 7.

7.Немного передохнув, перейти к шагу 1.

Конец примера.

Представленная процедура удовлетворяет интуитивному определению алгоритма. В ней есть семь предписаний, каждое из которых содержит конечный набор действий. Однако – это не алгоритм. Во первых, процедура описывает бесконечный процесс. Правда это легко исправить и сделать процесс конечным путем изменения описания шага 7. Во вторых, в ее описании указаны такие действия, как «Прочитать главу i », выяснить «Интересен ли предмет этой главы», «Немного передохнуть». Эти действия не являются четко определенными, так как зависят не только от читаемой книги, но и от читателя и даже от его настроения. Что значит «Немного передохнуть» - 15 минут, один час или отложить чтение книги до сессии. Для строгого формального определения алгоритма необходимо строго формально определить объекты, над которыми производятся действия в алгоритме, а также формально определить сами эти действия и порядок их выполнения.

В качестве строгого формального определения данных воспользуемся их изображением, заданным в некотором алфавите.

Введем алфавит $A_n = \{a_1, a_2, \ldots, a_n\}$. Расширим его еще одной, так называемой, несобственной буквой a_0 и рассмотрим алфавит $\underline{A}_n = A_n \cup \{a_0\}$.

5.1.Неформальное определение машины Тьюринга.

Для формализации действий над объектами (данными, словами в заданном алфавите) и порядка их выполнения, или для формального определения алгоритма вводится понятие машины Тьюринга (МТ), созданной А. Тьюрингом в 1936г.

МТ состоит из:

-ленты, бесконечной длины, разделенной на ячейки;

-головки, перемещающейся вдоль ленты от ячейки к ячейке.

В каждой ячейке ленты может находиться одна из букв расширенного алфавита $\underline{A}_n$. Если в ячейке содержится буква a_0, то данная ячейка считается пустой.

МТ работает в дискретные моменты времени $t_0, t_1, t_2, \dots , t_i, \dots$.

Головка МТ в каждый момент времени располагается над одной из ячеек ленты. Головка МТ может прочитать букву, находящуюся в этой ячейке, или распознать ее относительно других букв заданного алфавита.

Машина Тьюринга в каждый момент времени находится в некотором состоянии s_i. Обозначим, $S = \{s_0, s_1, s_2, \dots, s_q\}$- конечное полное множество всех дискретных состояний, в которых может находиться МТ. Выделим на множестве состояний одно из них s_0, называемое начальным состоянием. Его содержательный смысл понятен – это состояние, находясь в котором МТ начинает работу.

Перед началом работы МТ на ее ленту записываются исходные данные. В нашем понимании это одно или несколько слов из расширенного алфавита $\underline{A}_n$. Число этих слов определяется алгоритмом, который будет выполнять МТ в процессе своей работы. Буквы, входящие в каждое слово, записываются в последовательных ячейках ленты. Слова отделяются друг от друга несобственной буквой a_0. Ячейки ленты, не занятые исходными данными, а такие ячейки, в силу бесконечности ленты, обязательно будут как справа так и слева от исходных слов, заполняются несобственными буквами, т.е. объявляются пустыми.

Работа МТ.

В каждый дискретный момент времени t_i:

-МТ находится в некотором состоянии $s_i \in S$;

-головка МТ расположена над некоторой ячейкой ленты и читает находящуюся в ней букву a_j;

-МТ выполняет определенную команду из некоторого множества команд.

В результате выполнения команды МТ переходит в новое состояние s_{i+1}. Переход МТ в новое состояние определяется предыдущим состоянием и входным символом или буквой, анализируемой головкой. Головка МТ, в результате выполнения команды, может записать в читаемую ячейку новую букву из того же алфавита или оставить ее без изменения. Сама головка относительно читаемой ячейки, может остаться над этой же ячейкой, либо переместиться к следующей ячейке, расположенной справа либо слева от читаемой.

В начале работы МТ приводится в начальное состояние s_0 и головка помещается над начальной ячейкой. В качестве начальной берется ячейка, содержащая несобственную букву a_0, расположенную либо непосредственно слева от первой буквы первого исходного слова либо непосредственно справа от последней буквы последнего исходного слова.

Обозначим, (s_0, a_0) – пара, где s_0 – начальное состояние МТ, $a_0 \in \underline{A}_n$ – начальная буква, находящаяся в ячейке, над которой находится головка. Пара (s_0, a_0) определяет первую выполняемую МТ команду. В результате ее выполнения образуется новая пара (s_1, a_1) , где $s_1 \in S$– новое состояние МТ, $a_1 \in \underline{A}_n$ – новая буква, записанная в текущей ячейке в результате выполнения этой команды. Головка при этом либо остается на месте, либо смещается вправо или влево относительно читаемого символа. На следующем шаге, в следующий момент времени МТ переходит в новое состояние, а головка опять перемещается. Этот процесс повторяется некоторое число раз. Окончание работы МТ может быть следующим:

1) через конечное число шагов МТ останавливается по команде останова;
2) МТ никогда не останавливается;
3) для МТ возникает пара $(s_к, a_к)$, для которой не определена команда МТ. И в этом состоянии МТ находится бесконечно долго.

В последних двух случаях считается, что: либо некоторые команды написаны неправильно, либо неверны исходные данные, либо и то и другое вместе взятое.

Множество команд, посредством которых МТ переходит из одного состояния в другое над заданными исходными словами, называется программой, позволяющей по исходным словам построить новое слово.

5.2.Формальное определение машины Тьюринга.

Как следует из предыдущего параграфа, каждую команду МТ можно описать набором из пяти элементов

$(s_i, a_i, s_{i+1}, a_{i+1}, v)$, (1)

где – $s_i \in S$, $a_i \in \underline{A}_n$ - соответственно текущее состояние МТ и буква, над которой находилась головка в i-й момент времени;

$s_{i+1} \in S$, $a_{i+1} \in \underline{A}_n$ – соответственно новое состояние МТ и буква, записываемая головкой в текущую ячейку при выполнении команды;

$v \in \{r, l, u\}$ – символ, определяющий направление перемещения головки при выполнении команды. При этом r – означает перемещение головки вправо, l – перемещение головки влево, u – значение символа v, при котором головка остается на месте.

Таким образом, запись (1) означает, что МТ, находясь в состоянии $s_i \in S$ и читая букву $a_i \in \underline{A}_n$ за одну команду перешла в состояние $s_{i+1} \in S$, при этом головка на место буквы a_i записала букву $a_{i+1} \in \underline{A}_n$ сама головка переместилась вдоль ленты по направлению, определяемому значением символа v.

Определение МТ. М а ш и н о й Т ь ю р и н г а назовем четверку объектов

$МТ = (\underline{A}_n, S, s_0, P)$,

где $\underline{A}_n$ – входной и выходной алфавит МТ, состоящий из конечного множества букв;

S- конечное множество состояний;

$s_0 \in S$ – начальное состояние МТ;

P – множество наборов вида (1). Эти наборы определяют следующие функции:

-функцию перехода МТ из одного состояния в другое f_S: $S \times \underline{A}_n \to S$;

-функцию выхода МТ f_A: $S \times \underline{A}_n \to \underline{A}_n$;

-функцию управления головкой $f_V : S \times \underline{A}_n \to v$.

5.3.Работа машины Тьюринга.

Опишем работу МТ на примере сложения двух целых чисел. Для построения такой машины воспользуемся ее определением. В качестве алфавита МТ выберем $\underline{A}_n = A \cup \{\propto\}$. Здесь символ $\propto$ используется как несобственный. Собственно алфавит $A = \{|, +\}$. В этом алфавите всякому целому числу n соответствует n палочек вида $|$. Букву + будем помещать между двумя словами и понимать ее, во первых, как символ сложения двух чисел, а во вторых, как разделитель между двумя словами. Символ запятая (,) не является символом алфавита, так же как и фигурные скобки ({, }). Такие символы как фигурные скобки ({, }) и запятая (,) в алгоритмических языках программирования называются метасимволами и используются как средство описания различных синтаксических конструкций алгоритмического языка.

Пусть в нашем примере первое слово есть целое число m = 3, а второе слово n = 4.

Для описания работы МТ введем следующее вспомогательное определение. Конфигурацией МТ в i-й дискретный момент времени назовем упорядоченную тройку $C_i = (a^*, k, s)$, i = 0, 1, 2, …

где a^* - фрагмент бесконечной ленты, содержащий последовательность слов на ленте в i-й момент времени;

k – расстояние (в смысле числа ячеек) от ячейки ленты, содержащей первую букву первого слова из a^* до текущей ячейки, над которой находится головка;

s – текущее состояние МТ.

Для нашего примера начальная конфигурация МТ будет иметь вид C_0:

k			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a^*	.	.	$\propto$	|	|	|	+	|	|	|	|	$\propto$	$\propto$	$\propto$	$\propto$	.	.

$$a^* = \ldots\propto |||+||||\propto\propto\propto\propto\ldots \qquad (2)$$

$k = 0$

$s = s_0$.

В результате работы МТ после ее успешного останова на ленте должна появиться конфигурация C_{55}:

$$a^* = \ldots \propto \propto \propto \propto \propto \,|\,|\,|\,|\,|\,|\,|\, \propto \ldots \qquad (3)$$

$k = 4$,

$s = s_3$.

Здесь мы предположили, что МТ в процессе своей работы последовательно прошла от конфигурации C_0 до конфигурации C_{55}. Тот факт, что в заключительной конфигурации $k = 4$ означает, что в конце работы МТ головка остановится над ячейкой с этим номером. Закончила работу МТ в заключительном состоянии $s = s_3$.

В продолжении нашего примера нам необходимо ввести множество состояний МТ S, начальное состояние s_0 и множество P. Введем сначала множество P путем задания всех его пятерок:

Инструкция1: s_0 , $\propto$, s_0, $\propto$, r

Инструкция2: s_0 , $|$, s_2, $\propto$, r

Инструкция3: s_0 , +, s_3, $\propto$, u

Инструкция4: s_1 , $\propto$, s_0, $\propto$, r

Инструкция5: s_1 , $|$, s_1, $|$, l

Инструкция6: s_1 , +, s_1, +, l

Инструкция7: s_2 , $\propto$, s_1, $|$, u

Инструкция8: s_2 , $|$, s_2, $|$, r

Инструкция9: s_2 , +, s_2, +, r

Инструкция10: s_3 , $\propto$, s_3, $\propto$, u

Как видно из этого текста множество состояний МТ есть $S = \{s_0, s_1, s_2, s_3\}$, где s_0 – начальное состояние. Нетрудно заметить, что множество P есть набор инструкций работы МТ, реализующий отображения f_S, f_A, f_V, и такой, что МТ из конфигурации C_0 перешла в конфигурацию C_p. Множество P, как набор инструкций, можно назвать программой сложения двух целых чисел в алфавите
$\underline{A}_n = \{|, +\} \cup \{\propto\}$.

Рассмотрим работу этой программы. Начальная конфигурация (2) показывает, что начальное состояние МТ есть s_0 и начальная буква есть $\propto$. Для пары $(s_0, \propto)$ на множестве P допустимой является инструкция1. Ее применение приведет к сохранению пары $(s_0, \propto)$, а головка при этом сместится вправо относительно первой буквы первого слова из a^*. Получится конфигурация C_1 вида

$$a^* = \ldots \propto |\,|\,|+|\,|\,|\,| \propto \ldots \qquad (4)$$

$k= 1$

$s = s_0$.

В результате выполнения первой инструкции состояние МТ и запись на ленте не изменились, только головка передвинулась вправо на одну ячейку и в следующий момент времени будет читать букву, расположенную в правой ячейке, т.е. букву $|$. Тоже самое на языке отображений можно сказать следующим образом:

-функция перехода МТ из одного состояния в другое реализует значение
$s_0 = f_S(s_0, \propto)$;

-функция выхода реализует значение $\propto = f_A (s_0, \propto)$;
-функция управления головкой реализует значение $r = f_V (s_0, \propto)$ или $k = 1$.

На втором шаге для пары $(s_0, |)$ на множестве Р допустимой является инструкция2. Ее применение приведет к появлению пары $(s_2, \propto)$, а головка при этом сместится вправо относительно первой буквы первого слова из a^*. Получится конфигурация C_2 вида

$$a^* = \ldots \propto \alpha||+|||| \propto \ldots \qquad (5)$$
$$k = 2$$
$$s = s_2.$$

В результате выполнения второй инструкции состояние МТ приняло значение $s = s_2$, головка стирает на ленте букву | и на ее место записывает несобственную букву α. Головка передвинулась вправо еще на одну ячейку и в следующий момент времени будет читать букву |. Тоже самое на языке отображений можно сказать следующим образом:
-функция перехода МТ из одного состояния в другое реализует значение
$s_2 = f_S (s_0, |)$;
-функция выхода реализует значение $\propto = f_A (s_0, |)$;
-функция управления головкой реализует значение $r = f_V (s_0, \propto)$ или $k = 2$.

Следующей выполняется инструкция, определяемая парой $(s_2, |)$. Для этой пары допустимой является инструкция 8. Как видно из ее описания, она не меняет состояние МТ и состояние ячейки, а лишь сдвигает головку еще на одну ячейку вправо. Формируется конфигурация C_3 вида:

$$a^* = \ldots \propto \alpha||+|||| \propto \ldots \qquad (6)$$
$$k = 3$$
$$s = s_2.$$

Очевидно, МТ будет оставаться в состоянии $s = s_2$ до тех пор, пока головкой будет читаться одна и та же буква, в данном случае это |. Находясь в конфигурации C_3 и выполняя повторно инструкцию 8 состояние МТ и ячейки не изменятся, но следующей читаемой буквой будет +. Для конфигурации C_4 допустимой является пара $(s_2, +)$, обрабатываемая инструкцией 9. По определению этой инструкции состояние МТ не меняется и лишь головка сдвигается вправо еще на одну ячейку. Для конфигураций $C_5, C_6, \ldots, C_8$ допустимой является инструкция 8. Ее последовательное выполнение приводит к смещению головки вправо до несобственного символа α, т.е. выполнение инструкции 8 к конфигурации C_8 завершается формированием конфигурации C_9 вида:

$$a^* = \ldots \propto \alpha||+|||| \propto \ldots \qquad (7)$$
$$k = 9$$
$$s = s_2.$$

Для конфигурации C_9 и пары (s_2, α) допустимой является инструкция 7, которая переводит МТ в состояние s_1, а букву α заменяет на букву |. Головка при этом остается на месте. Сформировалась новая конфигурация C_{10} вида:

$$a^* = \ldots \propto \alpha||+||||| \propto \ldots$$
$$k = 9$$
$$s = s_1.$$

Таким образом, в результате последовательного выполнения инструкций из множества P МТ перешла из конфигурации C_0 в конфигурацию C_{10}. При этом в последней конфигурации слева от буквы + осталось две палочки вместо первоначальных трех, а справа от буквы + стало на одну палочку больше. Неформально это можно интерпретировать так: одна буква из первого слагаемого перешла во второе слагаемое.

Из конфигурации C_{10} путем выполнения инструкций 5 и 6, применимых к парам $(s_1, |)$, $(s_1, +)$, мы придем к конфигурации C_{18} вида:

$a^* = \ldots \propto \alpha||+||||| \propto \ldots$
$k = 1$
$s = s_1$.

Затем применяя инструкцию 4 к паре (s_1, α) получим конфигурацию C_{19} вида:

$a^* = \ldots \propto \alpha||+||||| \propto \ldots$
$k = 2$
$s = s_0$.

Фактически мы перевели головку в начальное положение, соответствующее конфигурации C_0 и теперь повторив все сначала из конфигурации C_{19}, получим перенос еще одной буквы $|$ из первого слагаемого во второе. В результате сформируется конфигурация C_{27} вида:

$a^* = \ldots \propto \alpha\ \alpha\ |+|||||| \propto \ldots$
$k = 10$
$s = s_1$.

Из конфигурации C_{27} путем аналогичных действий движения головки влево и вправо мы придем к конфигурации C_{45} вида:

$a^* = \ldots \propto \alpha \propto \propto+||||||| \propto \ldots$
$k = 11$
$s = s_1$.

Проделав еще 8 шагов, получим конфигурацию C_{54} вида:

$a^* = \ldots \propto \alpha \propto \propto+||||||| \propto \ldots$
$k = 4$
$s = s_0$.

Из этой конфигурации с помощью инструкции 3 для пары $(s_0, +)$ перейдем к конфигурации C_{55} вида:

$a^* = \ldots \propto \alpha \propto \propto \propto ||||||| \propto \ldots$
$k = 4$
$s = s_3$.

К паре $(s_3, \propto)$ применима инструкция 10, из определения которой видно, что из этого состояния МТ никогда не выйдет и головка будет все время оставаться на месте. Назовем последнюю инструкцию инструкцией останова. Последняя конфигурация дает нам выходное слово, равное сумме двух целых чисел в алфавите $\underline{A}_n = \{|, +\} \cup \{\propto\}$.

Нетрудно заметить, что построенная нами МТ может сложить любые два целых числа. Отличие в выполнении операции сложения для разных слагаемых будет в числе конфигураций и времени работы МТ.

Задачи и упражнения.

1.Рассмотреть работу МТ, описанной выше, для сложения целых чисел 5 и 4.

2.Построить МТ для сложения целых чисел в алфавите $\underline{A}_n = \{0, 1, +\} \cup \{\propto\}$.

3.Построить МТ для сложения целых чисел в алфавите $\underline{A}_n = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +\} \cup \{\propto\}$.

4.Построить МТ для вычитания целых чисел в алфавите $\underline{A}_n = \{|, -\} \cup \{\propto\}$.

5.Построить МТ для вычитания целых чисел в алфавите $\underline{A}_n = \{0, 1, +\} \cup \{\propto\}$.

6.Построить МТ для умножения целых чисел в алфавите $\underline{A}_n = \{|, *\} \cup \{\propto\}$.

7.Построить МТ для умножения целых чисел в алфавите $\underline{A}_n = \{0, 1, +\} \cup \{\propto\}$.

Библиографический список.

1.Бауэр Ф.Л., Гооз Г. Информатика. Вводный курс. В 2- х ч. Ч.1 / Пер. с нем. М.: Мир.1990. 336с.

2.Брой М. Информатика. Основополагающее введение. В 4- х ч. Ч.1 / Пер. с нем. – М.: Диалог-МИФИ. 1996. 299с.

3.Информатика в терминах и определениях Российского законодательства / Под ред. проф. В.А. Никитова. М.: Славянский диалог, 2000. 431с.

4.Федеральный закон № 24 – ФЗ от 20.02.95. «Об информации, информатизации и защите информации».

5.Ершов Ю.Л., Палютин Е.А. Математическая логика / М. Наука. Главная редакция физ-мат. литературы. 1979. 320с.

6.Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов / М. Наука. Главная редакция физ-мат. литературы. 1975. 240с.

7.Шрейдер Ю.А. Равенство, сходство, порядок / М. Наука. Главная редакция физ-мат. литературы. 1971. 256с.

8.Принципы программирования на абсолютно универсальных вычислительных машинах // Методическая разработка, составители Гайсарян С.С., Луговая И.З. М. МАИ. 1980. 58с.

9.Кнут Д. Искусство программирования, т. 1 / МИР. М. 1976. 320с.

10.Мальцев А.И. Алгоритмы и рекурсивные функции / Наука. Главная редакция физ-мат. литературы. М. 1986. 367с.

11.Успенский В.А. Лекции о вычислимых функциях / Гос.изд-во физ.-мат. литературы. М.1960. 492с.

12.Петер Р. Рекурсивные функции / Изд-во иностранной литературы. М. 1951. 264с.