# Основы математической логики и логического программирования

ЛЕКТОР: В.А. Захаров

**Лекция 14.**

# Полнота операционной семантики логических программ. Оператор непосредственного следования. Теоремы полноты.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ ЛОГИЧЕСКИХ ПРОГРАММ

## Проблема полноты

Пусть $\theta$ — правильный ответ на запрос $G_0$ к хорновской логической программе $\mathcal{P}$, т. е. $\mathcal{P} \models \forall \overline{z}\ G_0\theta$.

Существует ли такое успешное SLD-резолютивное вычисление, порожденное запросом $G_0$ к программе $\mathcal{P}$

$$(D_{j_1}, \theta_1, G_1), (D_{j_2}, \theta_2, G_2), \ldots, (D_{j_n}, \theta_n, \square),$$

которое вычисляет ответ $\theta$, т. е. $\theta = (\theta_1\theta_2 \ldots \theta_n)|_{Var_{G_0}}$?

Начнем изучение этого вопроса с более простой задачи. Пусть
$C_0$ — основной атом (т. е. $Var_{C_0} = \emptyset$),
$G_0 = ?C_0$,
$\varepsilon$ — правильный ответ на запрос $?C_0$ к программе $\mathcal{P}$.

Существует ли успешное SLD-резолютивное вычисление, порожденное запросом $?C_0$ к программе $\mathcal{P}$?

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Определение (множество успехов).

Множеством успехов — хорновской логической программы $\mathcal{P}$ называется множество $\mathbf{Succ}_{\mathcal{P}}$ всех тех **основных атомов** $C_0$, для которых запрос $G_0 = ?C_0$ к программе $\mathcal{P}$ порождает успешное SLD-резолютивное вычисление.

Как следует из теоремы корректности операционной семантики и теоремы об основном правильном ответе ,

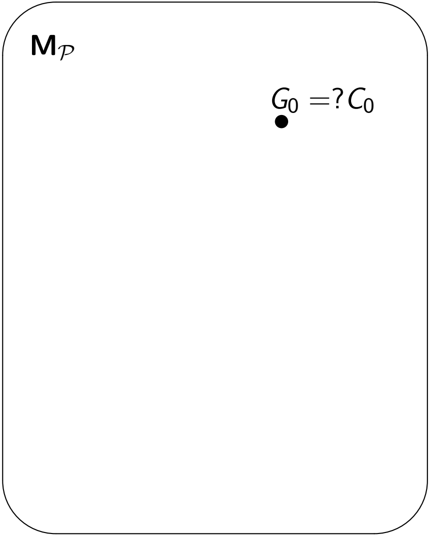
$$C_0 \in \mathbf{Succ}_{\mathcal{P}} \;\Rightarrow\; \mathcal{P} \models C_0 \;\Rightarrow\; C_0 \in M_{\mathcal{P}}.$$

Значит, $\mathbf{Succ}_{\mathcal{P}} \subseteq \mathbf{M}_{\mathcal{P}}$
(т. е. программа успешно вычисляет вычисляет утвердительный ответ на вопрос $?C_0$ только в том случае, когда утверждение $C_0$ истинно в наименьшей $H$-модели).

А верно ли обратное включение $\mathbf{Succ}_{\mathcal{P}} \supseteq \mathbf{M}_{\mathcal{P}}$
(все, что логически следует из программы можно вычислить)?

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

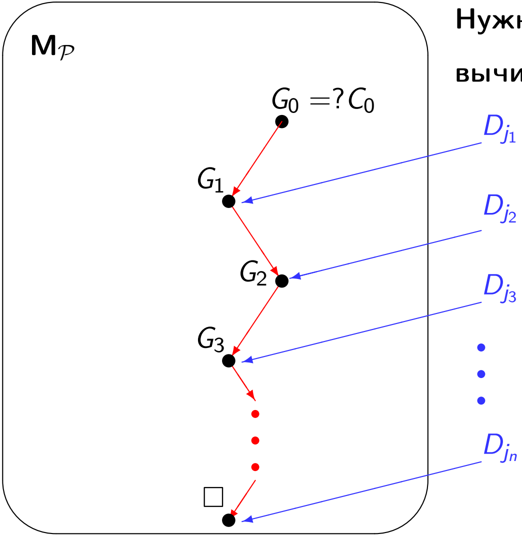
**Наименьшая эрбрановская модель**



# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

Наименьшая эрбрановская модель

Нужно построить вычисление



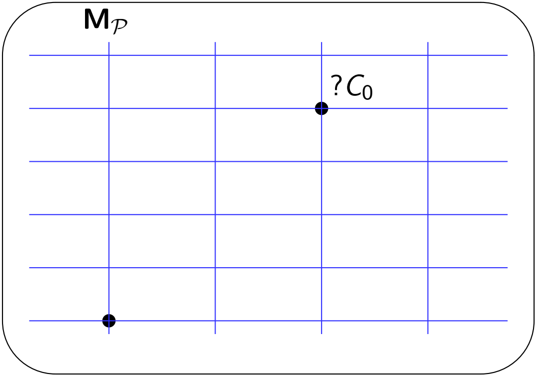
# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

Чтобы запрос $G_0 = ?C_0$ мог «отыскать путь» к пустому запросу $\square$ (успешное вычисление) в наименьшей модели $\mathbf{M}_{\mathcal{P}}$, нам нужно ввести в множестве основных атомов $\mathbf{M}_{\mathcal{P}}$ какие-нибудь ориентиры (систему координат), руководствуясь которыми можно вывести из запроса $G_0$ пустой запрос $\square$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

Чтобы запрос $G_0 = ?C_0$ мог «отыскать путь» к пустому запросу $\square$ (успешное вычисление) в наименьшей модели $\mathbf{M}_{\mathcal{P}}$, нам нужно ввести в множестве основных атомов $\mathbf{M}_{\mathcal{P}}$ какие-нибудь ориентиры (систему координат), руководствуясь которыми можно вывести из запроса $G_0$ пустой запрос $\square$.

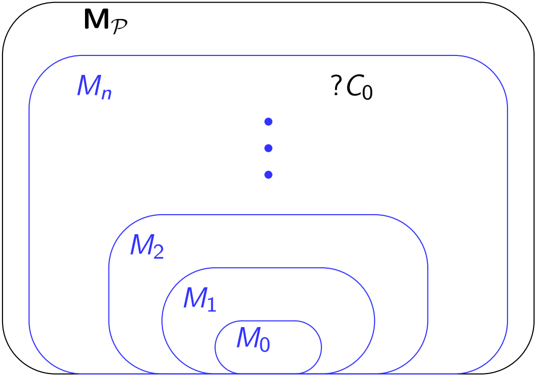
Например, так:



# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

Чтобы запрос $G_0 = ?C_0$ мог «отыскать путь» к пустому запросу $\square$ (успешное вычисление) в наименьшей модели $\mathbf{M}_{\mathcal{P}}$, нам нужно ввести в множестве основных атомов $\mathbf{M}_{\mathcal{P}}$ какие-нибудь ориентиры (систему координат), руководствуясь которыми можно вывести из запроса $G_0$ пустой запрос $\square$.

Или так:



# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

Для создания такой «системы слоев» в наименьшей модели $\mathbf{M}_{\mathcal{P}}$ воспользуемся оператором непосредственного следования .

## Определение

Пусть задана хорновская логическая программа $\mathcal{P}$, состоящая из программных утверждений $D_1, D_2, \dots, D_N$.
Оператором непосредственного следования $T_{\mathcal{P}}$ для программы $\mathcal{P}$ называется отображение

$$T_{\mathcal{P}} \; : \; 2^{B_{\mathcal{P}}} \; \rightarrow \; 2^{B_{\mathcal{P}}},$$

удовлетворяющее следующему соотношению:

для любой эрбрановской интерпретации $I$, $I \subseteq B_{\mathcal{P}}$, имеет место равенство

$$T_{\mathcal{P}}(I) = \{A_0 \; : \; D = A_0 \leftarrow A_1, \dots, A_k \in [\mathcal{P}], \{A_1, \dots, A_k\} \subseteq I\}.$$

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

### Пояснения к определению

$B_{\mathcal{P}}$ — эрбрановский базис программы $\mathcal{P}$, т. е. множество всех **основных** атомов программы.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

$B_{\mathcal{P}}$ — эрбрановский базис программы $\mathcal{P}$, т. е. множество всех **основных** атомов программы.

Таким образом, каждое подмножество $I$, $I \subseteq B_{\mathcal{P}}$ — это эрбрановская интерпретация, в которой истинными объявляются те и только те атомы, которые содержатся в множестве $I$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

В математике (теории множеств) для каждого множества $M$ запись $2^M$ обозначает совокупность всех возможных подмножеств множества $M$, включая само множество $M$ и пустое множество $\emptyset$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

В математике (теории множеств) для каждого множества $M$ запись $2^M$ обозначает совокупность всех возможных подмножеств множества $M$, включая само множество $M$ и пустое множество $\emptyset$.

Таким образом, $2^{B_{\mathcal{P}}}$ — это множество всех эрбрановских интерпретаций программы $\mathcal{P}$,

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

В математике (теории множеств) для каждого множества $M$ запись $2^M$ обозначает совокупность всех возможных подмножеств множества $M$, включая само множество $M$ и пустое множество $\emptyset$.

Таким образом, $2^{B_\mathcal{P}}$ — это множество всех эрбрановских интерпретаций программы $\mathcal{P}$,

и оператор непосредственного следования $T_\mathcal{P} : 2^{B_\mathcal{P}} \to 2^{B_\mathcal{P}}$ преобразует одни эрбрановские интерпретации программы $\mathcal{P}$ в другие эрбрановские интерпретации.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

Как же проводится это преобразование интерпретаций?

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

Как же проводится это преобразование интерпретаций?

$[\mathcal{P}]$ — это множество всех основных примеров программных утверждений (правил и фактов) программы $\mathcal{P}$,

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

Как же проводится это преобразование интерпретаций?

$[\mathcal{P}]$ — это множество всех основных примеров программных утверждений (правил и фактов) программы $\mathcal{P}$,

и каждое утверждение $D = A_0 \leftarrow A_1, A_2, \ldots, A_k$ понимается как высказывание «если $A_1$ и $A_2$ и ... и $A_k$, то $A_0$».

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

Как же проводится это преобразование интерпретаций?

$[\mathcal{P}]$ — это множество всех основных примеров программных утверждений (правил и фактов) программы $\mathcal{P}$,

и каждое утверждение $D = A_0 \leftarrow A_1, A_2, \ldots, A_k$ понимается как высказывание «если $A_1$ и $A_2$ и ... и $A_k$, то $A_0$».

Таким образом,
$T_{\mathcal{P}}(I) = \{A_0 \ : \ D = A_0 \leftarrow A_1, \ldots, A_k \in [\mathcal{P}], \{A_1, \ldots, A_k\} \subseteq I\}$.
— это совокупность всех тех основных атомов $A_0$, которые являются **непосредственными следствиями** истинных в интерпретации $I$ утверждений $A_1, \ldots, A_k$ на основании правил и фактов программы $\mathcal{P}$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пояснения к определению

Значит, оператор $T_{\mathcal{P}} : 2^{B_{\mathcal{P}}} \to 2^{B_{\mathcal{P}}}$ , определяемый соотношением

$$T_{\mathcal{P}}(I) = \{A_0 : D = A_0 \leftarrow A_1, \dots, A_k \in [\mathcal{P}], \{A_1, \dots, A_k\} \subseteq I\},$$

вычисляет для каждой эрбрановской интерпретации $I$ новую интерпретацию $T_{\mathcal{P}}(I)$, состоящую из всех тех атомов, которые являются непосредственными следствиями, извлекаемыми из интерпретации $I$ при помощи фактов и правил программы $\mathcal{P}$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

### Пример 1.

Пусть задана хорновская логическая программа

$$\mathcal{P}: \quad P(f(X)) \leftarrow P(X),$$

сигнатуры $\sigma = \langle Const = \{c\}, Func = \{f\}, Pred = \{P\}\rangle$, состоящая из одного-единственного программного утверждения.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

**Пример 1.**

Пусть задана хорновская логическая программа

$$\mathcal{P}: \quad P(f(X)) \leftarrow P(X),$$

сигнатуры $\sigma = \langle Const = \{c\}, Func = \{f\}, Pred = \{P\} \rangle$, состоящая из одного-единственного программного утверждения.

Если $I = \emptyset$, то $T_{\mathcal{P}}(I) = \emptyset$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 1.

Пусть задана хорновская логическая программа

$$\mathcal{P}: \quad P(f(X)) \leftarrow P(X),$$

сигнатуры $\sigma = \langle Const = \{c\}, Func = \{f\}, Pred = \{P\}\rangle$, состоящая из одного-единственного программного утверждения.

Если $I = \emptyset$, то $T_{\mathcal{P}}(I) = \emptyset$.

Действительно, для любого основного терма $t$ имеем $P(t) \notin I$, и это означает, что правило $P(f(x)) \leftarrow P(x)$ не дает непосредственных следствий из интерпретации $I$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 2.

Пусть задана хорновская логическая программа

$$\mathcal{P}: \quad P(f(X)) \leftarrow P(X),$$

сигнатуры $\sigma = \langle Const = \{c\}, Func = \{f\}, Pred = \{P\}\rangle$, состоящая из одного-единственного программного утверждения.

Если $I = B_{\mathcal{P}}$, то $T_{\mathcal{P}}(I) = B_{\mathcal{P}} \setminus \{P(c)\}$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 2.

Пусть задана хорновская логическая программа

$$\mathcal{P}: \quad P(f(X)) \leftarrow P(X),$$

сигнатуры $\sigma = \langle Const = \{c\}, Func = \{f\}, Pred = \{P\}\rangle$, состоящая из одного-единственного программного утверждения.

Если $I = B_{\mathcal{P}}$, то $T_{\mathcal{P}}(I) = B_{\mathcal{P}} \setminus \{P(c)\}$.

Действительно, коль скоро в интерпретации $I$ истинно все-все-все, то непосредственными следствиями интерпретации $I$ по правилу программы $\mathcal{P}$ будут все атомы вида $P(f(t))$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 2.

Пусть задана хорновская логическая программа

$$\mathcal{P}: \quad P(f(X)) \leftarrow P(X),$$

сигнатуры $\sigma = \langle Const = \{c\}, Func = \{f\}, Pred = \{P\}\rangle$, состоящая из одного-единственного программного утверждения.

Если $I = B_{\mathcal{P}}$, то $T_{\mathcal{P}}(I) = B_{\mathcal{P}} \setminus \{P(c)\}$.

Действительно, коль скоро в интерпретации $I$ истинно все-все-все, то непосредственными следствиями интерпретации $I$ по правилу программы $\mathcal{P}$ будут все атомы вида $P(f(t))$.

Однако программа $\mathcal{P}$ не позволяет получить атом $P(c)$ в качестве непосредственного следствия интерпретации $I$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 3.

Пусть задана хорновская логическая программа

$$
\begin{array}{lll}
\mathcal{P}: & P(X) \leftarrow R(X), P(c); & (1) \\
 & R(b) \leftarrow P(a); & (2) \\
 & R(a) \leftarrow; & (3) \\
 & P(c) \leftarrow; & (4)
\end{array}
$$

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 3.

Пусть задана хорновская логическая программа

$$\begin{array}{lll} \mathcal{P}: & P(X) \leftarrow R(X), P(c); & (1) \\ & R(b) \leftarrow P(a); & (2) \\ & R(a) \leftarrow; & (3) \\ & P(c) \leftarrow; & (4) \end{array}$$

Тогда

$T_{\mathcal{P}}(\emptyset) = \{R(a), P(c)\}$ (факты любой программы не требуют никаких предпосылок, и поэтому являются непосредственными следствиями любой интерпретации)

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 3.

Пусть задана хорновская логическая программа

$$\begin{array}{lll} \mathcal{P}: & P(X) \leftarrow R(X), P(c); & (1) \\ & R(b) \leftarrow P(a); & (2) \\ & R(a) \leftarrow; & (3) \\ & P(c) \leftarrow; & (4) \end{array}$$

Тогда

$T_{\mathcal{P}}(\emptyset) = \{R(a), P(c)\}$

$T^2_{\mathcal{P}}(\emptyset) = T_{\mathcal{P}}(T_{\mathcal{P}}(\emptyset)) = \{R(a), P(c), P(a)\}$(факты (3) и (4) по прежнему являются непосредственными следствиями интерпретации $T_{\mathcal{P}}(\emptyset)$, но теперь кроме них еще одно логическое следствие можно получить по правилу (1))

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 3.

Пусть задана хорновская логическая программа

$$\begin{array}{lll} \mathcal{P}: & P(X) \leftarrow R(X), P(c); & (1) \\ & R(b) \leftarrow P(a); & (2) \\ & R(a) \leftarrow; & (3) \\ & P(c) \leftarrow; & (4) \end{array}$$

Тогда

$T_{\mathcal{P}}(\emptyset) = \{R(a), P(c)\}$

$T^2_{\mathcal{P}}(\emptyset) = T_{\mathcal{P}}(T_{\mathcal{P}}(\emptyset)) = \{R(a), P(c), P(a)\}$

$T^3_{\mathcal{P}}(\emptyset) = \{R(a), P(c), P(a), R(b)\}$(а теперь по правилу (2) добавляется еще одно непосредственное следование)

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 3.

Пусть задана хорновская логическая программа

$$\begin{array}{lll} \mathcal{P}: & P(X) \leftarrow R(X), P(c); & (1) \\ & R(b) \leftarrow P(a); & (2) \\ & R(a) \leftarrow; & (3) \\ & P(c) \leftarrow; & (4) \end{array}$$

Тогда

$T_{\mathcal{P}}(\emptyset) = \{R(a), P(c)\}$

$T^2_{\mathcal{P}}(\emptyset) = T_{\mathcal{P}}(T_{\mathcal{P}}(\emptyset)) = \{R(a), P(c), P(a)\}$

$T^3_{\mathcal{P}}(\emptyset) = \{R(a), P(c), P(a), R(b)\}$

$T^4_{\mathcal{P}}(\emptyset) = \{R(a), P(c), P(a), R(b), P(b)\}$(и вновь по правилу (2) добавляется еще одно непосредственное следование)

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

## Пример 3.

Пусть задана хорновская логическая программа

$$\mathcal{P}: \quad \begin{array}{ll} P(X) \leftarrow R(X), P(c); & (1) \\ R(b) \leftarrow P(a); & (2) \\ R(a) \leftarrow; & (3) \\ P(c) \leftarrow; & (4) \end{array}$$

Тогда

$T_{\mathcal{P}}(\emptyset) = \{R(a), P(c)\}$

$T^2_{\mathcal{P}}(\emptyset) = T_{\mathcal{P}}(T_{\mathcal{P}}(\emptyset)) = \{R(a), P(c), P(a)\}$

$T^3_{\mathcal{P}}(\emptyset) = \{R(a), P(c), P(a), R(b)\}$

$T^4_{\mathcal{P}}(\emptyset) = \{R(a), P(c), P(a), R(b), P(b)\}$

$T^5_{\mathcal{P}}(\emptyset) = T^4_{\mathcal{P}}(\emptyset)$, и вообще $T^n_{\mathcal{P}}(\emptyset) = T^4_{\mathcal{P}}(\emptyset)$ при любых $n \geq 5$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

Оператор непосредственного следования $T_{\mathcal{P}}$ обладает рядом полезных свойств. Например, $T_{\mathcal{P}}$ монотонен.

**Лемма (о монотонности $T_{\mathcal{P}}$)**

$$I \subseteq J \;\Rightarrow\; T_{\mathcal{P}}(I) \subseteq T_{\mathcal{P}}(J)$$

**Доказательство:**

Пусть $A_0 \in T_{\mathcal{P}}(I)$. Тогда по определению оператора $T_{\mathcal{P}}$ существует такой основной пример программного утверждения $D = A_0 \leftarrow A_1, \dots, A_n \in [\mathcal{P}]$, что $\{A_1, \dots, A_n\} \subseteq I$.
Поскольку $I \subseteq J$, верно $\{A_1, \dots, A_n\} \subseteq J$, и поэтому $A_0 \in T_{\mathcal{P}}(J)$.

$\square$

# Оператор непосредственного следования

Условимся $n$-кратную композицию оператора непосредственного следования обозначать $T^n_{\mathcal{P}}$, т. е. $T^n_{\mathcal{P}}(I) = \underbrace{T_{\mathcal{P}}(T_{\mathcal{P}}(\dots T_{\mathcal{P}}}_{n \text{ раз}}(I)\dots))$.

### Теорема (об устройстве $\mathbf{M}_{\mathcal{P}}$)

Для любой хорновской логической программы $\mathcal{P}$ верно равенство

$$\mathbf{M}_{\mathcal{P}} = \bigcup_{i=0}^{\infty} T^i_{\mathcal{P}}(\emptyset) \ .$$

# Оператор непосредственного следования

## Доказательство:

Вначале покажем, что справедливы следующие включения:

$$T^0_{\mathcal{P}}(\emptyset) \subseteq T^1_{\mathcal{P}}(\emptyset) \subseteq T^2_{\mathcal{P}}(\emptyset) \subseteq \dots \subseteq T^i_{\mathcal{P}}(\emptyset) \subseteq T^{i+1}_{\mathcal{P}}(\emptyset) \subseteq \dots$$

Это можно показать по индукции.

**Базис индукции.** $T^0_{\mathcal{P}}(\emptyset) = \emptyset \subseteq T^1_{\mathcal{P}}(\emptyset)$.

**Индуктивный переход.** Если $T^{i-1}_{\mathcal{P}}(\emptyset) \subseteq T^i_{\mathcal{P}}(\emptyset)$, то, согласно лемме о монотонности оператора $T_{\mathcal{P}}$, получаем

$$T^i_{\mathcal{P}}(\emptyset) = T_{\mathcal{P}}(T^{i-1}_{\mathcal{P}}(\emptyset)) \subseteq T_{\mathcal{P}}(T^i_{\mathcal{P}}(\emptyset)) = T^{i+1}_{\mathcal{P}}(\emptyset).$$

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

Доказательство: 1. Покажем $\mathbf{M}_{\mathcal{P}} \supseteq \bigcup_{i=0}^{\infty} T^{i}_{\mathcal{P}}(\emptyset)$.

Индукцией по $i$ покажем, что $\mathbf{M}_{\mathcal{P}} \supseteq T^{i}_{\mathcal{P}}(\emptyset)$.

**Базис индукции.** $T^{0}_{\mathcal{P}}(\emptyset) = \emptyset \subseteq \mathbf{M}_{\mathcal{P}}$.

**Индуктивный переход.** Допустим $A_0 \in T^{i+1}_{\mathcal{P}}(\emptyset)$. По определению оператора $T_{\mathcal{P}}$ это означает, что существует такой основной пример программного утверждения $D = A_0 \leftarrow A_1, \ldots, A_n \in [\mathcal{P}]$, что $\{A_1, \ldots, A_n\} \subseteq T^{i}_{\mathcal{P}}(\emptyset)$. Согласно индуктивному предположению $\{A_1, \ldots, A_n\} \subseteq \mathbf{M}_{\mathcal{P}}$. Тогда, согласно лемме о модели для логической программы, получаем $A_0 \in \mathbf{M}_{\mathcal{P}}$.

# ОПЕРАТОР НЕПОСРЕДСТВЕННОГО СЛЕДОВАНИЯ

**Доказательство:** 2. Покажем $\mathbf{M}_{\mathcal{P}} \subseteq \bigcup_{i=0}^{\infty} T^i_{\mathcal{P}}(\emptyset)$.

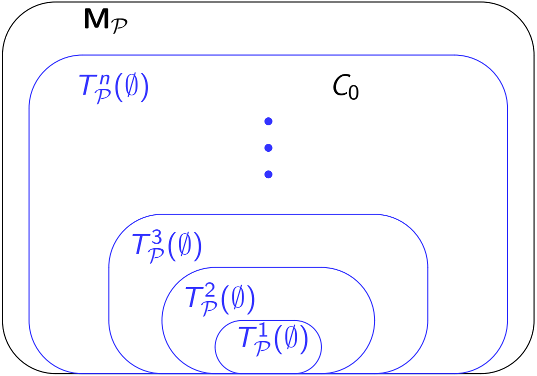
Для этого достаточно показать, что $\bigcup_{i=0}^{\infty} T^i_{\mathcal{P}}(\emptyset)$ — одна из моделей программы $\mathcal{P}$ (почему этого достаточно? ).

Воспользуемся леммой о модели для логической программы.
Пусть $D = A_0 \leftarrow A_1, \dots, A_n$ — произвольный пример программного утверждения из $[\mathcal{P}]$ и $\{A_1, \dots, A_n\} \subseteq \bigcup_{i=0}^{\infty} T^i_{\mathcal{P}}(\emptyset)$.
Тогда, согласно свойству оператора $T_{\mathcal{P}}$, существует такое $m$, что $\{A_1, \dots, A_n\} \subseteq T^m_{\mathcal{P}}(\emptyset)$ (почему это так? ).
Тогда, согласно определению оператора $T_{\mathcal{P}}$, верно $A_0 \in T^{m+1}_{\mathcal{P}}(\emptyset)$, и, поэтому, $A_0 \in \bigcup_{i=0}^{\infty} T^i_{\mathcal{P}}(\emptyset)$.

Значит, $\bigcup_{i=0}^{\infty} T^i_{\mathcal{P}}(\emptyset)$ — модель для $\mathcal{P}$. $\mathbf{M}_{\mathcal{P}} \subseteq \bigcup_{i=0}^{\infty} T^i_{\mathcal{P}}(\emptyset)$. $\square$

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

Теперь мы знаем, что наименьшая эрбрановская модель $\mathbf{M}_{\mathcal{P}}$ имеет слоистое устройство.




И это поможет нам доказать индукцией по числу слоев, что для каждого основного атома $C_0 \in \mathbf{M}_{\mathcal{P}}$ запрос $?C_0$ имеет успешное вычисление.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Теорема полноты (первая)

$$\mathbf{M}_{\mathcal{P}} \subseteq \mathbf{Succ}_{\mathcal{P}}$$

Доказательство проведем в два этапа

Этап 1. Покажем, что если $C_0 \in \mathbf{M}_{\mathcal{P}}$, то запрос $?C_0$ к логической программе $[\mathcal{P}]$, состоящей из всех возможных **основных примеров** программы $\mathcal{P}$, имеет успешное вычисление.

Этап 2. Покажем, что если основной запрос $?C_0$ к логической программе $[\mathcal{P}]$ имеет успешное вычисление, то и запрос $?C_0$ к логической программе $\mathcal{P}$ также имеет успешное вычисление.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

**Доказательство** (Этап 1)

Пусть $C_0 \in \mathbf{M}_{\mathcal{P}}$. По теореме об устройстве наименьшей модели, верно равенство $\mathbf{M}_{\mathcal{P}} = \bigcup_{i=0}^{\infty} T^i_{\mathcal{P}}(\emptyset)$, и поэтому существует такое $m$, что $C_0 \in T^m_{\mathcal{P}}(\emptyset)$.
Доказательство существования успешного вычисления проведем индукцией по $m$: $C_0 \in T^m_{\mathcal{P}}(\emptyset)$.

**Базис индукции.** $m = 1$. Если $C_0 \in T_{\mathcal{P}}(\emptyset)$, то в программе $[\mathcal{P}]$ существует факт $C_0 \leftarrow$. Значит, запрос $?C_0$ к программе $[\mathcal{P}]$ имеет успешное вычисление $(C_0 \leftarrow, \varepsilon, \square)$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство (Этап 1)

**Индуктивный переход.** $m \to m+1$. Пусть $C_0 \in T_{\mathcal{P}}^{m+1}(\emptyset)$. Тогда в программе $[\mathcal{P}]$ существует такое утверждение $D_0 = C_0 \leftarrow C_1, C_2, \ldots, C_n$, что $\{C_1, \ldots, C_n\} \subseteq T_{\mathcal{P}}^{m}(\emptyset)$. Тогда

- Запрос $?C_0$ к программе $[\mathcal{P}]$ имеет SLD-резольвенту $?C_1, C_2, \ldots, C_n$;
- Согласно индуктивному предположению, каждый запрос $?C_i$, где $1 \leq i \leq n$, имеет успешное SLD-резолютивное вычисление

$$(D_{i1}, \varepsilon, G_{i1}), \ldots, (D_{ik_i}, \varepsilon, \square).$$

почему в качестве унификаторов здесь стоит пустая подстановка $\varepsilon$?

# Полнота операционной семантики

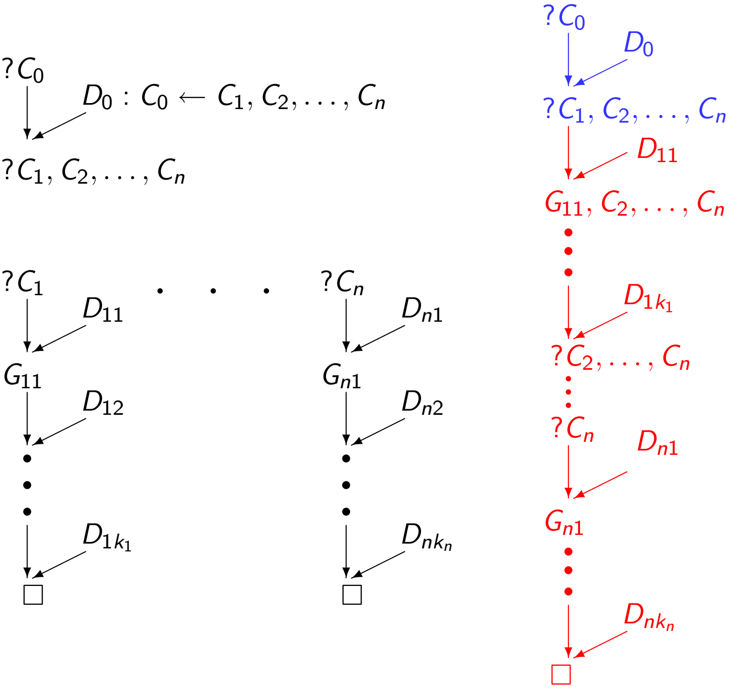
**Доказательство** (Этап 1)

**Индуктивный переход.** Учитывая, что подцели $C_1, C_2, \ldots, C_n$ — это основные атомы, и, следовательно, они не изменяются в результате применения к ним любых подстановок, получаем успешное вычисление запроса $?C_0$ к программе $[\mathcal{P}]$:

$$
\begin{tikzcd}
?\,C_0 \arrow[d] & D_0 : C_0 \leftarrow C_1, C_2, \ldots, C_n \arrow[dl] \\
?\,C_1, C_2, \ldots, C_n &
\end{tikzcd}
$$

$$
\begin{tikzcd}
?C_0 \arrow[d] & D_0 : C_0 \leftarrow C_1, C_2, \ldots, C_n \arrow[dl] \\
?C_1, C_2, \ldots, C_n &
\end{tikzcd}
$$

$$
\begin{tikzcd}
?C_1 \arrow[d] & D_{11} \arrow[dl] & \cdots & ?C_n \arrow[d] & D_{n1} \arrow[dl] \\
G_{11} \arrow[d] & D_{12} \arrow[dl] & & G_{n1} \arrow[d] & D_{n2} \arrow[dl] \\
\vdots \arrow[d] & D_{1k_1} \arrow[dl] & & \vdots \arrow[d] & D_{nk_n} \arrow[dl] \\
\square & & & \square &
\end{tikzcd}
$$


$?C_0$
$D_0 : C_0 \leftarrow C_1, C_2, \ldots, C_n$
$?C_1, C_2, \ldots, C_n$
$?C_1$
$D_{11}$
$G_{11}$
$D_{12}$
$D_{1k_1}$
$?C_n$
$D_{n1}$
$G_{n1}$
$D_{n2}$
$D_{nk_n}$
$?C_0$
$D_0$
$?C_1, C_2, \ldots, C_n$
$D_{11}$
$G_{11}, C_2, \ldots, C_n$
$D_{1k_1}$
$?C_2, \ldots, C_n$
$?C_n$
$D_{n1}$
$G_{n1}$
$D_{nk_n}$

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство (Этап 1)

**Индуктивный переход.** Учитывая, что подцели $C_1, C_2, \ldots, C_n$ — это основные атомы, и, следовательно, они не изменяются в результате применения к ним любых подстановок, получаем успешное вычисление запроса $?C_0$ к программе $[\mathcal{P}]$:

И этим завершается построение успешного вычисления запроса $?C_0$ к программе $[\mathcal{P}]$.
Конец 1-го этапа док-ва.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

**Доказательство** (Этап 2)

Теперь необходимо показать, что если основной запрос $?C_0$ к программе $[\mathcal{P}]$ имеет успешное вычисление, то этот же запрос $?C_0$ к программе $\mathcal{P}$ также имеет успешное вычисление.

Мы докажем, на самом деле, более общее утверждение, которое сформулируем в виде отдельной леммы.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Лемма о подъеме (для логических программ)

Пусть $G_0$ — запрос к логической программе $\mathcal{P}$, $y_1, \ldots, y_k$ — целевые переменные, $t_1, \ldots, t_k$ — основные термы.
Пусть запрос $G_0' = G_0\{y_1/t_1, \ldots, y_k/t_k\}$ к логической программе $[\mathcal{P}]$ имеет успешное вычисление

$$comp' = (D_1', \varepsilon, G_1'), (D_2', \varepsilon, G_2'), \ldots, (D_N', \varepsilon, \Box),$$

в котором каждое программное утверждение $D_i' \in [\mathcal{P}]$ является основным примером $D_i' = D_i\theta_i$ некоторого утверждения программы $\mathcal{P}$.
Тогда запрос $G_0$ к логической программе $\mathcal{P}$ также имеет успешное вычисление

$$comp = (D_1, \eta_1, G_1), (D_2, \eta_2, G_2), \ldots, (D_N, \eta_N, \Box),$$

и при этом существует подстановка $\rho$, для которой верно равенство $\{y_1/t_1, \ldots, y_k/t_k\}\theta_1\theta_2 \ldots \theta_N = \eta_1\eta_2 \ldots \eta_N\rho$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Лемма о подъеме (для логических программ)

$G_0$

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Лемма о подъеме (для логических программ)

$$\begin{tikzcd} G_0 \arrow[d, "\mu"] \\ G_0\mu \end{tikzcd}$$

# Полнота операционной семантики

## Лемма о подъеме (для логических программ)

$$
\begin{tikzcd}
G_0 \arrow[ddd, "\mu"] & & & & & & & & \\
& D_1 \arrow[d, "\theta_1"] & & D_2 \arrow[d, "\theta_2"] & & & & D_N \arrow[d, "\theta_N"] & \\
& D_1\theta_1 \arrow[dr, "\varepsilon"'] & & D_2\theta_2 \arrow[dr, "\varepsilon"'] & & & & D_N\theta_N \arrow[dr, "\varepsilon"'] & \\
G_0\mu \arrow[rr] & & G'_1 \arrow[rr] & & G'_2 & \cdots & G'_N \arrow[rr] & & \square
\end{tikzcd}
$$

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Лемма о подъеме (для логических программ)

$$
\begin{tikzcd}
G_0 \arrow[rr] \arrow[ddd, "\mu"] & & G_1 \arrow[rr] & & G_2 & \cdots & G_N \arrow[rr] & & \square \\
& D_1 \arrow[ur, "\eta_1"'] \arrow[d, "\theta_1"] & & D_2 \arrow[ur, "\eta_2"'] \arrow[d, "\theta_2"] & & & & D_N \arrow[ur, "\eta_N"'] \arrow[d, "\theta_N"] & \\
& D_1\theta_1 \arrow[dr, "\varepsilon"'] & & D_2\theta_2 \arrow[dr, "\varepsilon"'] & & & & D_N\theta_N \arrow[dr, "\varepsilon"'] & \\
G_0\mu \arrow[rr] & & G'_1 \arrow[rr] & & G'_2 & \cdots & G'_N \arrow[rr] & & \square
\end{tikzcd}
$$

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Лемма о подъеме (для логических программ)

$$
\begin{tikzcd}
G_0 \arrow[rr] \arrow[ddd, "\mu"] & & G_1 \arrow[rr] & & G_2 & \cdots & G_N \arrow[rr] & & \square \\
& D_1 \arrow[ur, "\eta_1"'] \arrow[d, "\theta_1"] & & D_2 \arrow[ur, "\eta_2"'] \arrow[d, "\theta_2"] & & & & D_N \arrow[ur, "\eta_N"'] \arrow[d, "\theta_N"] & \\
& D_1\theta_1 \arrow[dr, "\varepsilon"'] & & D_2\theta_2 \arrow[dr, "\varepsilon"'] & & & & D_N\theta_N \arrow[dr, "\varepsilon"'] & \\
G_0\mu \arrow[rr] & & G_1' \arrow[rr] & & G_2' & \cdots & G_N' \arrow[rr] & & \square
\end{tikzcd}
$$

$$\mu\theta_{\mathbf{1}}\theta_{\mathbf{2}}\ldots\theta_{\mathbf{N}} = \eta_{\mathbf{1}}\eta_{\mathbf{2}}\ldots\eta_{\mathbf{N}}\rho$$

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

**Доказательство** (Этап 2)

Для нужд доказываемой теоремы достаточно использовать тот вариант леммы о подъеме, когда запрос $G_0$ состоит из одной-единственной подцели.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

Воспользуемся индукцией по длине успешного вычисления. Обозначим символом $\mu$ подстановку $\{y_1/t_1, \ldots, y_k/t_k\}$.

Тогда $G_0' = G_0\mu$.

**Базис индукции.** $N = 1$.

# Полнота операционной семантики

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.

$$\begin{tikzcd}
G'_0 \arrow[dd] & \\
& D'_1 \arrow[dl, "\varepsilon"] \\
\square &
\end{tikzcd}$$

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.

$$\begin{tikzcd} ?\,C_0\mu \arrow[dd] & \\ & A_0\theta_1 \leftarrow \arrow[dl, "\varepsilon"] \\ \square & \end{tikzcd}$$

Это возможно, если

запрос $G'_0$ состоит из одной подцели,
т. е. $G_0 = C_0$, $G'_0 = C_0\mu$,

а программное утверждение $D'_1$ — это факт,
т. е. $D_1 = A_0 \leftarrow$, $D'_0 = A_0\theta_1 \leftarrow$,

и при этом $C_0\mu = A_0\theta_1$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.

$$\begin{tikzcd}
?C_0\mu \arrow[dd] & \\
& A_0\theta_1 \leftarrow \arrow[dl, "\varepsilon"] \\
\square &
\end{tikzcd}$$

Итак, $G_0 = C_0$, $D_1 = A_0 \leftarrow$, $C_0\mu = A_0\theta_1$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.

$$\begin{tikzcd}
?C_0\mu \arrow[dd] & \\
 & A_0\theta_1 \leftarrow \arrow[dl, "\varepsilon"] \\
\square &
\end{tikzcd}$$

Итак, $G_0 = C_0$, $D_1 = A_0 \leftarrow$, $C_0\mu = A_0\theta_1$.

Но при этом $Var_{G_0} \cap Var_{D_1} = \emptyset$ (почему? ),

и $Var_{C_0\mu} = \emptyset$ (почему? ).

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.

$$\begin{tikzcd} ?C_0\mu \arrow[dd] & \\ & A_0\theta_1 \leftarrow \arrow[dl, "\varepsilon"] \\ \square & \end{tikzcd}$$

Итак, $G_0 = C_0$, $D_1 = A_0 \leftarrow$, $C_0\mu = A_0\theta_1$.

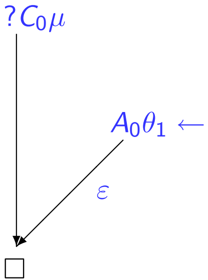
Но при этом $Var_{G_0} \cap Var_{D_1} = \emptyset$ (почему? ),
и $Var_{C_0\mu} = \emptyset$ (почему? ).

Значит, подстановка $\mu$ не влияет на $A_0$,
и подстановка $\theta_1$ не влияет на $C_0\mu$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.




Итак, $G_0 = C_0$, $D_1 = A_0 \leftarrow$, $C_0\mu = A_0\theta_1$.

Но при этом $Var_{G_0} \cap Var_{D_1} = \emptyset$ (почему? ),
и $Var_{C_0\mu} = \emptyset$ (почему? ).

Значит, подстановка $\mu$ не влияет на $A_0$,
и подстановка $\theta_1$ не влияет на $C_0\mu$.

Значит, $C_0\mu\theta_1 = C_0\mu = A_0\theta_1 = A_0\mu\theta_1$,
т. е. атомы $C_0$ и $A_0$ имеют унификатор $\mu\theta_1$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.

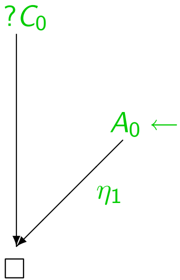
$$
\begin{tikzcd}
?C_0\mu \arrow[dd] & \\
& A_0\theta_1 \leftarrow \arrow[dl, "\varepsilon"] \\
\square &
\end{tikzcd}
$$

Итак, $G_0 = C_0$, $D_1 = A_0 \leftarrow$ и $C_0\mu\theta_1 = A_0\mu\theta_1$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.



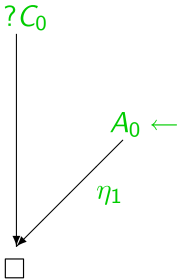

Итак, $G_0 = C_0$, $D_1 = A_0 \leftarrow$ и $C_0\mu\theta_1 = A_0\mu\theta_1$.

Значит, атомы $C_0$ и $A_0$ имеют НОУ $\eta_1$,
и тогда запрос $G_0 = ?C_0$ и факт $A_0 \leftarrow$
порождают SLD-резольвенту $\square$ при помощи $\eta_1$.

# Полнота операционной семантики

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.




Итак, $G_0 = C_0$, $D_1 = A_0 \leftarrow$ и $C_0\mu\theta_1 = A_0\mu\theta_1$.

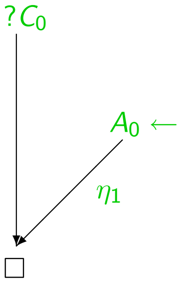
Значит, атомы $C_0$ и $A_0$ имеют НОУ $\eta_1$, и тогда запрос $G_0 = ?C_0$ и факт $A_0 \leftarrow$ порождают SLD-резольвенту $\square$ при помощи $\eta_1$.

Получаем успешное вычисление запроса $G_0 = ?C_0$ к программе $\mathcal{P}$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Базис индукции.** $N = 1$.




Итак, $G_0 = C_0$, $D_1 = A_0 \leftarrow$ и $C_0\mu\theta_1 = A_0\mu\theta_1$.

Значит, атомы $C_0$ и $A_0$ имеют НОУ $\eta_1$,
и тогда запрос $G_0 = ?C_0$ и факт $A_0 \leftarrow$
порождают SLD-резольвенту $\Box$ при помощи $\eta_1$.

Получаем успешное вычисление запроса $G_0 = ?C_0$
к программе $\mathcal{P}$.

$\eta_1 \in НОУ(C_0, A_0)$, а $\mu\theta_1$ — унификатор $C_0$ и $A_0$.
Значит, $\mu\theta_1 = \eta_1\rho$ для некоторой подстановки $\rho$.

Базис индукции доказан.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$$
\begin{tikzcd}
G'_0 \arrow[dd] & \\
& D'_1 \arrow[dl, "\varepsilon"] \\
G'_1 \arrow[d] & \\
\vdots\ \theta_2 & \\
N \text{ шагов} & \\
\vdots\ \theta_N \arrow[d] & \\
\square &
\end{tikzcd}
$$

# Полнота операционной семантики

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$?C'_1, \ldots, C'_m$

$A'_0 \leftarrow A'_1, \ldots, A'_n$

$\varepsilon$

$?A'_1, \ldots, A'_n, C'_2, \ldots, C'_m$

$\vdots\ \theta_2$

$N$ шагов

$\vdots\ \theta_N$

$\square$

Чтобы сделать 1-ый шаг вычисления, нужно, чтобы

$G'_0 = C'_1, \ldots, C'_m$,

$D'_1 = A'_0 \leftarrow A'_1, \ldots, A'_n$,

и при этом $C'_1 = A'_0$.

# Полнота операционной семантики

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$?(C_1, \dots, C_m)\mu$

$\downarrow \quad (A_0 \leftarrow A_1, \dots, A_n)\theta_1 \quad \varepsilon$

$?(A_1, \dots, A_n)\theta_1, (C_2, \dots, C_m)\mu$

$\vdots \; \theta_2$

$N$ шагов

$\vdots \; \theta_N$

$\square$

Чтобы сделать 1-ый шаг вычисления, нужно, чтобы

$G'_0 = C'_1, \dots, C'_m$,

$D'_1 = A'_0 \leftarrow A'_1, \dots, A'_n$,

и при этом $C'_1 = A'_0$.

Значит,

$G_0 = C_1, \dots, C_m$, $D_1 = A_0 \leftarrow A_1, \dots, A_n$,

и при этом

$C'_1, \dots, C'_m = (C_1, \dots, C_m)\mu$,

$A'_0 \leftarrow A'_1 \dots, A'_n = (A_0 \leftarrow A_1, \dots, A_n)\theta_1$ и

$C_1\mu = A_0\theta_1$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$$
\begin{array}{l}
?(C_1, \dots, C_m)\mu \\
\quad (A_0 \leftarrow A_1, \dots, A_n)\theta_1 \\
\downarrow \; \swarrow \varepsilon \\
?(A_1, \dots, A_n)\theta_1, (C_2, \dots, C_m)\mu \\
\downarrow \\
\vdots \; \theta_2 \\
N \text{ шагов} \\
\vdots \; \theta_N \\
\downarrow \\
\square
\end{array}
$$

Так же, как и в базисе индукции, можно убедиться в том, что

$\theta_1$ не влияет на $C_1, \dots, C_m$,

а $\mu$ не влияет на $A_0, A_1, \dots, A_n$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$$\begin{tikzcd}
?(C_1, \dots, C_m)\mu \arrow[d] & (A_0 \leftarrow A_1, \dots, A_n)\theta_1 \arrow[dl, "\varepsilon"] \\
?(A_1, \dots, A_n)\theta_1, (C_2, \dots, C_m)\mu \arrow[d, dotted, "\theta_2"] & \\
N \text{ шагов} \arrow[d, dotted, "\theta_N"] & \\
\Box &
\end{tikzcd}$$

Так же, как и в базисе индукции, можно убедиться в том, что

$\theta_1$ не влияет на $C_1, \dots, C_m$,

а $\mu$ не влияет на $A_0, A_1, \dots, A_n$.

Поэтому $C_1\mu = A_0\theta_1$ влечет $C_1\mu\theta_1 = A_0\mu\theta_1$,

и $(A_1, \dots, A_n)\theta_1, (C_2, \dots, C_m)\mu = (A_1, \dots, A_n, C_2, \dots, C_m)\mu\theta_1$

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$?(C_1, \dots, C_m)\mu$

$(A_0 \leftarrow A_1, \dots, A_n)\theta_1$

$\varepsilon$

$?(A_1, \dots, A_n, C_2, \dots, C_m)\mu\theta_1$

$\theta_2$

$N$ шагов

$\theta_N$

$\square$

Так же, как и в базисе индукции, можно убедиться в том, что

$\theta_1$ не влияет на $C_1, \dots, C_m$,

а $\mu$ не влияет на $A_0, A_1, \dots, A_n$.

Поэтому $C_1\mu = A_0\theta_1$ влечет $C_1\mu\theta_1 = A_0\mu\theta_1$,

и $(A_1, \dots, A_n)\theta_1, (C_2, \dots, C_m)\mu = (A_1, \dots, A_n, C_2, \dots, C_m)\mu\theta_1$

Т. к. $C_1\mu\theta_1 = A_0\mu\theta_1$,

существует такая $\eta_1 \in HOY(C_1, A_0)$,

что $\mu\theta_1 = \eta_1\rho_1$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$$\begin{tikzcd}
?C_1, \dots, C_m \arrow[dd] & \\
& A_0 \leftarrow A_1, \dots, A_n \arrow[dl, "\eta_1"] \\
?(A_1, \dots, A_n, C_2, \dots, C_m)\eta_1 &
\end{tikzcd}$$

НОУ $\eta_1$ порождает SLD-резольвенту $G_1 = ?(A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$ запроса $G_0 = ?C_1, \dots, C_n$ и правила $A_0 \leftarrow A_1, \dots, A_n$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$$
\begin{tikzcd}
?C_1, \dots, C_m \arrow[dd] & \\
& A_0 \leftarrow A_1, \dots, A_n \arrow[dl, "\eta_1"] \\
?(A_1, \dots, A_n, C_2, \dots, C_m)\eta_1 &
\end{tikzcd}
$$

НОУ $\eta_1$ порождает SLD-резольвенту
$G_1 = ?(A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$
запроса $G_0 = ?C_1, \dots, C_n$ и
правила $A_0 \leftarrow A_1, \dots, A_n$.

Поскольку $\mu\theta_1 = \eta_1\rho_1$,
запрос $G_1' = (A_1, \dots, A_n, C_2, \dots, C_m)\mu\theta_1$
является основным примером
запроса $G_1 = (A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$?C_1, \dots, C_m$

$A_0 \leftarrow A_1, \dots, A_n$

$\eta_1$

$?(A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$

$\eta_2$

$N$ шагов

$\eta_N$

$\square$

НОУ $\eta_1$ порождает SLD-резольвенту $G_1 = ?(A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$ запроса $G_0 = ?C_1, \dots, C_n$ и правила $A_0 \leftarrow A_1, \dots, A_n$.

Поскольку $\mu\theta_1 = \eta_1\rho_1$, запрос $G_1' = (A_1, \dots, A_n, C_2, \dots, C_m)\mu\theta_1$ является основным примером запроса $G_1 = (A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$.

Тогда по индуктивной гипотезе запрос $G_1 = (A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$ имеет успешное вычисление, и при этом $\rho_1\theta_2 \dots \theta_N = \eta_2 \dots \eta_N\rho$ .

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$?C_1, \dots, C_m$

$A_0 \leftarrow A_1, \dots, A_n$

$\eta_1$

$?(A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$

$\vdots \ \eta_2$

$N$ шагов

$\vdots \ \eta_N$

$\square$

Итак, получаем, что запрос

$G_0 = ?C_1, C_2, \dots, C_m$

имеет успешное вычисление
с последовательностью подстановок

$\eta_1, \eta_2, \dots, \eta_N$

и при этом

$\mu\theta_1 = \eta_1\rho_1$ и $\rho_1\theta_2 \dots \theta_N = \eta_2 \dots \eta_N\rho$ .

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Доказательство леммы о подъеме

**Индуктивный переход.** $N \to N+1$.

$?C_1, \dots, C_m$

$A_0 \leftarrow A_1, \dots, A_n$

$\eta_1$

$?(A_1, \dots, A_n, C_2, \dots, C_m)\eta_1$

$\vdots\ \eta_2$

$N$ шагов

$\vdots\ \eta_N$

$\square$

Итак, получаем, что запрос

$G_0 = ?C_1, C_2, \dots, C_m$

имеет успешное вычисление
с последовательностью подстановок

$\eta_1, \eta_2, \dots, \eta_N$

и при этом

$\mu\theta_1 = \eta_1\rho_1$ и $\rho_1\theta_2 \dots \theta_N = \eta_2 \dots \eta_N\rho$ .

Таким образом,

$\mu\theta_1\theta_2 \dots \theta_N = \eta_1\rho_1\theta_2 \dots \theta_N =$

$= \eta_1\eta_2 \dots \eta_N\rho$.

Что и требовалось доказать.

# Полнота операционной семантики

Лемма о подъеме для логических программ доказана, и вместе с ней завершена 2-я часть доказательства теоремы.

Таким образом, $\mathbf{M}_{\mathcal{P}} \subseteq \mathbf{Succ}_{\mathcal{P}}$

$\square$

## Следствие 1.

$$\mathbf{M}_{\mathcal{P}} = \mathbf{Succ}_{\mathcal{P}}$$

## Следствие 2.

Пусть $C_1, \dots, C_n$ — основные атомы, и $\mathcal{P}$ — хорновская логическая программа.

Тогда $\mathcal{P} \models C_1 \& \dots \& C_n$ тогда и только тогда, когда запрос $?C_1, \dots, C_n$ к программе $\mathcal{P}$ имеет успешное SLD-резолютивное вычисление.

доказательство — самостоятельно.

# Полнота операционной семантики

## Теорема полноты (главная).

Пусть $\theta$ — правильный ответ на запрос $G$ к хорновской логической программе $\mathcal{P}$.

Тогда существует такой вычисленный ответ $\eta$ на запрос $G$ к программе $\mathcal{P}$, что $\theta = \eta\rho$ для некоторой подстановки $\rho$.

## Доказательство.

Пусть $\theta = \{y_1/t_1, \dots, y_k/t_k\}$ и пусть $\bigcup\limits_{i=1}^{k} Var_{t_i} = \{z_1, \dots, z_m\}$.
Выберем некоторое множество констант $\{c_1, \dots, c_m\}$, **не содержащихся** в запросе $G$, в термах подстановки $\theta$ и в программе $\mathcal{P}$, и рассмотрим подстановку $\lambda = \{z_1/c_1, \dots, z_m/c_m\}$.

Т. к. $\theta$ — правильный ответ, то $\mathcal{P} \models \forall z_1 \dots \forall z_m G\theta$.
Поэтому $G\theta\lambda$ — основной пример запроса $G$ и $\mathcal{P} \models G\theta\lambda$.

# Полнота операционной семантики

## Доказательство.

А раз $G\theta\lambda$ — основной пример запроса $G$ и $\mathcal{P} \models G\theta\lambda$, то, согласно следствию 2 и лемме о подъеме для логических программ , запрос $G$ имеет успешное $SLD$-резолютивное вычисление с таким вычисленным ответом $\eta$, что

$$\theta\lambda = \eta\rho'$$

для некоторой подстановки $\rho'$.

Т. к. константы $c_1, \ldots, c_m$ не входят в состав запроса $G$ и программы $\mathcal{P}$, эти константы не входят в состав термов вычисленного ответа $\eta$, и, значит, могут содержаться только в подстановке $\rho'$.

# Полнота операционной семантики

**Доказательство.**

А теперь заменим в левой и правой частях равенства

$$\theta\lambda = \eta\rho'$$

все символы $c_1, \dots, c_m$ на символы $z_1, \dots, z_m$.

Тогда в левой части равенства подстановка $\lambda = \{z_1/c_1, \dots, z_m/c_m\}$ превращается в пустую подстановку $\varepsilon$.

А поскольку константы $c_1, \dots, c_m$ могут содержаться только в подстановке $\rho'$, в правой части равенства подстановка $\rho'$ превращается в некоторую новую подстановку $\rho$.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

**Доказательство.**

В итоге равенство

$$\theta\lambda = \eta\rho'$$

превращается в равенство

$$\theta = \eta\rho\ .$$

$\square$

Содержательный смысл теоремы полноты : всякий правильный ответ — это частный случай некоторого вычисленного ответа.

# ПОЛНОТА ОПЕРАЦИОННОЙ СЕМАНТИКИ

## Поясняющий пример.

Рассмотрим запрос $?P(U, V)$ к логической программе

$$\mathcal{P}: \quad \begin{array}{ll} P(f(X), Y) \leftarrow R(X); & (1) \\ R(b) \leftarrow; & (2) \\ R(c) \leftarrow; & (3) \end{array}$$

Легко видеть, что $\theta = \{U/f(b), V/c\}$ — это правильный ответ на запрос к программе.

Вместе с тем, единственный вычисленный ответ — это $\eta = \{U/f(b), V/Y\}$.

Все дело в том, что $\theta$ — это частный случай $\eta$: $\theta = \eta\{Y/c\}$.

КОНЕЦ ЛЕКЦИИ 14.