

**Введение в
математическую логику и
теорию алгоритмов**

Лекции 10 – 11

Алексей Львович Семенов

План

- Исчисления
- Породимые множества
- Грамматики
- Алгоритмы
- Вычислимые функции
- Алгоритмы Маркова

Общее понятие исчисления.

Предварительные определения

- Цепочка = конечная последовательность, которая может быть и пустой – Λ . Длина цепочки – число элементов в ней.
- Алфавит = цепочка различных символов.
- Слово (в данном алфавите) – цепочка символов.
- Ансамбль слов в данном алфавите – все слова.
Часто алфавит = $\{0, 1\}$.
- Ансамбль цепочек слов в данном алфавите – все цепочки слов.
- Ансамбль списков в данном алфавите – все цепочки, элементами которых являются слова или списки (индуктивное определение).
- **Задача.** Дать подробные индуктивные определения для понятий с этого экрана.

Действия и проверки. Описания

- Действие – исходное понятие. Действие:
 - слово, текст на понятном человеку языке; может выполняться и человеком, и каким-то устройством,
 - можно применить к любому исходному данному из фиксированного ансамбля, при этом ясно, что всегда получается результат применения – элемент (возможно, другого) фиксированного ансамбля.
- Действие – задает всюду определенную функцию.
- Проверка – действие с результатом 0 или 1.
- Проверка задает множество (где она дает 1), мы говорим, что она допускает его элементы (а другие – не допускает).

Исчисления. Создаваемые слова

- Исчисление в данном ансамбле – это пара из двух проверок:
- <проверка создания, проверка окончания>.
- Проверка создания применяется к цепочке объектов, проверка окончания – к объекту.
- Создаваемый исчислением объект определяется так:
если проверка создания допускает цепочку объектов a_0, \dots, a_n и все элементы этой цепочки, кроме последнего – создаваемы, то и последний элемент создаваем.
- Если проверка создания допускает цепочку из одного элемента, то его называют начальным объектом (в некоторых контекстах – аксиомой).
- **Задача.** Что, если таких у данного исчисления нет?

Исчисления. Породимые множества

- Объект порождаем данным исчислением, если он создаваем и его допускает проверка окончания.
- Множество, порождаемое исчислением, состоит из всех порождаемых им объектов.
- Породимое множество – множество, порождаемое каким-то исчислением.

Вывод

- Фиксируем исчисление.
- Если a_0, \dots, a_n – допускается проверкой создания, то говорим, что a_n создается из a_0, \dots, a_{n-1} (в данном исчислении).
- Вывод объекта a – цепочка объектов S , каждый из которых создается из какой-то цепочки объектов, встретившихся в S раньше него.
- **Задача.** Объект создаваем тогда и только тогда, когда у него имеется вывод.
- **Задача.** Пусть дано исчисление. Как организовать процесс выписывания всех выводов?
- **Задача.** Пусть дано исчисление. Как организовать процесс выписывания всех порождаемых (в нем) объектов (и только их)?

Примеры

Почему исчисление K модальной логики – это исчисление?

Проверка создания допускает:

- Цепочки из одной формулы (аксиомы):
 - тавтологии,
 - все формулы $(\Box (A \rightarrow B)) \rightarrow (\Box A \rightarrow \Box B)$.
- Цепочки из двух формул:
 - $\langle A, \Box A \rangle$,
 - $\langle A, \text{подстановка в } A \text{ формул вместо имен} \rangle$.
- Цепочки из трех формул:
 - $\langle A, A \rightarrow B, B \rangle$.

Проверка окончания:

все подходит.

Задача. Описать все проверки подробно.

Задача. Почему определение формулы – это исчисление?

Теоремы замкнутости для исчислений

Т. Объединение и пересечение породимых множеств породимы.

- Д. Объединение.
- А: <Проверка создания А, Проверка окончания А>.
- Б: < Проверка создания Б, Проверка окончания Б>.
- **Идея:**
- Создаем слова, следуя Проверке А и следуя Проверке Б,
- Берем то, что создано или по той, или по другой проверке.
- Проблема: как не перемешивать «по ходу дела» проверки А и Б?
- Выход: Метки для объектов, создаваемых по разным проверкам: Ах, Бу. Считаем, что символы А и Б в алфавит исчисления не входят.

Продолжение. Породимось объединения

- Припишем ко всем элементам цепочки, входящей в ту или иную Проверку создания, в начале символы А или Б.
- Объединим полученные проверки.
- Проверка создания допускает еще все пары $\langle Ax, x \rangle$, где x допускает проверка окончания А, и пары $\langle Bx, x \rangle$, где x допускает проверка окончания Б.
- **Задача.** Проверка окончания?
- **Задача.** Почему пересечение проверок – проверка?
- **Задача.** Доказать теорему для пересечения.
- **Задача.** Как обстоит дело с дополнением?

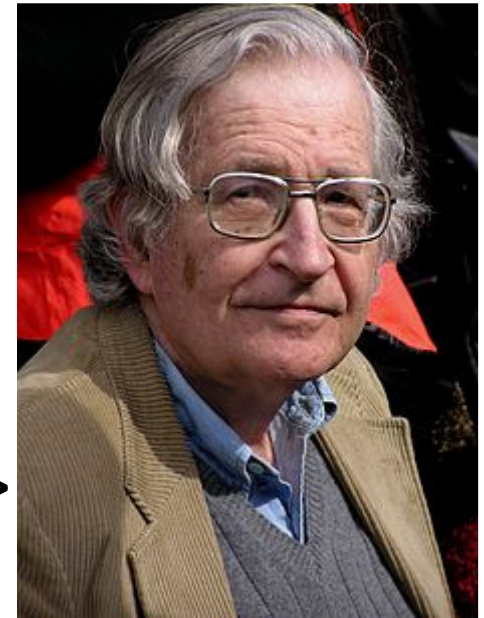
Грамматики

(Ноам Хомски, 07.12.1928 -)

Определение.

Грамматика Γ – это цепочка $\langle \Sigma, \Omega, P, S \rangle$

- Σ – основной алфавит Γ
- Ω – вспомогательный алфавит Γ
- S – начальный символ Γ
- $\Sigma \cap \Omega = \emptyset$, объединение Σ и Ω – это алфавит Γ , обозначим его Δ .
- P – это конечное множество пар слов в алфавите Δ . Эти пары называются заменами.



Грамматика

определяет исчисление Γ^*

Проверка создания Γ^* допускает:

- S
 - Всякий вывод в исчислении начинается с S .
- Для каждой замены $\langle u, v \rangle$ из Π все пары вида $\langle t_1 u r, t_1 v r \rangle$, где t, r – произвольные слова в алфавите Δ .
 - Один шаг вывода состоит в замене в слове некоторого вхождения u на v .
- Правило окончания для грамматики Γ состоит из всех слов в алфавите Σ .
 - Порождаемые слова не могут содержать букв из вспомогательного алфавита.

Проверка истинна на бесконечном множестве, описание грамматики – слово в конечном алфавите.

Примеры грамматик

- В них, следуя традиции, и для наглядности, используется символ стрелочки \rightarrow в заменах и выводах.

- **Задача.** Как породить все цепочки из 0 и 1?

- **Решение.** $\Gamma = \langle \Sigma, \Omega, \Pi, S \rangle$, основной алфавит $\Sigma = \{0, 1\}$, вспомогательный алфавит $\Omega = \{S\}$,

$$\Pi = \{S \rightarrow S0, S \rightarrow S1, S \rightarrow \Lambda\}.$$

Пример вывода: $S \rightarrow S0 \rightarrow S10 \rightarrow S010 \rightarrow S0010 \rightarrow 0010$.

- **Задача.** Как породить все десятичные числа? (Пример десятичного числа: – 3.141592.) Как породить все свободные переменные логики отношений?

- **Задача.** Что делает грамматика с основным алфавитом $\{a\}$, вспомогательным $\{S, B, M, E\}$ и правилами

$$\Pi = \{S \rightarrow BaE, B \rightarrow BM, Ma \rightarrow aM, ME \rightarrow E, B \rightarrow \Lambda, E \rightarrow \Lambda\} ?$$

- **Задача.** Как породить все слова, состоящие из одинакового количества букв a и b ?

- **Задача.** Построить множество, которое породить грамматикой нельзя.

Тезис Поста (вариант).

Всякое породимое множество
порождается некоторой
грамматикой.

Эмиль Пост (11.02.1897— 21.04.1954)



Алгоритмы

- Алгоритм – описание вида
- А
- Пока Б
– В
- Г
- где А, В, Г – действия, Б – проверка
- А - начало
- Б - продолжение
- В - переработка
- Г - извлечение результата.
- Интуитивные соображения...

Применение алгоритма

к исходному данному x – это (конечная или бесконечная) последовательность: $x_0, x_1, x_2, \dots, x_n, \dots$, получаемая так:

- $x_0 = x$,
- x_1 – результат применения действия A к x_0 ,
- Прежде чем получать следующий элемент последовательности, нужно к последнему, уже полученному, скажем, x_i , применить проверку B .
 - Если проверка допускает x_i , то к нему применить действие B , то, что получится, и будет следующим элементом последовательности x_{i+1} .
 - Если проверка не допускает x_i , то к нему применить действие Γ , результат этого применения будет последним в последовательности.
- Результат применения алгоритма к x – последний элемент последовательности.

Вычислимые функции

- Алгоритм задает (вычисляет) функцию, возможно, не всюду определенную.
- Функция, вычисляемая каким-то алгоритмом, называется вычислимой.
- Одно из основных понятий данного курса и математики
- Множество перечислимо, если оно пусто или является множеством значений всюду (на заданном ансамбле) определенной вычислимой функции.
- Множество разрешимо, если его характеристическая функция (в данном ансамбле) вычислима.

Универсальность

- Фиксирован язык описания алгоритмов.
- Можно считать, что алфавит = $\{0, 1\}$.
- Любое слово – описание, быть может, – нигде не определённой функции.
- Универсальный алгоритм: $U(\langle A, x \rangle) = A(x)$.
- Этот алгоритм выясняет, является ли исходное данное парой, если – да, то выделяет в первом элементе действия начала, продолжения, переработки, извлечения результата и затем применяет описанный алгоритм ко второму элементу.
- Можно показать, что существует универсальное перечислимое множество U – такое множество пар $\langle x, y \rangle$, что для всякого перечислимого множества A есть такое значение x , что $A = \{y \mid \langle x, y \rangle \in U\}$.

Диагонали (повторение)

В квадратной таблице выписано (по горизонтали) конечное число цепочек.

Как построить цепочку, которой нет в 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

Диагональ (конечная)

№чл.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

Взять диагональ и испортить её в каждом члене.

Например, прибавляем 1 и получаем **24161**.

$t(i,j)$ – это элемент, стоящий в i -ой строке и j -ом столбце.

Цепочка $a(i) = t(i,i) + 1$.

Пусть она в строке c . Тогда $t(c,c) = t(c,c) + 1$. Противоречие.

Диагональ несчетности (Кантора)

Аргумент	0	1	2	3	4	...
№ функц.						
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 $1 - t(i,i)$, не $t(i,i)$, нули заменили на единицы, единицы – на нули.

О словах

- Двоичные слова
- Линейный порядок:
Λ, 0, 1, 00, 01, 10, 11, 000, 001,...
- Кодирование слов в произвольном алфавите и элементов других ансамблей двоичными словами

Существуют ли невычислимые функции?

- Сколько есть вычислимых функций?
- Не больше, чем описаний.
- Не больше, чем слов.
- Счетное количество.
- Всех функций – несчетное.

Диагональ невычислимости

Имена строк и столбцов – слова.

В клетке (i,j) – значение функции с описанием i на аргументе j .

Если значения нет, оставим клетку пустой:

Аргумент	0	1	01	...	1111
Функция					
0				...	
1				...	
01	11	1	0111	...	01
.....
001...	0	11	111	...	(диаг) 10

Диагональ невычислимости

Аргумент	0	1	01	...	1111
Функция					
0				...	
1				...	
01	11	1	0111	...	01
.....
001...	0	11	111	...	(диаг) 10

- «Диагональная» функция $t(i,i)$ вычислима. Вычислима и «испорченная диагональ» $d(i)$, которая равна нулю, если $t(i,i) > 0$, и равна 1, если $t(i,i) = 0$. Кажется, что d не должно быть в таблице! Если она в строке i , то $d(i,i)$ – и ноль, и не ноль.
- В чем тут дело?
- В клетке (i) не стоит ничего!

Диагональ невычислимости

- Функция d вычислима и принимает только значения 0 и 1.
- Функцию d нельзя продолжить до вычислимой всюду определенной.
- Диагональное рассуждение дает противоречие.
- Пусть D – множество слов, на которых $d = 1$.
- Может ли множество D оказаться разрешимым?
- Если оно разрешимо, то существует слово k , являющееся описанием вычислимой характеристической (всюду определенной) функции этого множества.

Алгоритмы Маркова

- **Определение.** Описание алгоритма Маркова – это цепочка $\Phi = \langle \Sigma, \Delta, \Pi \rangle$, где
- Σ – основной алфавит Φ , у нас $\{0,1\}$, $\Sigma \subseteq \Delta$,
- Π – цепочка слов вида $u \rightarrow v$ или $u \rightarrow \bullet v$ – замен Φ ; u, v – в алфавите Δ ; $\rightarrow, \bullet \notin \Delta$,
- u называется левой частью замены, v – правой.
- Замены, содержащие $\rightarrow \bullet$, называются заключительными.

Алгоритмы Маркова

- Алгоритм?
 - Задан объект – слово
- Начало:
 - не требуется делать ничего – тождественное преобразование объекта.
- Продолжение:
 - 0, если нет ни одной замены, левая часть которой входит в объект, или в объекте есть ●; иначе 1.
- 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: 0, 1 и дополнительного символа | (палочки). Цепочка замен:

|0 → 0||

1 → 0|

0 →

Исходный объект: 101. Работа алгоритма:

0|01

00||1

00||0|

00|0|||

000||||

00||||

0||||

||||



Андрей Марков мл.
(9 [22].09.1903 – 11.10.1979)

Тезис Черча.

Всякая вычислимая функция
вычислима некоторым
алгоритмом Маркова.

Алонзо Черч (14.06.1903 — 11.08. 1995)



Алгоритмические проблемы

- Проблемы построения алгоритмов
- Проблема построения алгоритма разрешения – вычислимость характеристической функции

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: Юрий Матиясевич (02.03.1947 -)

