

# Математическая логика и логическое программирование

[mk.cs.msu.ru](http://mk.cs.msu.ru) → Лекционные курсы

→ Математическая логика и логическое программирование (3-й поток)

## Блок 38

Хорновские логические программы:  
деревья SLD-резолютивных вычислений,  
стратегии вычисления и их полнота,  
стандартная стратегия вычисления

Лектор:  
**Подымов Владислав Васильевич**  
E-mail:  
[valdus@yandex.ru](mailto:valdus@yandex.ru)

# Вступление

$$\mathcal{R}_1 : p(X, Y) \leftarrow q(X), r(Y);$$

$$\mathcal{R}_2 : p(X, X) \leftarrow r(X);$$

$$\mathcal{R}_3 : q(\mathbf{b}); \quad \mathcal{R}_4 : r(\mathbf{c}); \quad \mathcal{R}_5 : s(\mathbf{b});$$

Запросом  $?p(X, Y), s(X)$  к этой программе порождаются, например, такие SLD-резолютивные вычисления, построенные согласно **стандартному правилу выбора подцели**:

$$?p(X, Y), s(X)$$

$$\mathcal{R}'_1 \downarrow \theta_1 = \{X'/X, Y'/Y\}$$

$$?q(X), r(Y), s(X)$$

$$\mathcal{R}'_3 \downarrow \theta_2 = \{X/\mathbf{b}\}$$

$$?r(Y), s(\mathbf{b})$$

$$\mathcal{R}'_4 \downarrow \theta_3 = \{Y/\mathbf{c}\}$$

$$?s(\mathbf{b})$$

$$\mathcal{R}'_5 \downarrow \theta_4 = \varepsilon$$

□

$$?p(X, Y), s(X)$$

$$\mathcal{R}'_2 \downarrow \eta_1 = \{X'/X, Y/X\}$$

$$?r(X), s(X)$$

$$\mathcal{R}'_4 \downarrow \eta_2 = \{X/\mathbf{c}\}$$

$$?s(\mathbf{c})$$

тупик

# Вступление

Способ выбора правил программы при построении SLD-резолютивных вычислений существенно влияет на то, какой именно результат будет получен (и будет ли получен хоть какой-нибудь результат)

Этот факт нетруден для осознания:

- ▶ Правило — это описание одного из способов решения рассматриваемой задачи
- ▶ Бывают как успешные, так и неуспешные способы решения задач
  - ▶ Например, если заданный элемент списка располагается только в голове, то искать его в хвосте — заведомый неуспех
- ▶ Задачу можно успешно решать по-разному, получая разные ответы
  - ▶ Например, если в ответе требуется произвольный элемент списка, то можно выдать голову списка, или голову его хвоста, или голову хвоста его хвоста, ...

Чтобы умело рассуждать о способах выбора правил для получения ответов, объединим всевозможные SLD-резолютивные вычисления программ в единую удобную структуру

# Деревья SLD-резолютивных вычислений ХЛП

Дерево SLD-резолютивных вычислений для запроса  $Q$  к программе  $\mathcal{P}$  и правила выбора подцели  $\mathfrak{R}$ , — это размеченное корневое ориентированное (возможно, бесконечное) дерево  $T_{\mathcal{P}, Q}^{\mathfrak{R}}$ , устроенное так:

1. Каждая вершина помечена запросом
2. Корень помечен запросом  $Q$
3. Дуги имеют вид  $[\mathcal{Q}_1] \xrightarrow{\mathcal{R}, \theta} [\mathcal{Q}_2]$ :
  - ▶  $\mathcal{Q}_2$  — SLD-резольвента  $\mathcal{Q}_1$  и утверждения из  $\mathcal{P}$  с вариантом  $\mathcal{R}$  и унифициатором  $\theta$
  - ▶ Резольвента строится для подцели  $\mathfrak{R}(\mathfrak{S})$ , где  $\mathfrak{S}$  — вычисление от корня до изображённой вершины  $[\mathcal{Q}_1]$
  - ▶ Дуги из  $[\mathcal{Q}_1]$  отвечают выбору всевозможных правил из  $\mathcal{P}$  для построения резольвент с  $\mathcal{Q}_1$  и упорядочены по порядку записи правил в  $\mathcal{P}$
4. Листьями являются запросы, у которых нет ни одной SLD-резольвенты правилами из  $\mathcal{P}$

$T_{\mathcal{P}, Q}$  — дерево  $T_{\mathcal{P}, Q}^{\mathfrak{R}}$  для **стандартного** правила  $\mathfrak{R}$

Порядок дуг дерева в иллюстрациях — слева направо

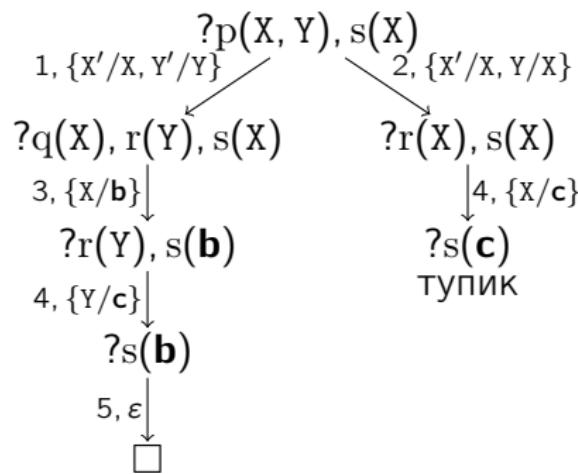
# Деревья SLD-резолютивных вычислений ХЛП

## Примеры

Программа  $\mathcal{P}$ :

- 1 :  $p(X, Y) \leftarrow q(X), r(Y);$
- 2 :  $p(X, X) \leftarrow r(X);$
- 3 :  $q(\mathbf{b}); \quad 4 : r(\mathbf{c}); \quad 5 : s(\mathbf{b});$

Дерево  $T_{\mathcal{P}, ?p(X, Y), s(X)}$ :



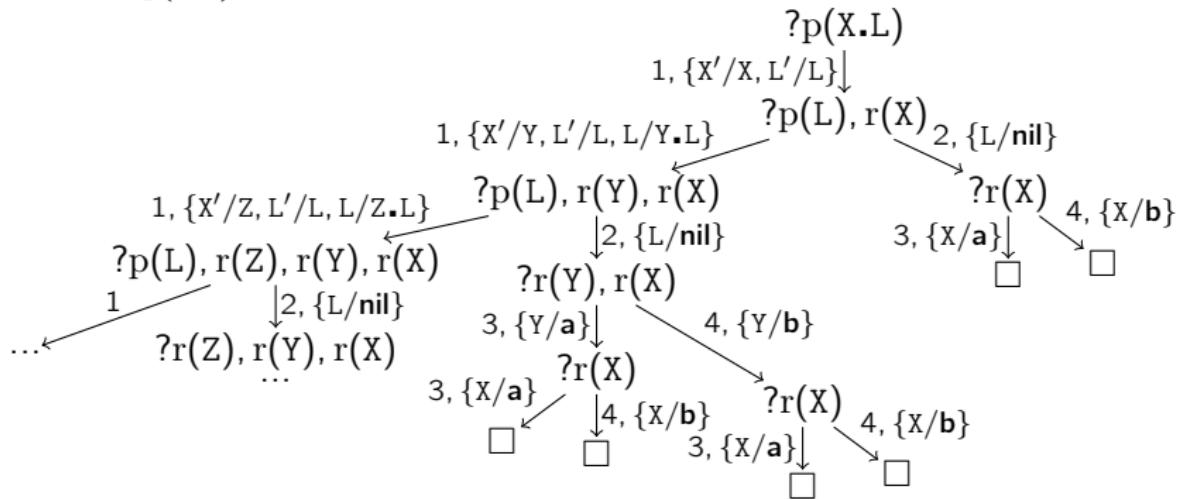
# Деревья SLD-резолютивных вычислений ХЛП

## Примеры

Программа  $\mathcal{P}$ :

$$\begin{aligned} 1 : \quad & p(X.L) \leftarrow p(L), r(X); \\ 2 : \quad & p(\text{nil}); \quad 3 : r(\mathbf{a}); \quad 4 : r(\mathbf{b}); \end{aligned}$$

Дерево  $T_{\mathcal{P}, ?p(X.L)}$ :



# Стратегии вычисления ХЛП

Деревья вычислений бывают разные:

- ▶ Конечные и бесконечные
- ▶ С конечным и с бесконечным числом ветвей
- ▶ Всюду успешные и с тупиками
- ▶ Содержащие один ответ, или много ответов, или ни одного ответа

Каждая ветвь в дереве  $T_{Q,\mathcal{P}}^{\mathfrak{R}}$  отвечает SLD-резолютивному вычислению программы  $\mathcal{P}$  для запроса  $Q$  согласно правилу  $\mathfrak{R}$ , и в дереве перечислены все такие вычисления

Значит, перебор всех таких вычислений можно устроить как **обход** дерева  $T_{Q,\mathcal{P}}^{\mathfrak{R}}$

# Стратегии вычисления ХЛП

Стратегия вычисления ХЛП состоит из

- ▶ правила выбора подцели и
- ▶ способа обхода дерева вычислений

Стратегия вычисления с правилом выбора подцели  $\mathfrak{X}$  называется **полной**, для любого запроса  $Q$  к любой программе  $\mathcal{P}$  она позволяет построить (перечислить) все успешные  $\mathfrak{X}$ -вычисления  $\mathcal{P}$  для запроса  $Q$

Деревом SLD-резолютивных вычислений для запроса  $Q$  к программе  $\mathcal{P}$ , **построенным согласно заданной стратегии вычисления**, будем называть фрагмент дерева SLD-резолютивных вычислений для  $Q$ ,  $\mathcal{P}$  и правила выбора подцели из стратегии, состоящий из всех вершин, посещаемых при обходе, и всех соединяющих их дуг

# Стратегии вычисления ХЛП

Два основных вида обходов деревьев вычислений ХЛП:

- ▶ **Обход в ширину**: вершины дерева обходятся поярусно по неубыванию удалённости от корня
  - ▶ Это можно представить как перебор всех вычислений по возрастанию длины
- ▶ **Обход в глубину**: при обходе из вершины сначала последовательно по порядку обходятся вершины по исходящим дугам, и по окончании обхода выполняется возврат в предыдущую вершину
  - ▶ Это можно представить себе как последовательные попытки достроить (одно) рассматриваемое вычисление до более длинного пошагово всеми возможными способами по порядку правил

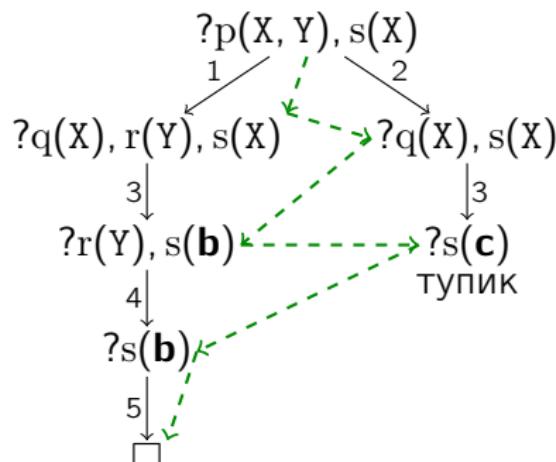
# Стратегии вычисления ХЛП

**Пример** дерева вычислений, построенного согласно стандартному правилу выбора подцели и обходу в ширину

Программа  $\mathcal{P}$ :

- 1 :  $p(X, Y) \leftarrow q(X), r(Y);$
- 2 :  $p(X, X) \leftarrow r(X);$
- 3 :  $q(\mathbf{b}); \quad 4 : r(\mathbf{c}); \quad 5 : s(\mathbf{b});$

Дерево вычислений (зелёным пунктиром обозначен порядок обхода вершин дерева):

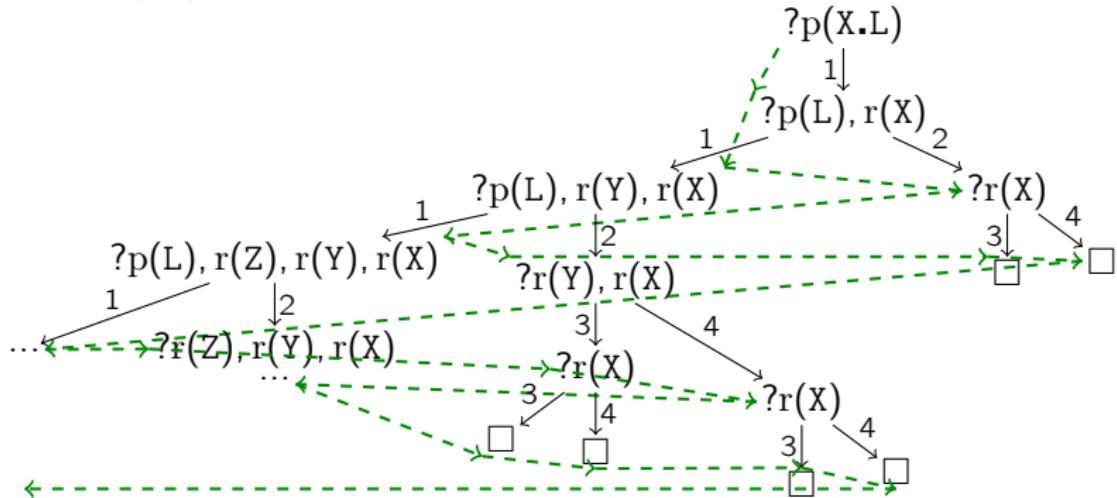


## Стратегии вычисления ХЛП

**Пример** дерева вычислений, построенного согласно стандартному правилу выбора подцели и обходу в ширину

- 1 :  $p(X.L) \leftarrow p(L), r(X);$
- 2 :  $p(\text{nil});$     3 :  $r(\mathbf{a});$     4 :  $r(\mathbf{b});$

Дерево  $T_{\mathcal{P},?p(x.L)}$ :



# Стратегии вычисления ХЛП

Стратегия обхода в ширину полна:

- ▶ Из каждой вершины исходит конечное число дуг (не больше чем правил в программе), а значит, каждый ярус дерева конечен
- ▶ Каждая вершина каждого яруса рано или поздно будет посещена
- ▶ Каждое успешное вычисление конечно, а значит, завершается на некотором ярусе
- ▶ Значит, каждое успешное вычисление рано или поздно будет построено

Но у этой стратегии есть и серьёзный недостаток, присущий обходам больших графов в ширину:

- ▶ При обходе нужно хранить в памяти **все** вершины очередного яруса
- ▶ Число таких вершин может расти экспоненциально относительно номера яруса
  - ▶ можете посчитать, на каком ярусе двоичного дерева или, например, дерева со степенью ветвления 4 содержится гугол вершин

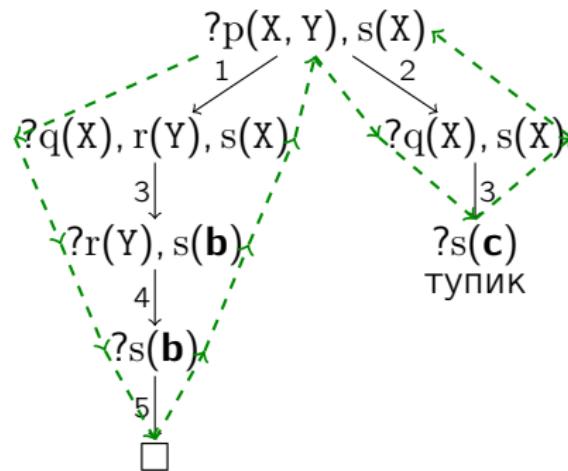
# Стратегии вычисления ХЛП

**Пример** дерева вычислений, построенного согласно стандартному правилу выбора подцели и обходу в глубину

Программа  $\mathcal{P}$ :

- 1 :  $p(X, Y) \leftarrow q(X), r(Y);$
- 2 :  $p(X, X) \leftarrow r(X);$
- 3 :  $q(\mathbf{b}); \quad 4 : r(\mathbf{c}); \quad 5 : s(\mathbf{b});$

Дерево вычислений:

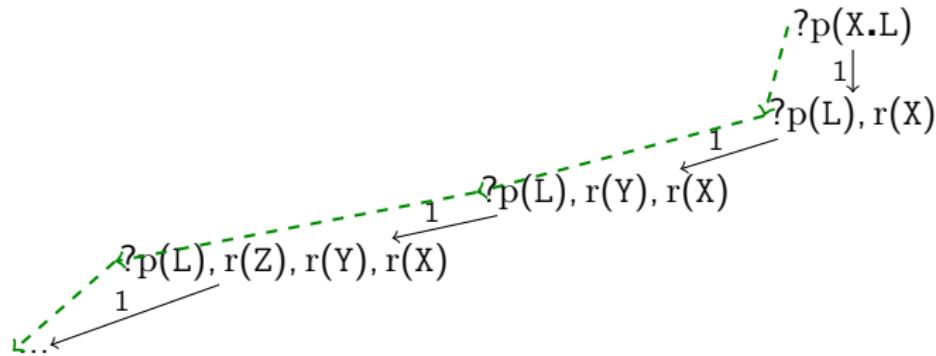


# Стратегии вычисления ХЛП

**Пример** дерева вычислений, построенного согласно стандартному правилу выбора подцели и обходу в глубину

- 1 :  $p(X.L) \leftarrow p(L), r(X);$
- 2 :  $p(\text{nil});$     3 :  $r(\mathbf{a});$     4 :  $r(\mathbf{b});$

Дерево вычислений:



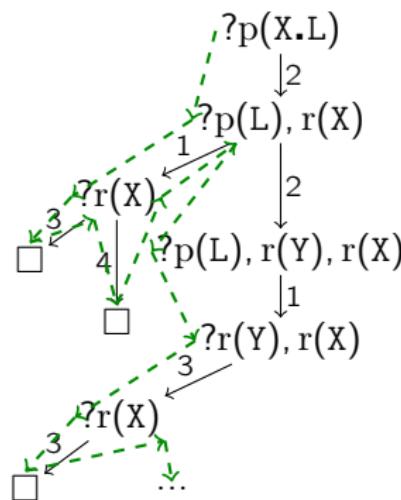
До бесконечности обходится самая левая ветвь дерева, ни одно успешное вычисление не найдено

# Стратегии вычисления ХЛП

**Пример** дерева вычислений, построенного согласно стандартному правилу выбора подцели и обходу в глубину

- 1 :  $p(\text{nil})$ ;
- 2 :  $p(X.L) \leftarrow p(L), r(X)$ ;
- 3 :  $r(a)$ ;      4 :  $r(b)$ ;

Дерево вычислений:



## Стандартная стратегия вычисления ХЛП

По этим примерам видно, что стратегия, основанная на обходе в глубину

- ▶ неполна и
- ▶ чувствительна к порядку правил: даже если успешных вычислений сколь угодно много, в зависимости от порядка правил могут быть
  - ▶ найдены они все, или
  - ▶ найдены некоторые успешные вычисления, но не все, или
  - ▶ не найдены никакие успешные вычисления

Но использование этой стратегии позволяет избежать проблем с использованием памяти, возникающих при обходе в ширину, и она удобна и достаточно эффективна для использования на практике

Поэтому в **стандартной стратегии вычисления ХЛП**, используемой обычно в интерпретаторах, используются **стандартное правило выбора подцели** и **обход в глубину**

Выбор надлежащего порядка правил и атомов в телах правил при этом становится задачей программиста