



# **Введение в математическую логику и теорию алгоритмов**

Лекция 10

**Алексей Львович Семенов**

# План

- Программа Гильберта
- Непротиворечивая и полная математика.
- Логика отношений
- Исчисление логики отношений
- Исчисления
- Породимые множества
- Грамматики. Тезис Поста

# Программа Гильберта

- Построение непротиворечивой и полной математики
  - Построение аксиоматической теории – исчисления («игры»)
  - Доказательство непротиворечивости и полноты «надежными», «финитными» средствами (анализом «игры»)

# Логика отношений

- **Синтаксис: Индуктивное построение формул**
- **Семантика: Интерпретации**

# Отношения, задаваемые формулами логики отношений (Семантика)

- Множество  $D$
- Отношения – подмножества  $D^N$
- Конечное число аргументов
- Имена отношений  $\sum, \exists n$
- Свободные переменные  $x_1, \dots$
- **Задача.** Формулы  $R(\mathbf{x})$  и  $Q(\mathbf{x})$  эквивалентны в структуре  $M$  т. е.  $M \models R(\mathbf{x}) \equiv Q(\mathbf{x})$  титтк значения  $R(\mathbf{x})$  и  $Q(\mathbf{x})$  (отношения в  $D^N$ ) совпадают.
- О. Формулы эквивалентны, если они эквивалентны в любой структуре (данной сигнатуры).

# Предваренная нормальная форма

- Формула находится в предваренной нормальной форме (п.н.ф.), если она не содержит кванторов, или имеет вид:
- $Q_1 x_1 \dots Q_n x_n \Phi$ , где все  $Q_i$  – это  $\forall$  или  $\exists$ , а в  $\Phi$  кванторов нет.
- **Задача.** Дать индуктивное определение формулы (находящейся) в п.н.ф.
- **Задача.** Построить алгоритм, который по всякой формуле логики отношений строит формулу (находящуюся) в п.н.ф., ей эквивалентную (ее предваренную нормальную форму).
- Можно переименовывать связанные переменные...

# Предваренная нормальная форма

$$\models (\forall u \Phi[u/x]) \equiv (\forall v \Phi[v/x])$$

$$\models (\exists u \Phi[u/x]) \equiv (\exists v \Phi[v/x])$$

$$\models (\mathbf{Q}u \Phi[u/x]) \tau \Psi \equiv (\mathbf{Q}u (\Phi[u/x] \tau \Psi)), \tau \in \{\wedge, \vee\}, \mathbf{Q} \in \{\forall, \exists\}, \Psi \text{ не содержит } u$$

$$\models (\neg (\forall u \Phi[u/x])) \equiv (\exists v \neg \Phi[v/x])$$

$$\models (\neg (\exists u \Phi[u/x])) \equiv (\forall v \neg \Phi[v/x])$$

# Теоремы о логике отношений

## Теорема перечислимости.

Множество общезначимых формул перечислимо: есть процесс деятельности, позволяющий для всякой общезначимой формулы когда-нибудь узнать, что она общезначима.

## Теорема компактности для счетного множества утверждений.

Если любое конечное подмножество теории имеет модель, то и вся теория имеет модель.

**Задача.** Как обстоит дело с общезначимыми и с необщезначимыми формулами в логике высказываний?

# Исчисление для логики отношений (дедуктика)

- Будет указано **индуктивное определение** выводимой формулы, формализующее практику математических доказательств.

# Частные случаи тавтологий логики высказываний в логике отношений

- Возьмем тавтологию логики высказываний, например:  
 $A_1 \rightarrow (A_2 \rightarrow A_1). \quad (*)$
- Подставим в (\*) вместо имен высказываний  $A_1$  и  $A_2$  формулы (замкнутые или незамкнутые) логики отношений.
- Например, вместо  $A_1$  подставим  $\forall u_1(P_5(u_1))$ , а вместо  $A_2$  подставим  $P_4(x_1, x_1)$ :  
 $\forall u_1(P_5(u_1)) \rightarrow (P_4(x_1, x_1) \rightarrow \forall u_1(P_5(u_1)))$ .
- То, что получилось, называется *частным случаем тавтологии* (\*) логики высказываний в логике отношений.
- Любая такая формула истинна в любой структуре при любой интерпретации.
- Вместо «частный случай тавтологии...» говорим просто «тавтология».

# Исчисление логики отношений

- Фиксируем сигнатуру  $\Sigma = \langle Pr \rangle$ .
- Исчисление (одно для данной сигнатуры) задаётся *аксиомами* (являющимися формулами сигнатуры  $\Sigma$ ) и *правилами вывода*.
- **Аксиомы:**
  - A1. частные случаи тавтологий логики высказываний,
  - A2. формулы вида  $\forall u \Phi[u/x] \rightarrow \Phi[t/x]$ ,
  - A3. формулы вида  $\Phi[t/x] \rightarrow \exists u \Phi[u/x]$ ,где  $\Phi$  – формула,  $x$  – свободная переменная ( $x \in FVar$ ),  $u$  – связанная переменная ( $u \in BVar$ ), не входящая в  $\Phi$ ,  $t$  – свободная переменная.

# Исчисление логики отношений

Правила вывода:

$$R1 \quad \frac{\Phi, \quad \Phi \rightarrow \Psi}{\Psi} \quad (\text{modus ponens, (MP)})$$

$$R2 \quad \frac{\Phi \rightarrow \Psi}{\Phi \rightarrow \forall u \Psi}$$

$$R3 \quad \frac{\begin{array}{c} [u/x] \\ \Psi \rightarrow \Phi \end{array}}{\exists u \Psi[u/x] \rightarrow \Phi}$$

В R2, R3  $x$  не входит в  $\Phi$ .

Правила R2 и R3 называются правилами Бернайса.

Индуктивное  
определение  
выводимой формулы:

Аксиомы выводимы.  
Если уже выведены формулы, написанные в верхней части правила, то написанная внизу формула - выводима. Говорят, что она выводима из них.

# Примеры выводов

Вывод – цепочка формул, где каждая формула – аксиома или выводима из предшествующих в цепочке.

Пример 1. (1)  $\vdash \forall u P(x) [u/x] \rightarrow P(x) [x/x]$  (аксиома A2)

$\vdash \forall u P(u) \rightarrow P(x)$  (аксиома A2)

(2)  $\vdash \forall u P(u) \rightarrow \forall v P(v)$  (по правилу R2 из (1))

(В этом выводе  $P$  – имя одноместного отношения.)

Пример 2. Пусть  $\Phi$  - любая формула в нашей сигнатуре.

•  $\vdash (\forall u \Phi[u/x] \rightarrow \Phi) \rightarrow ((\Phi \rightarrow \exists u \Phi[u/x]) \rightarrow (\forall u \Phi[u/x] \rightarrow \exists u \Phi[u/x]))$   
(Частный случай тавтологии  $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$ .)

(2)  $\vdash \forall u \Phi[u/x] \rightarrow \Phi$  (A2,  $\Phi$  – это  $\Phi[x/x]$ )

(3)  $\vdash (\Phi \rightarrow \exists u \Phi[u/x]) \rightarrow (\forall u \Phi[u/x] \rightarrow \exists u \Phi[u/x])$  (по MP из (2) и (1))

(4)  $\vdash \Phi \rightarrow \exists u \Phi[u/x]$  (A3)

(5)  $\vdash \forall u \Phi[u/x] \rightarrow \exists u \Phi[u/x]$  (по MP из (4) и (3))

# Пример вывода

Пример 3. (Используем обычное обозначение для двуместного отношения «меньше или равно».)

- (1)  $\vdash \forall u (u \leq y) \rightarrow x \leq y$  (A2, терм  $t = x$ )
- (2)  $\vdash x \leq y \rightarrow \exists v (x \leq v)$  (A3, терм  $t = y$ )
- (3)  $\vdash (\forall u (u \leq y) \rightarrow x \leq y) \rightarrow$   
 $\rightarrow ((x \leq y \rightarrow \exists v (x \leq v)) \rightarrow (\forall u (u \leq y) \rightarrow \exists v (x \leq v)))$   
(частный случай тавтологии  $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$ )
- (4)  $\vdash (x \leq y \rightarrow \exists v (x \leq v)) \rightarrow (\forall u (u \leq y) \rightarrow \exists v (x \leq v))$  (по MP из (1) и (3))
- (5)  $\vdash \forall u (u \leq y) \rightarrow \exists v (x \leq v)$  (по MP из (2) и (4))
- (6)  $\vdash \forall u (u \leq y) \rightarrow \forall u \exists v (u \leq v)$  (по R2 из (5))
- (7)  $\vdash \exists v \forall u (u \leq v) \rightarrow \forall u \exists v (u \leq v)$  (по R3 из (6))

Заметим, что полученная формула – общезначима.

# ИСТИННОСТЬ ВЫВОДИМОГО

**Теорема об истинности выводимого.** Всякая выводимая формула является общезначимой.

**Структура доказательства (индукция по построению).**

- A1 Частные случаи тавтологий логики высказываний – общезначимы.
- A2 Формулы вида  $\forall u \Phi[u/x] \rightarrow \Phi[t/x]$  – общезначимы.
- A3 Формулы вида  $\Phi[t/x] \rightarrow \exists u \Phi[u/x]$  – общезначимы.
- R1 Если формулы  $\Phi$  и  $\Phi \rightarrow \Psi$  общезначимы, то формула  $\Psi$  – общезначима.
- R2 Если формула  $\Phi \rightarrow \Psi$  общезначима и  $\Phi$  не содержит  $x$ , то формула  $\Phi \rightarrow \forall u \Psi[u/x]$  – общезначима.
- R3 Если формула  $\Psi \rightarrow \Phi$  общезначима и  $\Phi$  не содержит  $x$ , то формула  $\exists u \Psi[u/x] \rightarrow \Phi$  – общезначима.

Доказательство рассматривает определение истинности, значения на последовательности, и т. д.

# Выводимость истинного

- **Теорема Гёделя о полноте.**  
Общезначимость в логике отношений совпадает с выводимостью в исчислении логики отношений.
- Теорема в курсе не доказывалась
- **Задача.** Доказать теорему.
- Решение – не обязательно.

# Общее понятие исчисления.

## Предварительные определения

- Цепочка = конечная последовательность, которая может быть и пустой –  $\Lambda$ . Длина цепочки – число элементов в ней.
- Алфавит = конечная цепочка символов
- Слово (в данном алфавите) – цепочка символов этого алфавита.
- Ансамбль слов в данном алфавите – все слова. Часто:  
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, являющееся текстом (цепочкой) на понятном человеку языке;
  - Его можно применить к любому исходному данному из фиксированного ансамбля, при этом ясно, что всегда получается результат применения – элемент (возможно, другого) фиксированного ансамбля
  - Оно может применяться, выполняться и человеком, и каким-то устройством,
- **Действие – задает всюду определенную функцию**
- Проверка – действие с результатом 0 или 1
- Проверка задает характеристическую функцию множества (где она дает 1), мы говорим, что она допускает его элементы (а другие – не допускает)

# Исчисления. Создаваемые объекты

- Исчисление в данном ансамбле – это пара из двух проверок:
- <проверка возможности, проверка завершения>.
- проверка возможности (создания) применяется к цепочке объектов, проверка завершения (окончания) – к объекту.
- создаваемый исчислением объект индуктивно определяется так:

Если проверка возможности допускает цепочку объектов  $a_1, \dots, a_n$  и все элементы этой цепочки, кроме последнего – создаваемы,

то и последний элемент создаваем.

- Если проверка возможности допускает цепочку из одного элемента, то его называют начальным объектом (в некоторых контекстах – аксиомой).
- **Задача.** Что, если таких у данного исчисления нет?
- Общее представление о деятельности, использующей уже

# Исчисления. Породимые множества

- Объект порождаем данным исчислением, если он создаваем и его допускает проверка завершения.
- Исчисление порождает множество из всех порождаемых им объектов – множество, порождаемое этим исчислением.
- Породимое множество – множество, порождаемое каким-то исчислением.

Эмиль Пост

(11.02.1897 — 21.04.1954)



# Вывод

- Фиксируем исчисление.
- Если  $a_1, \dots, a_n$  – допускается проверкой возможности, то говорим, что  $a_n$  создается из  $a_1, \dots, a_{n-1}$  (в данном исчислении).
- Вывод объекта  $a$  – цепочка объектов  $S$ , каждый из которых создается из какой-то цепочки объектов, встретившихся в  $S$  раньше него.
- Шаг вывода – добавление одного элемента в цепочку
- **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 возможности 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 завершения**

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 возможности А, Проверка завершения А>,  
А>,
- Б: < Проверка возможности Б, Проверка завершения 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

# Продолжение. Породимось объединение

- Припишем ко всем элементам цепочки, входящей в ту или иную Проверку возможности, в начале символы А или Б.
- Объединим полученные проверки.
- Проверка возможности допускает еще все пары  $\langle Ax, x \rangle$ , где проверка завершения А допускает  $x$ , и пары  $\langle Bx, x \rangle$ , где проверка завершения Б допускает  $x$ .
- **Задача.** Проверка завершения ?

**Задача.** Почему пересечение (конъюнкция) проверок – проверка?

**Задача.** Доказать теорему для пересечения.

**Задача.** Как обстоит дело с дополнением?

# Грамматика

(Ноам Хомски, 07.12.1928 - )



## Определение.

Грамматика  $\Gamma$  – это цепочка  $\langle \Sigma, \Omega, \Pi, s \rangle$

$\Sigma$  – основной алфавит  $\Gamma$

$\Omega$  – вспомогательный алфавит  $\Gamma$

$s$  – начальный символ  $\Gamma$ ,  $s \in \Omega$

$\Sigma \cap \Omega = \emptyset$ , объединение  $\Sigma$  и  $\Omega$  – это алфавит  $\Gamma$ , обозначим его  $\Delta$ .

$\Pi$  – это конечное множество пар слов в алфавите  $\Delta$ . Эти пары называются заменами.

# Грамматика

определяет исчисление  $\Gamma^*$

Проверка возможности  $\Gamma^*$  допускает:

- $S$ 
  - Всякий вывод в исчислении начинается с  $S$ .
- Для каждой замены  $\langle u, v \rangle$  из  $\Pi$  все пары вида  $\langle t_1 u t_2, t_1 v t_2 \rangle$ , где  $t_1, t_2$  – произвольные слова в алфавите  $\Delta$ 
  - Один шаг вывода состоит в замене в слове некоторого входящего в него  $u$  на  $v$ .

Проверка завершения для грамматики  $\Gamma$  допускает все слова в алфавите  $\Sigma$ .

- Порождаемые слова не могут содержать букв из вспомогательного алфавита.

Описание грамматики – слово в конечном алфавите.

# Примеры грамматик

- В них, следуя традиции, и для наглядности, используется символ стрелочки 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 заменами  
 $\Pi = \{S \rightarrow BaE, B \rightarrow BM, Ma \rightarrow aaM, ME \rightarrow E, B \rightarrow \Lambda, E \rightarrow \Lambda\}$  ?
- **Задача.** Построить грамматику порождающую 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.**

**Соответствие между  
интеллектуальной  
реальностью и теоретико-  
множественной математикой**