Математическая логика и теория алгоритмов

Лектор: А. Л. Семенов

Лекция 2

Попытка расширить пределы вычислимого

Наряду с теми операциями над вычислимыми функциями, которые мы рассматривали, возможны более сложные конструкции. Например, можно, использовать графы в которых имеются вершины двух типов – функциональные, имеющие один вход и один выход и условные, имеющие один вход и два выхода. Далее можно в вершины таких графов помещать описания вычислимых функций и свойств (с учетом числа выходов). То, что получается называется блок-схемами. Понятно, как блок-схема работает.

Теорема о структурном программировании. Функции, вычисляемые блок-схемами, вычислимы.

Множество описаний алгоритмов. Универсальное действие. Универсальный алгоритм. Универсальная функция. Универсальное породимое множество

Мы уже обращались к описаниям вычислимых функций. В частности, мы рассматривали операции над вычислимыми функциями и операции над их описаниями. Полезно считать, что какой-то язык, в котором мы описываем алгоритмы, фиксирован. Таким образом, у нас фиксировано множество всех описаний алгоритмов. Мы можем использовать кодирование и считать, что это множество слов в алфавите 01. Это множество разрешимо – мы можем каждое слово в алфавите 01 попытаться рассмотреть как описание алгоритма и выяснить, действительно ли оно таковым является. Можно считать и что любое слово в этом алфавите – описание алгоритма, просто про некоторые слова ясно, что они не применимы ни к какому исходному данному.

Универсальный алгоритм:

УА (<A,x>) = A(x)

Этот алгоритм выясняет, является ли полученное слово кодом пары, если – да, то выделяет в ней объект (исходное данное), описания действий начала, продолжения, переработки, извлечения результата и дальше эти описания применяет.

Можно показать, что существует и универсальное перечислимое множество У – такое множество пар <x,y>, что для всякого перечислимого множества есть такое значение x, что это множество состоит в точности из таких y, что <x,y> лежит в У.

Диагонали

В квадратной таблице выписано (по горизонтали) конечное число конечных последовательностей.

Как построить последовательность, которой нет в этой таблице?

1	2	8	2	3
0	3	7	3	5
7	6	0	3	9
8	2	7	5	8
3	0	6	5	0

Можно взять диагональ таблицы и «испортить» её в каждом члене.

Например, прибавляем ко всем диагональным элементам 1 и получаем 24161.

Можно даже написать формулу для этого. Пронумеруем строки таблицы – номерами последовательностей и столбцы – номерами членов в них:

№чл.посл. / № посл.	**0**	**1**	**2**	**3**	**4**
0	1	2	8	2	3
1	0	3	7	3	5
2	7	6	0	3	9
3	8	2	7	5	8
4	3	0	6	5	0

Рассмотренная последовательность, которой точно нет в таблице, задается формулой: a(i) = t(i,i) +1, где t(i,j) – это элемент, стоящий в i–ой строке и j–ом столбце таблицы. Почему ее нет? Пусть она есть в строке с. Тогда t(c,c) = t(c,c) +1, чего не может быть.

Конечно, вы помните, как доказывается несчетность множества всех подмножеств натурального ряда, другими словами – несчетность множества характеристических функций таких подмножеств или последовательностей нулей или единиц.

Там тоже рассматривается гипотетическая таблица – теперь бесконечная, а в ней – диагональ.

Аргумент / № функц.	0	1	2	3	4	…
0	1	0	0	1	0	
1	1	1	0	1	0	
2	1	0	0	1	1	
3	0	0	0	1	0	
4	0	1	0	1	0	

………

Функция, которой нет в таблице – это 1 - t(i,i), другими словами – не t(i,i), то есть мы в диагонали нули заменили на единицы, а единицы – на нули.

Рассмотрим, наконец, таблицу, имеющую непосредственное отношение к вычислимости.

Заметим, что все слова можно линейно упорядочить. Например, так:

Λ, 0, 1, 00, 01, 10, 11, 000, 001,…

Существуют ли невычислимые функции?

Сколько есть вычислимых функций? Не больше, чем описаний, то есть - не больше, чем слов. Значит, их счетное количество. Но всех функций – несчетное число. Значит, существуют невычислимые функции.

В качестве имен строк и имен столбцов нашей таблицы возьмем все слова в этом порядке, в качестве содержимого клетки (i,j) возьмем значение функции с описанием i на аргументе j. Если значения нет, оставим клетку пустой:

Аргумент / Функция	0	1	01	…	1111
0				…	
1				…	
01	11	1	01	…	01
……	…	…	…	…	…
001…	0	11	111	…	(диаг) 10

Таким образом, наша таблица – это таблица универсальной функции, о которой мы недавно говорили.

«Диагональная» функция t(i,i) этой таблицы вычислима. Вычислима и функция d(i), которая равна нулю, если t(i,i)>0, и 1, если t(i,i)=0. Но, вроде бы диагональное рассуждение показывает, что функции d(i) нет в таблице! Ведь если ее описание есть i, то в клетке (i,i) должны стоять одновременно и ноль и не ноль. В чем тут дело? Противоречия здесь, конечно, нет – в этой клетке не стоит ничего, функция с описанием i в точке i не определена.

Что же мы установили? Наша функция d вычислима и принимает только значения 0 и 1. Значит, она задает перечислимое множество. Может ли это множество оказаться разрешимым? Докажем, что нет. Если оно разрешимо, то существует слово k, являющееся описанием вычислимой характеристической (всюду определенной) функции этого множества. Вот для такого предположения диагональное рассуждение действительно дает противоречие: функция с описанием k совпадает с функцией d там, где та определена. Значит в точке k ее значение должно, с одной стороны, быть t(k,k), с другой, – отличаться от него.

Итак, мы доказали теорему:

Теорема. Существует такое перечислимое множество A, заданное вычислимой функцией f, что:

- Оно не разрешимо
- Функцию f нельзя продолжить до вычислимой всюду определенной, в частности:
- Характеристическая функция множества A не вычислима.

Теперь мы вернемся к вопросу о том, каким можно взять запас действий, чтобы получить все мыслимые исчисления и алгоритмы.

Грамматики

Определение. Грамматика Γ – это цепочка <Σ,Ω,Π,S>

Σ – основной алфавит Γ

Ω – вспомогательный алфавит Γ

S – начальный символ Γ

Σ⋂Ω=∅, объединение Σ и Ω – это алфавит Γ, обозначим его Δ.

Π – это множество пар слов в алфавите Δ

Грамматика определяет следующее

правило создания Г*. В Г* входят:

1. S
2. Для каждой пары <u,v> из П все пары вида <tup,tvp>, где t,p – произвольные слова в алфавите Δ

Таким образом, всякий вывод в исчислении начинается с S. Один шаг вывода состоит в замене в слове некоторого вхождения u на v. Правило окончания для грамматики Г состоит из всех слов в алфавите Σ. Таким образом, породимые слова не могут содержать букв из вспомогательного алфавита. Заметим, что хотя правило создания – бесконечное множество, но его описание, как и описание всего исчисления, может считаться словом в конечном алфавите (как всегда – в алфавите 01).

Тезис Поста. Всякое породимое множество порождаемо некоторой грамматикой.

Утверждение Тезиса Поста не является математической теоремой. Оно представляет собой вывод из наблюдений за деятельностью человека, прежде всего в области доказательства математических теорем. За последовавшим в 1920-30 гг. осознанием справедливости этого тезиса, опыт построения различных породимых множеств не дал оснований сомневаться в тезисе Поста.

Определение Описание алгоритма Маркова – это цепочка $\Phi = \langle\Sigma,\Delta,\Pi\rangle$, где:

- Σ – *алфавит* Φ, у нас 01
- Π – цепочка слов вида *$u \rightarrow v$, или $u \rightarrow\bullet v$ - замен* Φ, *u, v* в алфавите Δ включающем Σ, $\rightarrow$, $\bullet \notin \Delta$, *u* называется левой частью замены, *v* - правой.
- Замены, содержащие $\rightarrow\bullet$, называются заключительными.

Какой алгоритм задает данное описание? Пусть задан объект – слово, с которым будет работать наш алгоритм.

Начало:

- не требуется делать ничего – тождественное преобразование объекта.

Продолжение:

- Л, если нет ни одной замены, левая часть которой входит в объект, или в объекте есть •, иначе И

Переработка:

- Найти первую замену, левая часть которой входит в объект, найти первое ее вхождение в объект и заменить его на правую часть этой замены.

Извлечение результата

- Стереть в слове все символы •, которые есть.

Пример: алфавит из трех символов 01 и дополнительный символ | - палочка.

Цепочка замен:

1. |0 → 0||
2. 1 → 0|
3. 0 →

Исходный объект 101

0|01

00||1

00||0|

00|0|||

000|||||

00|||||

0|||||

|||||

Как можно описать, что делает этот алгоритм с произвольным двоичным словом?

Тезис Черча. Всякая вычислимая функция вычислима некоторым алгоритмом Маркова.

В различных отраслях математики возникают т. н. алгоритмические проблемы, то есть проблемы построения алгоритма. Одним из важных видов таких проблем является проблема построения разрешающего алгоритма, то есть доказательство вычислимомсти характеристической функции некоторого множества.

Одной из первых явных формулировок алгоритмической проблемы является:

10-ая Проблема Гильберта

- Построить алгоритм, который по всякому алгебраическому уравнению от нескольких неизвестных с целыми коэффициентами выясняет, есть ли у него решение в целых числах.

Отрицательное решение этой проблемы получил Юрий Матиясевич (02.03.1947 -)