**Лекция 4. Тема занятия.** Универсальная тьюрингова программа. Нумерация множеств. Алгоритмические свойства множеств.

**Краткое содержание лекционного занятия.** Универсальная тьюрингова программа. Нумерация множеств. Алгоритмические свойства множеств.

**ЦЕЛЬ**: Ознакомить с понятиями: Универсальная тьюрингова программа. Нумерация множеств. Алгоритмическиае свойства множеств.

**ВОПРОСЫ**:

1. Универсальная тьюрингова программа.
2. Нумерация множеств.
3. Алгоритмическиае свойства множеств.

**ОСНОВНЫЕ ПОНЯТИЯ:** Универсальная тьюрингова программа. Нумерация множеств. Алгоритмическиае свойства множеств.

**Универсальная тьюрингова программа.** Существует универсальная тьюрингова программа *U*, которая может имитировать любую программу, работающую над фиксированным алфавитом *A* = {*a*1, *a*2 ,..., *an*}. Эта программа *U*, получив на входе псевдослово, содержащее в определенном виде код произвольной программы *T* и псевдослово *X*, должна оставить на ленте код программы *T* и псевдослово *T* (*X*) – результат работы программы *T* на псевдослове *X*.

Пусть Ψ*n* – множество всех функций из *N* в *N* таких, что каждая из них определена в точке 0 и вычислима программой, состоящей не более чем из n тьюринговых команд. Рассмотрим функцию *s* (*n*) = max *f* (0), где максимум берется по всем функциям из Ψ*n*. Очевидно, *s*(*n*) всюду определена и монотонна. Более того справедлива

***Теорема.*** *Для любой всюду определенной вычислимой функции f* : *N* → *N существует k* ∈ *N такое, что при любом m* ≥ *k выполняется неравенство f* (*m*) < *s* (*m*).

Действительно, пусть *f* (*n*) – произвольная всюду определенная функ-ция из *N* в *N*, тогда, очевидно, функция *F*(*n*) = max(*f* (3*n*), *f* (3*n* + 1), *f* (3*n* + 2)) + 1 будет также всюду определенной и вычислимой. Пусть в унарном коде ее вычисляет программа *TF*, состоящая из *k* команд. Рассмотрим программу *T* = [[1,*r*]*n*, *TF*], которая сначала записывает на ленте число *n*, а затем ра-ботает как *TF*. Очевидно, она состоит из 2*n* + *k* команд и вычисляет неко-торую функцию *F* ′ ∈ Ψ2*n+k*. По определению *F* ′ и *s* имеем *F*(*n*) = *F*′(0) ≤ *s* (2*n* + *k*). Используя мо-нотонность функции *s*, получим *F* (*n*) ≤ *s* (3*n*) при всех *n* ≥ *k*, следователь-но, *f* (3*n* + *i*) < *s* (3*n* + *i*), (*i* = 0, 1, 2). Отсюда следует, что при *m* ≥ 3*k* вы-полняется неравенство *f* (*m*) < *s* (*m*), что и требовалось доказать.

***Следствие.*** Функция *s* (*n*) невычислима, так как она растет быстрее, чем любая вычислимая функция. Заметим, что при любом фиксированном *n* значение *s* (*n*) можно по-пытаться вычислить путем перебора всех программ длины ≤ *n* и вычисле-ния для каждой из них времени работы до момента остановки или доказа-тельства ее незавершаемости. Но вопрос о завершаемости программ в общем виде алгоритмически неразрешим. Уточним основные моменты этого утверждения.

Пусть *T* – тьюрингова программа, работающая в алфавите *A*, и пусть kod (*T*) – слово в алфавите *A*, кодирующее программу *T* (на деталях кодирования не останавливаем

Программа *T* называется *самоприменимой*, если при подаче ей на вход ее собственного кода, она через конечное число шагов остановится, в противном случае программа называется несамоприменимой. Пусть *M* – множество кодов самоприменимых программ.

***Теорема.*** *Множество M алгоритмически неразрешимо*.

Доказательство. Пусть M – алгоритмически разрешимо. Тогда существуют две программы T1 и T2 такие, что T1 останавливается только на словах из M, а T2 – только на словах из A\* \M.

Тогда, если T2 на своем собственном коде остановится, то kod (T2) ∈ 129 ∈ M по определению M, но kod (T2) ∉ M по определению T2.

Если же T2 на своем собственном коде не остановится, то по определению M kod(T2) ∉ M, а по определению T2 kod(T2) ∈ M. Итак, в любом случае имеем противоречие.

 Еще одним примером алгоритмически неразрешимого множества является множество M1 кодов программ, которые останавливаются при пустом входе. Легко показать, что если бы M1 было разрешимым, то множество M тоже было бы разрешимым.

***Нумерация множеств***

Определение 1.  Множество X называют счетным, если можно установить взаимно однозначное отображение  между множеством неотрицательных целых чисел Z0 и множеством X.

Определение 2. Множество называют не более чем счетным, если оно счетно или конечно.

Определение 3. Перечислением или нумерацией множества X называется отображение  множества Z0 на множество X.

Перечисление f определяет на множестве X некоторую бесконечную последовательность  элементов из X такую, что каждый из элементов множества X встречается в этой последовательности, по крайней мере, один раз.

Если отображение f - взаимно однозначно, то f называют перечислением или нумерацией без повторений.

Определение 4. Множество X *называется эффективно счетным*, если существует функция , устанавливающая взаимно однозначное соответствие между множествами *Z0* и *X* такая, что *f* и *f--1* - вычислимые функции.

Теорема. Следующие множества являются эффективно счетными:

а) ; б) ; в)  - множество всех конечных последовательностей целых неотрицательных чисел.

Доказательство.

а) Докажем сначала эффективную счетность множества , состоящего из упорядоченных пар (*x, y*) с целочисленными неотрицательными компонентами *x*и*y.* Геометрически это множество представляет целочисленную решетку (рис.1).



Рис.1. Целочисленная решетка

Перенумеровать точки этой решетки можно различными способами, например, так, как показано на рисунке 2.



Рис. 2. Нумерация точек целочисленной решетки

Предложенная нумерация устанавливает взаимно однозначное отображение  между множествами  и . Алгоритмический характер процесса вычисления значений функции  очевиден. Следовательно, по тезису Черча  - вычислимая функция.

Для вычисления значений обратной функции  можно, например, в соответствии с предложенным алгоритмом последовательно нумеровать точки целочисленной решетки до номера *z*. Пара (*x, y*) координат точки с номером *z* является значением обратной функции . По тезису Черча  - вычислимая функция.

Таким образом, множество  эффективно счетно.

б) Теперь несложно доказать эффективную счетность множества . Для этого определим взаимно однозначное отображение  множества  упорядоченных троек неотрицательных целых чисел на множество  следующим образом . Из вычислимости функций  и  вытекает вычислимость функций  и . Поэтому множество  эффективно счетно.

в) Для доказательства эффективной счетности множества всех конечных последовательностей целых неотрицательных чисел  рассмотрим функцию

,

сопоставляющую каждому упорядоченному набору (*a1, a2, ..., ak*) из *k* неотрицательных целых чисел некоторое неотрицательное целое число.

Для доказательства взаимной однозначности функции  используем тот факт, что у каждого целого числа имеется ровно одно представление в двоичной системе счисления. Запишем значение функции  в двоичной системе счисления:

 (\*)

Например,

 =
=  = *616 - 1* = *615*;

 =
= *2320 -1* = *2319*;

 = *544 -1* = *543*;

 = *520 - 1* = *519*;

 = *3 - 1* = *2*;

 = *32 - 1* = *31*;

 = *1 - 1* = *0*.

Однозначность отображения  следует из того, что

.

Так как, кроме того, для любого неотрицательного числа *x* существует, очевидно, кортеж (*c1, c2, ..., cn*) такой, что , то функция  устанавливает взаимно однозначное соответствие между множеством  и множеством .

Покажем, как вычисляются значения обратной функции  для произвольного неотрицательного числа *x*.

В соответствии с формулой (\*)

,

где  - запись числа *x + 1* в двоичной системе счисления.

Например,

;

;

;

;

;

;

;

.

В силу тезиса Черча функции  и  вычислимы. Следовательно,  - эффективно вычислимое множество.

Определение 5. Число (*P*) называется *геделевым номером* *программы* *P* или просто *номером программы* *P.*

Отображение играет важную роль в теории алгоритмов. Название числа (*P*) связано с именем К. Геделя, впервые в 1931 году предложившего идею кодирования нечисловых объектов натуральными числами.

Ниже программу *P* с геделевым номером *n* будем обозначать . Из взаимной однозначности отображения следует  при , хотя обе эти программы  и  могут вычислять одну и ту же функцию.

Пример 1. Найдем геделев номер программы *P*:

,

,

вычисляющей функцию *f*(*x*)*= x + 2.*

.

(*S*(*1*)) = *4  (1 - 1*) + *1 = 1*;

.

Пример 2. Вычислим программу  по ее геделеву номеру *m*.

1) *m = 0.*

; .

Следовательно,

: *1*. *Z*(*1*).

2) *m = 1.*

; .

Следовательно,

: *1. S*(*1*).

3) *m = 2.*

; .

Следовательно,

: *1. Z*(*1*),

*2. Z*(*1*).

4) *m = 3.*

; .

Следовательно,

: *1. T*(*1, 1*).

Заметим, что различные программы  и  вычисляют одну и ту же функцию *f*(*x*)*= 0*.

**Алгоритмические свойства множеств.**

Основные понятия и обозначения, связанные с множествами и операциями над ними:

• Множества состоят из элементов. Запись x ∈ M означает, что x является элементом множества M.

• Говорят, что множество A является подмножеством множества B (запись: A ⊂ B), если все элементы A являются элементами B.

• Множества A и B равны (запись: A = B), если они содержат одни и те же элементы (другими словами, если A ⊂ B и B ⊂ A).

• Если A — подмножество B, не равное всему B, то A называют собственным подмножеством B (запись: A ( B).

• Пустое множество ∅ не содержит ни одного элемента и является подмножеством любого множества.

• Пересечение A∩B двух множеств A и B состоит из элементов, которые принадлежат обоим множествам A и B. Это записывают так: A ∩ B = {x | x ∈ A и x ∈ B} (читается: множество таких x, что . . .).

• Объединение A∪B состоит из элементов, которые принадлежат хотя бы одному из множеств A и B: A ∪ B = {x | x ∈ A или x ∈ B}.

• Разность A\B состоит из элементов, которые принадлежат A, но не принадлежат B: A \ B = {x | x ∈ A и x /∈ B}. Если множество B является подмножеством множества A, разность A \ B называют также дополнением B до A.

Законы алгебры множеств

1. Закон идемпотентности. А∪А= А∩А=А

2. Закон коммутативности А∪В= В∪А А∩В= В∩А

3. Закон ассоциативности А∪(В ∪С)= (А∪В) ∪С А∩(В ∩С)= (А∩В) ∩С

4. Закон дистрибутивности А∪В ∩ С= (А∪В) ∩ (А ∪ С) А ∩ (В ∪С)= А ∩ В ∪А ∩ С

5. Закон поглощения А∪ А∩ В = А А∩ (А∪ В) = А

6. Закон полупоглощения, или Блейка-Порецкого

А∪ A ∩ В =А∪ В А∩ ( A ∪ В)= А∩ В

7. Закон Де-Моргана

A∪B = A∩B A∩B = A∪B

8. Закон склеивания

 (А∪В) ∩( A ∪В)=В

А∩В ∪ A ∩В = В

9. Закон двойного дополнения A =А

10. Действия с константами

A ∩А=∅ A ∪А=U

 ∅∩А=∅ ∅∪А=A

A∩U=A A∪U=U

∅ =U U =∅

Свойства разности:

1. (А ∪В)\С= (А\С) ∪(В\С)

2. (А ∩В)\С= (А\С) ∩ (В\С)

3. А \(В∪ С)= (А\В) ∩ (А\С) =(А\В)\С

4. А \(В ∩ С)= (А\В) ∪ (А\С)

5. (А\В)\С ≠ А\ (В\С):

{\displaystyle A\cap (B\cup C)=(A\cap B)\cup (A\cap C)}

**ВЫВОДЫ:** Универсальной машиной Тьюринга называют [машину Тьюринга](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_%D0%A2%D1%8C%D1%8E%D1%80%D0%B8%D0%BD%D0%B3%D0%B0), которая может заменить собой любую машину Тьюринга.